Universidad Nacional Mayor de San Marcos

Facultad de Ciencias Matemáticas

Escuela Profesional de Computación Científica



Plan Curricular 2026

– Reporte Final–

Última modificación: Febrero de 2026

Este documento está basado en la propuesta internacional de ACM/IEEE-CS *Computing Curricula*, que a su vez sirvió como base de la versión en Español creada por la Sociedad Peruana de Computación (SPC)

ISBN:

Todos los derechos reservados. Ernesto Cuadros-Vargas, Universidad Nacional Mayor de San Marcos (UNMSM).

Equipo de Trabajo

Dra. Maria Zegarra-Garay
Decana de la Facultad de Ciencias Matemáticas
email: *mzegarrag@unmsm.edu.pe*

Mag. Jorge Icaro Condado Jáuregui
Director (e) de Computación Científica
email: *jcondadoj@unmsm.edu.pe*

Dr. Frida Mereyda López Córdova
Dra. Ing de Sistemas e Informática, UNMSM (2025)
Docente del Departamento Académico de Matemáticas
email: *flopezc@unmsm.edu.pe*

Mg. Victoria Ysabel Rojas Rojas
Profesora de la Facultad de Ciencias Matemáticas de la UNMSM
email: *vrojasr@unmsm.edu.pe*

Mg. Rubén Arbañil
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *rarbanilr@unmsm.edu.pe*

Lic. Máximo Gerardo Gonzales Chávez
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *mgonzalesc1@unmsm.edu.pe*

Mg. Oscar Benito Pacheco
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *obenitop@unmsm.edu.pe*

Dra. Rosa Medina Aguilar
Profesora de la Facultad de Ciencias Matemáticas de la UNMSM
email: *rmedinaa1@unmsm.edu.pe*

Mg. Luz Teresa Quispe Vega
Profesora de la Facultad de Ciencias Matemáticas de la UNMSM
email: *lquispev@unmsm.edu.pe*

Mg. Miriam Gisell Pescorán Florencio
Profesora de la Facultad de Ciencias Matemáticas de la UNMSM
email: *mpescoranf@unmsm.edu.pe*

Lic. Joseph Wilmer Simpe Laura
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *jsimpel@unmsm.edu.pe*

Mg. Francisco Felipe Quiroz Chueca
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *fquirozc@unmsm.edu.pe*

Mg. Rómulo Fernando Lomparte Alvarado
Profesor de la Facultad de Ciencias Matemáticas de la UNMSM
email: *rlompartea@unmsm.edu.pe*

Dr. Omar Stevenson Guzmán Rea
Dr. Matemáticas. Universidad Federal de Pernambuco (2020)
Docente del Departamento Académico de Matemáticas
email: *oguzmanr@unmsm.edu.pe*

Ernesto Cuadros-Vargas
Dr. en Ciencia de la Computación, Universidad de Sao Paulo, Brasil (2004)
Orador distinguido para la *Association of Computing Machinery* (ACM) (2025-2027)
Miembro del Directorio de Gobernadores de la Sociedad de Computación del IEEE (2020-2023)
Miembro del *Steering Committee ACM/IEEE-CS Computing Curricula 2020 (CS2020)*
Miembro del *Steering Committee ACM/IEEE-CS Computing Curricula/Computer Science (CS2013)*
Secretario Ejecutivo del Centro Latinoamericano de Estudios en Informática (CLEI) 2008-2016
email: *ecuadros@spc.org.pe*

Representantes estudiantiles

Maria Fernanda Rupay Caballero
Estudiante de Computación Científica,
UNMSM, 2024
email: *maria.rupayc@unmsm.edu.pe*

Sebastián Alfredo Borja Ramos
Estudiante de Computación Científica,
UNMSM, 2024
email: *sebastian.borja@unmsm.edu.pe*

Angel Aarón Reyes Cáceres
Estudiante de Computación Científica,
UNMSM, 2024
email: *angel.reyesc@unmsm.edu.pe*

Carlos Jesús López Bravo
Estudiante de Computación Científica,
UNMSM, 2024
email: *carlos.lopez22@unmsm.edu.pe*

Julio André Carrillo Montero
Estudiante de Computación Científica,
UNMSM, 2023
email: *julio.carrillom@unmsm.edu.pe*

Adrian Enrique Borstelmann Segura
Estudiante de Computación Científica,
UNMSM, 2025
email: *adrian.borstelmanns@unmsm.edu.pe*

Jan Lucas Mancinelli Vite
Estudiante de Computación Científica,
UNMSM, 2023
email: *jan.mancinelli@unmsm.edu.pe*

Adriano Emer Valdivia
Estudiante de Computación Científica,
UNMSM, 2025
email: *adriano.valdivia@unmsm.edu.pe*

Cristian Josue Bernal Nuñez
Estudiante de Computación Científica,
UNMSM, 2023
email: *cristian.bernal@unmsm.edu.pe*

Jhanilde Pozo Hernandez
Estudiante de Computación Científica,
UNMSM, 2023
email: *jhanilde.pozo@unmsm.edu.pe*

Franco David Zevallos Silva
Estudiante de Computación Científica,
UNMSM, 2025
email: *franco.zevalloss@unmsm.edu.pe*

Angello Andre Manyari Medina
Estudiante de Computación Científica,
UNMSM, 2023
email: *angello.manyari@unmsm.edu.pe*

Resumen ejecutivo

Este documento representa el informe de la nueva malla curricular 2026 del Escuela Profesional de Computación Científica (Matemática Computacional) de la Universidad Nacional Mayor de San Marcos (*https://www.unmsm.edu.pe*) en la ciudad de Lima-Perú.

Este documento presenta el diseño curricular de la carrera de Matemática Computacional, una disciplina fundamental que se sitúa en la intersección entre la matemática pura y aplicada, y las ciencias de la computación. El diseño se alinea con los estándares internacionales de acreditación en matemáticas y se nutre de los marcos de referencia del *Computing Curricula* CC2020, desarrollado por la *Association for Computing Machinery* (ACM) y la *IEEE Computer Society* (IEEE-CS). Esta doble vertiente garantiza una formación sólida y moderna.

La propuesta adapta estos referentes globales a nuestro contexto regional, incorporando:

- Demandas específicas del sector productivo local en modelado, simulación y análisis de datos.
- Fortalezas institucionales en investigación matemática y su aplicación a problemas complejos.
- Tendencias emergentes que requieren un pensamiento abstracto y algorítmico robusto.

La Matemática Computacional se distingue de otros perfiles de la computación por su enfoque central en la teoría y los fundamentos matemáticos para resolver problemas complejos. Se diferencia y complementa a:

- **Ciencia de la Computación (CS)**: Profundizando en la base matemática de los algoritmos y la teoría de la computación.
- **Ingeniería de Software (SE)**: Aportando el rigor formal para la verificación y corrección de sistemas.
- **Ciencia de Datos (DS)**: Proporcionando el andamiaje teórico para el aprendizaje automático y el análisis estadístico.
- **Ingeniería de Computación (CE)**: Colaborando en el modelado y simulación de sistemas físicos y digitales.

El modelo pedagógico se sustenta en tres ejes fundamentales:

- **Rigor Matemático**: Dominio profundo del análisis, álgebra, probabilidad y estructuras discretas como base de todo el desarrollo computacional.
- **Implementación Efectiva**: Capacidad para traducir modelos y teorías matemáticas en algoritmos eficientes y software robusto.
- **Aplicación Interdisciplinaria**: Habilidad para modelar problemas del mundo real en áreas como finanzas, bioinformática, inteligencia artificial y física computacional.

Esta malla producirá profesionales con:

- Capacidad para desarrollar modelos matemáticos novedosos y diseñar algoritmos para problemas sin solución trivial.
- Competencias para trabajar en equipos multidisciplinarios, actuando como el puente entre la teoría abstracta y la solución práctica.
- Una formación ética que le permita comprender las implicaciones de los modelos algorítmicos en la sociedad, promoviendo la transparencia y la equidad.

Esta propuesta incorpora mecanismos de actualización continua, asegurando su relevancia frente a la rápida evolución de los campos de la matemática aplicada y la computación, y manteniendo un alineamiento con los más altos estándares de acreditación internacional.

Índice general

Agradecimientos xxvii

Abreviaturas xxix

1. Introducción **1**
1.1. Definiciones básicas . . . 1
1.2. Campo y mercado ocupacional . . . 1
1.3. Importancia de la carrera en la sociedad . . . 2
1.4. Información institutional . . . 3
1.4.1. Misión . . . 3
1.4.2. Visión . . . 3
1.5. Información de acreditacion . . . 3
1.5.1. Resultados de la carrera *Outcomes* . . . 3
1.5.2. Objetivos educacionales . . . 4
1.5.3. Aprendizaje esperado (*Learning Outcomes*) . . . 4
1.6. Perfiles . . . 4
1.6.1. Perfil del estudiante . . . 4
1.6.2. Perfil del graduado . . . 5
1.6.3. Perfil Profesional . . . 5
1.6.4. Perfil del docente . . . 6
1.7. Grados y Títulos . . . 6
1.8. Recursos para dictado de clases . . . 6
1.9. Organización del documento . . . 7

2. Cuerpo del conocimiento de Matemática Computacional **9**
2.1. Análisis Numérico y Científico (NSA) . . . 10
2.1.1. NSA/Aritmética Computacional y Análisis de Error (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 10
2.1.2. NSA/álgebra Lineal Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 10
2.1.3. NSA/Búsqueda de Raíces y Optimización Numérica Básica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 11
2.1.4. NSA/Interpolación y Aproximación de Funciones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 11
2.1.5. NSA/Diferenciación e Integración Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr) 11
2.1.6. NSA/Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 12
2.2. Modelado Matemático y Simulación (MMS) . . . 12
2.2.1. MMS/Formulación de Modelos Matemáticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 12
2.2.2. MMS/Simulación de Sistemas Discretos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 13
2.2.3. MMS/Simulación de Sistemas Continuos (Core Tier1: 1 hr, Core Tier2: 1 hr) . 13
2.2.4. MMS/Análisis de Sensibilidad y Robustez (Core Tier1: 1 hr, Core Tier2: 1 hr) 13
2.2.5. MMS/Validación y Verificación de Modelos (Core Tier1: 1 hr, Core Tier2: 1 hr) 14
2.3. Probabilidad y Computación Estadística (PSC) . . . 14
2.3.1. PSC/Probabilidad Avanzada y Procesos Estocásticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 14

2.3.2. PSC/Inferencia Estadística Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr) 15
2.3.3. PSC/Diseño de Experimentos Computacionales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 15
2.3.4. PSC/Métodos de Remuestreo y Validación (Core Tier1: 1 hr, Core Tier2: 1 hr) 15
2.4. Estructuras de Datos y Algoritmos (DSA) . . . 16
2.4.1. DSA/Complejidad Algorítmica y Análisis Asintótico (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 16
2.4.2. DSA/Estructuras de Datos Fundamentales y Avanzadas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 16
2.4.3. DSA/Algoritmos de Búsqueda y Ordenamiento (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 17
2.4.4. DSA/Algoritmos de Grafos y Optimización Combinatoria (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 17
2.4.5. DSA/Algoritmos Paralelos y Distribuidos (Core Tier1: 1 hr, Core Tier2: 1 hr) . 17
2.5. Computación de Alto Rendimiento (HPC) . . . 17
2.5.1. HPC/Arquitecturas de Computadoras Paralelas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 18
2.5.2. HPC/Programación Paralela de Memoria Compartida (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 18
2.5.3. HPC/Programación Paralela de Memoria Distribuida (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 18
2.5.4. HPC/Programación de Aceleradores (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 19
2.5.5. HPC/Métodos Numéricos Paralelos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 19
2.6. Optimización y Aprendizaje Automático (OML) . . . 19
2.6.1. OML/Optimización Convexa y No Convexa (Core Tier1: 1 hr, Core Tier2: 1 hr) 20
2.6.2. OML/Optimización con Restricciones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 20
2.6.3. OML/Optimización para Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 20
2.6.4. OML/Optimización a Gran Escala y Distribuida (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 21
2.6.5. OML/Metaheurísticas y Optimización Bioinspirada (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 21
2.7. Matemáticas para Ciencia de Datos y Aprendizaje Automático (DSM) . . . 21
2.7.1. DSM/Fundamentos del Aprendizaje Estadístico (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 22
2.7.2. DSM/Métodos de Aprendizaje Supervisado (Core Tier1: 1 hr, Core Tier2: 1 hr) 22
2.7.3. DSM/Métodos de Aprendizaje No Supervisado (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 22
2.7.4. DSM/Evaluación y Selección de Modelos (Core Tier1: 1 hr, Core Tier2: 1 hr) . 22
2.8. Matemáticas para Inteligencia Artificial (AIM) . . . 23
2.8.1. AIM/Teoría del Aprendizaje Profundo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 23
2.8.2. AIM/Arquitecturas Neuronales Avanzadas (Core Tier1: 1 hr, Core Tier2: 1 hr) 23
2.8.3. AIM/Aprendizaje por Refuerzo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 24
2.8.4. AIM/Procesamiento de Lenguaje Natural con Transformers (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 24
2.8.5. AIM/Modelos Generativos y GANs (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 24
2.9. Computación Simbólica y Discreta (SDC) . . . 25
2.9.1. SDC/Teoría de Grupos y Algoritmos Algebraicos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 25
2.9.2. SDC/Teoría de Números Computacional y Criptografía (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 25
2.9.3. SDC/Sistemas de Computación Simbólica (Core Tier1: 1 hr, Core Tier2: 1 hr) 25
2.9.4. SDC/Lógica Computacional y Verificación Formal (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 26
2.10. Señales, Imágenes y Visualización (SIV) . . . 26
2.10.1. SIV/Análisis de Fourier y Wavelets (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 26

2.10.2. SIV/Procesamiento Digital de Imágenes y Visión Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 27
2.10.3. SIV/Visualización Científica y Computación Visual (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 27
2.10.4. SIV/Gráficos por Computadora y Geometría Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 27
2.11. Computación Cuántica y Algoritmos (QCA) . . . 28
2.11.1. QCA/Fundamentos de Mecánica Cuántica para Computación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 28
2.11.2. QCA/Qubits y Circuitos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 28
2.11.3. QCA/Algoritmos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 28
2.11.4. QCA/Criptografía Post-Cuántica y Simulación Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 29
2.12. Ciberseguridad y Privacidad (CSP) . . . 29
2.12.1. CSP/Criptografía Simétrica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 29
2.12.2. CSP/Criptografía Asimétrica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 30
2.12.3. CSP/Computación con Preservación de la Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 30
2.12.4. CSP/Teoría de la Información y Compresión de Datos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 30
2.13. Matemáticas Computacionales para Dominios Específicos (CDM) . . . 31
2.13.1. CDM/Bioinformática y Biología Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 31
2.13.2. CDM/Finanzas Computacionales y Ciencia del Riesgo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 31
2.13.3. CDM/Física Computacional y Química Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 32
2.13.4. CDM/Ciencia de Materiales Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr) 32
2.13.5. CDM/Modelado Climático y Sistemas Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 32
2.14. Gobernanza y Gestión de Datos (DGM) . . . 32
2.14.1. DGM/ética en IA y Algoritmos Responsables (Core Tier1: 1 hr, Core Tier2: 1 hr) 33
2.14.2. DGM/Gestión de Proyectos de Datos y MLOps (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 33
2.14.3. DGM/Gobernanza de Datos y Cumplimiento Normativo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 33

3. Cuerpo del conocimiento de Ciencia de la Computación–*Computer Science* 35
3.1. Inteligencia Artificial (AI) . . . 37
3.1.1. AI/Introducción a la IA: Conceptos y Comportamiento Inteligente (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 39
3.1.2. AI/Características del Problema y Naturaleza de los Agentes (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 39
3.1.3. AI/Estado y Búsqueda No Informada (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . 39
3.1.4. AI/Búsqueda Heurística y CSP (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 40
3.1.5. AI/Búsqueda Adversarial y Optimización Estocástica (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . . 40
3.1.6. AI/Representación del Conocimiento y Razonamiento Probabilístico Básico (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 41
3.1.7. AI/Independencia, Modelos de Markov y Toma de Decisiones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 41
3.1.8. AI/Fundamentos de Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 41
3.1.9. AI/Aprendizaje Automático: Datos, Representaciones y Evaluación (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 42
3.1.10. AI/Redes Neuronales y Ética (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 43

3.1.11. AI/Formulación del Aprendizaje Automático y Evaluación (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 43
3.1.12. AI/Aprendizaje Profundo, RL y Ética (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . 43
3.1.13. AI/Entrenamiento y Optimización de Redes Neuronales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 44
3.1.14. AI/Técnicas de Regularización para Redes Profundas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 44
3.1.15. AI/Arquitecturas de Redes Convolucionales y Residuales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 44
3.1.16. AI/Arquitecturas Transformer y Modelos Generativos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 44
3.1.17. AI/Aplicaciones e Impacto Social de la IA (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 45
3.1.18. AI/Aplicaciones Adicionales e Impacto Social de la IA (Core Tier1: 2 hrs, Core Tier2: 2 hrs) . . . 45
3.1.19. AI/Representación Lógica y Razonamiento . . . 46
3.1.20. AI/Representación Probabilística y Razonamiento . . . 46
3.1.21. AI/Planificación . . . 47
3.1.22. AI/Agentes y Sistemas Cognitivos . . . 47
3.1.23. AI/Ingeniería de Bucles de Agentes (Loop Engineering) . . . 48
3.1.24. AI/Procesamiento del Lenguaje Natural . . . 48
3.1.25. AI/Robótica . . . 49
3.1.26. AI/Percepción y Visión por Computadora . . . 50
3.2. Fundamentos Algorítmicos (AL) . . . 51
3.2.1. AL/Estructuras de Datos Fundamentales (Core Tier1: 5 hrs, Core Tier2: 3 hrs) 52
3.2.2. AL/Algoritmos Fundamentales (Core Tier1: 4 hrs, Core Tier2: 2 hrs) . . . 52
3.2.3. AL/Algoritmos Avanzados (Core Tier1: 3 hrs, Core Tier2: 2 hrs) . . . 53
3.2.4. AL/Estrategias Algorítmicas (Core Tier1: 6 hrs) . . . 53
3.2.5. AL/Marco de Análisis de Complejidad (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . 54
3.2.6. AL/Notación Asintótica y Clases de Complejidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 54
3.2.7. AL/Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 55
3.2.8. AL/Teoría de la Complejidad Computacional (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 55
3.2.9. AL/Lenguajes Formales y Autómatas (Core Tier1: 4 hrs, Core Tier2: 10 hrs) . 55
3.2.10. AL/Computabilidad y Decidibilidad (Core Tier1: 6 hrs, Core Tier2: 14 hrs) . . 56
3.2.11. AL/Sociedad, ética y la Profesión . . . 57
3.3. Arquitectura y Organización (AR) . . . 57
3.3.1. AR/Lógica Digital y Sistemas Digitales (Core Tier2: 3 hrs) . . . 58
3.3.2. AR/Representación de Datos a Nivel de Máquina (Core Tier1: 1 hr) . . . 58
3.3.3. AR/Organización de Máquina a Nivel de Ensamblador (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 59
3.3.4. AR/Jerarquía de Memoria (Core Tier1: 6 hrs) . . . 60
3.3.5. AR/Interconexión y Comunicación (Core Tier1: 1 hr) . . . 60
3.3.6. AR/Organización Funcional (Core Tier2: 2 hrs) . . . 60
3.3.7. AR/Rendimiento y Eficiencia Energética (Core Tier2: 3 hrs) . . . 61
3.3.8. AR/Arquitecturas Heterogéneas (Core Tier2: 2 hrs) . . . 61
3.3.9. AR/Arquitecturas de Procesador Seguras (Core Tier2: 2 hrs) . . . 62
3.3.10. AR/Arquitecturas Cuánticas (Core Tier2: 2 hrs) . . . 62
3.3.11. AR/Problemas de Sostenibilidad . . . 63
3.4. Gestión de Datos (DM) . . . 63
3.4.1. DM/El Rol de los Datos y el Ciclo de Vida de los Datos (Core Tier1: 2 hrs) . . 65
3.4.2. DM/Conceptos Fundamentales de Sistemas de Bases de Datos (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 65
3.4.3. DM/Modelado de Datos (Core Tier1: 2 hrs, Core Tier2: 3 hrs) . . . 66
3.4.4. DM/Bases de Datos Relacionales (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . . 66

3.4.5. DM/Construcción de Consultas (Core Tier1: 2 hrs, Core Tier2: 4 hrs) 67
3.4.6. DM/Procesamiento de Consultas (Core Tier2: 4 hrs) 67
3.4.7. DM/Internos de DBMS (Core Tier2: 4 hrs) 68
3.4.8. DM/Sistemas NoSQL (Core Tier2: 2 hrs) 68
3.4.9. DM/Seguridad y Privacidad de Datos (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . 69
3.4.10. DM/Análisis de Datos (Core Tier2: 3 hrs) 69
3.4.11. DM/Bases de Datos Distribuidas/Computación en la Nube 70
3.4.12. DM/Bases de Datos Semiestructuradas y No Estructuradas 70
3.4.13. DM/Sociedad, ética y Profesión 70
3.5. Fundamentos de los Lenguajes de Programación (FPL) 71
3.5.1. FPL/Fundamentos de Programación Orientada a Objetos (Core Tier1: 2 hrs, Core Tier2: 1 hr) 72
3.5.2. FPL/Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (Core Tier1: 2 hrs, Core Tier2: 1 hr) 73
3.5.3. FPL/Programación Funcional (Core Tier1: 4 hrs, Core Tier2: 3 hrs) 73
3.5.4. FPL/Programación Lógica (Core Tier2: 2 hrs) 74
3.5.5. FPL/Scripting de Shell (Core Tier1: 2 hrs) 75
3.5.6. FPL/Programación Dirigida por Eventos y Reactiva (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 75
3.5.7. FPL/Computación Paralela y Distribuida (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 76
3.5.8. FPL/Programación Orientada a Aspectos 76
3.5.9. FPL/Fundamentos de Sistemas de Tipos (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 77
3.5.10. FPL/Sistemas de Tipos: Polimorfismo y Tipado Complementario (Core Tier1: 1 hr, Core Tier2: 1 hr) 77
3.5.11. FPL/Teoría de Tipos Avanzada (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 78
3.5.12. FPL/Modelo de Ejecución y Memoria de Sistemas (Core Tier1: 2 hrs) 78
3.5.13. FPL/Gestión de Memoria (Core Tier1: 1 hr) 79
3.5.14. FPL/Traducción y Ejecución de Lenguajes (Core Tier1: 2 hrs, Core Tier2: 3 hrs) 79
3.5.15. FPL/Abstracción y Representación de Programas (Core Tier2: 3 hrs) 80
3.5.16. FPL/Análisis Sintáctico 81
3.5.17. FPL/Análisis Semántico del Compilador 81
3.5.18. FPL/Análisis de Programas y Analizadores 82
3.5.19. FPL/Generación de Código 82
3.5.20. FPL/Comportamiento en Tiempo de Ejecución y Sistemas 83
3.5.21. FPL/Construcciones de Programación Avanzadas 83
3.5.22. FPL/Pragmática del Lenguaje 84
3.5.23. FPL/Semántica Formal 84
3.5.24. FPL/Metodologías de Desarrollo Formal 85
3.5.25. FPL/Principios de Diseño de Lenguajes de Programación 85
3.5.26. FPL/Sociedad, ética y Profesión 86
3.6. Gráficos y Técnicas Interactivas (GIT) 86
3.6.1. GIT/Conceptos Fundamentales de Gráficos y Técnicas Interactivas (Core Tier1: 4 hrs, Core Tier2: 3 hrs) 87
3.6.2. GIT/Visualización (Core Tier2: 6 hrs) 87
3.6.3. GIT/Renderizado Aplicado y Técnicas (Core Tier2: 15 hrs) 88
3.6.4. GIT/Modelado Geométrico (Core Tier2: 6 hrs) 88
3.6.5. GIT/Sombreado y Renderizado Avanzado (Core Tier2: 6 hrs) 88
3.6.6. GIT/Animación por Computadora (Core Tier2: 6 hrs) 89
3.6.7. GIT/Simulación (Core Tier2: 6 hrs) 89
3.6.8. GIT/Inmersión (Core Tier2: 6 hrs) 89
3.6.9. GIT/Interacción (Core Tier2: 4 hrs) 89
3.6.10. GIT/Procesamiento de Imágenes (Core Tier2: 6 hrs) 90
3.6.11. GIT/Computación Tangible/Física (Core Tier2: 6 hrs) 90
3.6.12. GIT/Sociedad, ética y Profesión 90
3.7. Interacción Humano-Computadora (HCI) 91

3.7.1. HCI/Comprender al Usuario: Objetivos individuales e interacciones con otros (Core Tier1: 2 hrs, Core Tier2: 5 hrs) . . . 91
3.7.2. HCI/Responsabilidad y Rendición de Cuentas en el Diseño (Core Tier1: 2 hrs, Core Tier2: 2 hrs) . . . 92
3.7.3. HCI/Accesibilidad y Diseño Inclusivo (Core Tier1: 2 hrs, Core Tier2: 2 hrs) . . 92
3.7.4. HCI/Evaluación del Diseño (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 92
3.7.5. HCI/Diseño de Sistemas (Core Tier1: 1 hr, Core Tier2: 5 hrs) . . . 93
3.7.6. HCI/Sociedad, ética y Profesión . . . 93
3.8. Fundamentos Matemáticos y Estadísticos (MSF) . . . 93
3.8.1. MSF/Matemáticas Discretas (Core Tier1: 15 hrs, Core Tier2: 6 hrs) . . . 94
3.8.2. MSF/Matemáticas Discretas Avanzadas (Core Tier1: 15 hrs, Core Tier2: 6 hrs) 94
3.8.3. MSF/Probabilidad (Core Tier1: 11 hrs, Core Tier2: 29 hrs) . . . 95
3.8.4. MSF/Estadística (Core Tier1: 10 hrs, Core Tier2: 30 hrs) . . . 95
3.8.5. MSF/álgebra Lineal (Core Tier1: 5 hrs, Core Tier2: 35 hrs) . . . 96
3.8.6. MSF/Cálculo (Core Tier2: 40 hrs) . . . 96
3.9. Redes y Comunicaciones (NC) . . . 97
3.9.1. NC/Fundamentos de Redes y Comunicaciones (Core Tier1: 3 hrs) . . . 97
3.9.2. NC/Aplicaciones en Red (Core Tier1: 4 hrs) . . . 98
3.9.3. NC/Soporte de Fiabilidad (Core Tier2: 6 hrs) . . . 98
3.9.4. NC/Enrutamiento y Reenvío (Core Tier2: 4 hrs) . . . 99
3.9.5. NC/Comunicación de un solo salto (Core Tier2: 3 hrs) . . . 99
3.9.6. NC/Seguridad de Red (Core Tier2: 3 hrs) . . . 99
3.9.7. NC/Soporte de Movilidad (Core Tier2: 4 hrs) . . . 100
3.9.8. NC/Temas Emergentes (Core Tier2: 4 hrs) . . . 100
3.10. Sistemas Operativos (OS) . . . 100
3.10.1. OS/Rol y Propósito de los Sistemas Operativos (Core Tier1: 2 hrs) . . . 101
3.10.2. OS/Principios del Sistema Operativo (Core Tier1: 2 hrs) . . . 102
3.10.3. OS/Concurrencia (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 102
3.10.4. OS/Protección y Seguridad (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 103
3.10.5. OS/Planificación (Core Tier2: 2 hrs) . . . 104
3.10.6. OS/Modelo de Proceso (Core Tier2: 2 hrs) . . . 104
3.10.7. OS/Gestión de Memoria (Core Tier2: 1 hr) . . . 105
3.10.8. OS/Gestión de Dispositivos (Core Tier2: 1 hr) . . . 105
3.10.9. OS/API de Sistemas de Archivos e Implementación (Core Tier2: 2 hrs) . . . 106
3.10.10. OS/Sistemas de Archivos Avanzados (Core Tier2: 1 hr) . . . 106
3.10.11. OS/Virtualización (Core Tier2: 1 hr) . . . 107
3.10.12. OS/Sistemas en Tiempo Real y Embebidos (Core Tier2: 1 hr) . . . 108
3.10.13. OS/Tolerancia a fallos (Core Tier2: 1 hr) . . . 108
3.10.14. OS/Sociedad, ética y Profesión . . . 109
3.11. Computación Paralela y Distribuida (PDC) . . . 109
3.11.1. PDC/Programas: Paralelismo Declarativo (Core Tier1: 1 hr, Core Tier2: 1 hr) 110
3.11.2. PDC/Programas: Inicio de Actividades (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 111
3.11.3. PDC/Programas: Propiedades de Ejecución (Core Tier1: 1 hr, Core Tier2: 1 hr) 111
3.11.4. PDC/Programas: Distribución (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 111
3.11.5. PDC/Programas: Mapeos de Implementación (Core Tier1: 1 hr, Core Tier2: 1 hr) 112
3.11.6. PDC/Programación GPU . . . 112
3.11.7. PDC/Comunicación (Core Tier1: 2 hrs, Core Tier2: 4 hrs) . . . 112
3.11.8. PDC/Comunicación: Propiedades y Extensiones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 113
3.11.9. PDC/Memoria y Consistencia (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 114
3.11.10. PDC/Coordinación (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . . 114
3.11.11. PDC/Coordinación: Sincronización y Atomicidad (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 115
3.11.12. PDC/Coordinación: Propiedades Avanzadas (Core Tier1: 1 hr, Core Tier2: 2 hrs) 115
3.11.13. PDC/Evaluación (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . . 115
3.11.14. PDC/Evaluación: Métodos Formales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 116

3.11.15.PDC/Evaluación: Pruebas y Medición (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 116
3.11.16.PDC/Algoritmos (Core Tier1: 1 hr, Core Tier2: 3 hrs) 117
3.11.17.PDC/Algoritmos: Dominios de Aplicación (Core Tier1: 1 hr, Core Tier2: 3 hrs) 117
3.11.18.PDC/Algoritmos: Dominios Algorítmicos (Core Tier1: 1 hr, Core Tier2: 3 hrs) 118
3.12. Fundamentos de Desarrollo de Software (SDF) 118
3.12.1. SDF/Conceptos y Construcciones Básicas de Programación (Core Tier1: 10 hrs) 119
3.12.2. SDF/Prácticas de Programación: Pruebas, Depuración y Documentación (Core Tier1: 10 hrs) 120
3.12.3. SDF/Estructuras de Datos Fundamentales (Core Tier1: 12 hrs) 120
3.12.4. SDF/Algoritmos (Core Tier1: 6 hrs) 121
3.12.5. SDF/Prácticas de Desarrollo de Software (Core Tier1: 5 hrs) 121
3.12.6. SDF/Sociedad, ética y la Profesión 121
3.13. Ingeniería de Software (SE) 122
3.13.1. SE/Colaboración y Dinámica de Equipo (Core Tier1: 2 hrs, Core Tier2: 2 hrs) 123
3.13.2. SE/Diversidad, Stakeholders y Equipos Distribuidos (Core Tier1: 1 hr, Core Tier2: 1 hr) 124
3.13.3. SE/Control de Versiones y CI/CD (Core Tier1: 1 hr, Core Tier2: 2 hrs) 124
3.13.4. SE/Entorno de Desarrollo e IDE (Core Tier1: 1 hr, Core Tier2: 2 hrs) 125
3.13.5. SE/Ingeniería de Requisitos (Core Tier2: 2 hrs) 125
3.13.6. SE/Evolución y Estimación de Requisitos (Core Tier2: 1 hr) 126
3.13.7. SE/Principios y Arquitectura de Software (Core Tier1: 1 hr, Core Tier2: 2 hrs) 126
3.13.8. SE/Diseño de API y Modelado (Core Tier1: 1 hr, Core Tier2: 1 hr) 127
3.13.9. SE/Calidad y Evaluación del Diseño (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 127
3.13.10.SE/Prácticas de Codificación (Core Tier1: 1 hr, Core Tier2: 2 hrs) 128
3.13.11.SE/Construcción a Escala y Proceso (Core Tier1: 1 hr, Core Tier2: 2 hrs) . . . 128
3.13.12.SE/Conceptos de verificación y validación (Core Tier1: 1 hr, Core Tier2: 1 hr) 129
3.13.13.SE/Planificación y Tipos de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 129
3.13.14.SE/Prácticas Avanzadas de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 130
3.13.15.SE/Herramientas y Análisis de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 130
3.13.16.SE/Pruebas de Rendimiento y Evaluación Comparativa (Core Tier1: 1 hr, Core Tier2: 1 hr) 131
3.13.17.SE/Compatibilidad y Versionado de API (Core Tier2: 2 hrs) 131
3.13.18.SE/Técnicas de Refactorización (Core Tier2: 1 hr) 132
3.13.19.SE/Conceptos de Confiabilidad (Core Tier2: 2 hrs) 132
3.13.20.SE/Ingeniería de Confiabilidad (Core Tier2: 1 hr) 133
3.13.21.SE/Métodos Formales 133
3.14. Seguridad (SEC) 134
3.14.1. SEC/Principios y Mentalidad de Seguridad (Core Tier1: 1 hr, Core Tier2: 3 hrs) 135
3.14.2. SEC/Control de Acceso y Aplicaciones (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . 136
3.14.3. SEC/Seguridad en IA y Hardware (Core Tier1: 1 hr, Core Tier2: 2 hrs) 136
3.14.4. SEC/Ética, Ley y Sociedad en Seguridad (Core Tier1: 1 hr, Core Tier2: 2 hrs) 137
3.14.5. SEC/Privacidad y Seguridad por Diseño (Core Tier1: 1 hr, Core Tier2: 1 hr) . 137
3.14.6. SEC/Inyección y Validación de Entrada (Core Tier1: 1 hr, Core Tier2: 2 hrs) . 138
3.14.7. SEC/Seguridad de Memoria y Tipos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 138
3.14.8. SEC/Flujo de Información y No Interferencia (Core Tier1: 1 hr, Core Tier2: 1 hr) 139
3.14.9. SEC/Análisis de Malware y Seguridad Avanzada (Core Tier1: 1 hr, Core Tier2: 3 hrs) 139
3.14.10.SEC/Fundamentos de Criptografía (Core Tier1: 1 hr, Core Tier2: 2 hrs) 140
3.14.11.SEC/Algoritmos Criptográficos (Core Tier1: 1 hr, Core Tier2: 2 hrs) 140
3.14.12.SEC/Protocolos, Aplicaciones y Criptografía Avanzada (Core Tier1: 1 hr, Core Tier2: 2 hrs) 140
3.14.13.SEC/Análisis de Amenazas e Ingeniería de Seguridad (Core Tier1: 1 hr, Core Tier2: 4 hrs) 141
3.14.14.SEC/Diseño e Ingeniería de Controles de Seguridad (Core Tier1: 1 hr, Core Tier2: 3 hrs) 142

3.14.15 SEC/Computación Confiable e Ingeniería de Privacidad (Core Tier1: 1 hr, Core Tier2: 3 hrs) . . . 142
3.14.16 SEC/Técnicas de Informática Forense (Core Tier2: 4 hrs) . . . 142
3.14.17 SEC/Aspectos Legales y Manejo de Incidentes (Core Tier2: 3 hrs) . . . 143
3.14.18 SEC/Gobierno, Gestión y Política de Seguridad (Core Tier2: 2 hrs) . . . 143
3.14.19 SEC/Gestión de Riesgos y Datos (Core Tier2: 1 hr) . . . 144
3.15. Sociedad, ética y la Profesión (SEP) . . . 144
3.15.1. SEP/Impacto Social e IA (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 145
3.15.2. SEP/Equidad Digital y Accesibilidad (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 146
3.15.3. SEP/Fundamentos y Marcos del Análisis Ético (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 146
3.15.4. SEP/Métodos y Herramientas de Análisis Ético (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 147
3.15.5. SEP/Valores, Naturaleza y Actualización Profesional (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 147
3.15.6. SEP/Códigos de Ética y Responsabilidad Legal (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 148
3.15.7. SEP/Responsabilidad Profesional y Política Pública (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 148
3.15.8. SEP/Disidencia Ética y Contexto Cultural (Core Tier1: 1 hr, Core Tier2: 1 hr) 149
3.15.9. SEP/Ética en el Lugar de Trabajo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 149
3.15.10 SEP/Fundamentos de la Propiedad Intelectual (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 149
3.15.11 SEP/Licencias, Plagio y Autoría (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 150
3.15.12 SEP/Derechos de PI Digital y Derechos de Autor (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 150
3.15.13 SEP/Open Source, Piratería y DRM (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 151
3.15.14 SEP/Datos, Vigilancia y Tecnología de Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 151
3.15.15 SEP/Fundamentos Filosóficos y Legislación de Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 151
3.15.16 SEP/Libertades Civiles y Expresión (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 152
3.15.17 SEP/Comunicación Técnica y Profesional (Core Tier1: 1 hr, Core Tier2: 1 hr) 152
3.15.18 SEP/Comunicación en Equipo y Cultura (Core Tier1: 1 hr, Core Tier2: 1 hr) . 153
3.15.19 SEP/Huella Ambiental de los Sistemas Informáticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 153
3.15.20 SEP/Efectos Sistémicos y Contexto Social (Core Tier1: 1 hr, Core Tier2: 1 hr) 154
3.15.21 SEP/Diseño Sostenible y Computación Ubicua (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 154
3.15.22 SEP/Fundamentos e Historia Temprana de la Informática (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 154
3.15.23 SEP/Historia Moderna: PC, Internet e IA (Core Tier1: 1 hr, Core Tier2: 1 hr) 155
3.15.24 SEP/Modelos y Estrategias Económicas (Core Tier2: 1 hr) . . . 155
3.15.25 SEP/Automatización, IA y Ética Económica (Core Tier2: 1 hr) . . . 156
3.15.26 SEP/Delitos y Ataques Informáticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 156
3.15.27 SEP/Leyes, Políticas y Responsabilidad en Seguridad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 157
3.15.28 SEP/Identidad y Diversidad en la Informática (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 157
3.15.29 SEP/Contexto Histórico e Intercultural de DEIA (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 157
3.15.30 SEP/Inclusión y Prácticas en Informática (Core Tier1: 1 hr, Core Tier2: 1 hr) 158
3.15.31 SEP/Diseño Accesible (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 158
3.16. Fundamentos de Sistemas (SF) . . . 159
3.16.1. SF/Componentes Hardware y Lógica (Core Tier1: 2 hrs) . . . 159

3.16.2. SF/Capas, Abstracciones y Escalabilidad (Core Tier1: 2 hrs) . . . 160
3.16.3. SF/Sistemas Digitales y Lógica (Core Tier1: 2 hrs) . . . 160
3.16.4. SF/Máquinas de Estado (Core Tier1: 2 hrs) . . . 161
3.16.5. SF/Procesamiento Secuencial y Paralelo (Core Tier1: 2 hrs) . . . 161
3.16.6. SF/Gestión de Recursos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 162
3.16.7. SF/Latencia, Caché y Jerarquía de Memoria (Core Tier1: 2 hrs, Core Tier2: 2 hrs) . . . 162
3.16.8. SF/Virtualización y Aislamiento (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 162
3.16.9. SF/Métricas y Benchmarks de Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr) 163
3.16.10. SF/Modelos Analíticos de Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 163
3.16.11. SF/Eventos, Herramientas y Experimentación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 164
3.16.12. SF/Confiabilidad del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 164
3.16.13. SF/Seguridad del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 165
3.16.14. SF/Diseño del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 165
3.16.15. SF/Sociedad, ética y la Profesión . . . 165
3.17. Desarrollo de Plataformas Especializadas (SPD) . . . 166
3.17.1. SPD/Aspectos Comunes: Plataformas, APIs y Restricciones (Core Tier1: 3 hrs, Core Tier2: 2 hrs) . . . 167
3.17.2. SPD/Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (Core Tier1: 2 hrs, Core Tier2: 1 hr) . . . 168
3.17.3. SPD/Frontend, Frameworks y Estándares Web (Core Tier2: 3 hrs) . . . 168
3.17.4. SPD/Arquitectura Cloud y Gestión de Datos (Core Tier2: 3 hrs) . . . 168
3.17.5. SPD/Plataformas Móviles: Fundamentos del Desarrollo (Core Tier2: 2 hrs) . . 169
3.17.6. SPD/Plataformas Móviles: Arquitectura y Restricciones (Core Tier2: 2 hrs) . . 169
3.17.7. SPD/Plataformas Móviles: Capacidades y Computación Asíncrona (Core Tier2: 1 hr) . . . 169
3.17.8. SPD/Plataformas Robóticas (Core Tier2: 4 hrs) . . . 170
3.17.9. SPD/Características y Arquitecturas Embebidas (Core Tier2: 1 hr) . . . 170
3.17.10. SPD/Sistemas, Programación y Sensores (Core Tier2: 2 hrs) . . . 171
3.17.11. SPD/Tiempo Real, Memoria y Verificación (Core Tier2: 3 hrs) . . . 171
3.17.12. SPD/Historia, Tipos y Características de Plataformas (Core Tier2: 1 hr) . . . 171
3.17.13. SPD/Sensores, Controladores y Ecosistemas eSports (Core Tier2: 1 hr) . . . 172
3.17.14. SPD/Arquitecturas y Renderizado (Core Tier2: 1 hr) . . . 172
3.17.15. SPD/Algoritmos, Medios y Mundos Virtuales (Core Tier2: 1 hr) . . . 173
3.17.16. SPD/Lenguajes, APIs y Herramientas de Desarrollo (Core Tier2: 1 hr) . . . 173
3.17.17. SPD/Motores de Juego (Core Tier2: 1 hr) . . . 173
3.17.18. SPD/Fundamentos del Diseño de Juegos (Core Tier2: 1 hr) . . . 174
3.17.19. SPD/Géneros, Audiencias y Alcance Social (Core Tier2: 1 hr) . . . 174
3.17.20. SPD/Plataformas Interactivas de Datos y Análisis (Core Tier2: 2 hrs) . . . 175
3.17.21. SPD/Programación con IA y Plataformas Cuánticas (Core Tier2: 1 hr) . . . 175
3.17.22. SPD/SPD-SEP/Móvil . . . 175
3.17.23. SPD/SPD-SEP Web . . . 176
3.17.24. SPD/Propiedad Intelectual en los Juegos . . . 176
3.17.25. SPD/Accesibilidad e Interfaz del Juego . . . 177
3.17.26. SPD/Salud, Seguridad y Sostenibilidad . . . 177
3.17.27. SPD/Representación, Inclusión e Igualdad . . . 178
3.17.28. SPD/SPD-SEP/Robótica . . . 178
3.17.29. SPD/SPD-SEP/Interactiva . . . 178

4. Cuerpo del conocimiento de Matemática **181**
4.1. Cálculo y Análisis (CAN) . . . 182
4.1.1. CAN/Límites y Continuidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 182
4.1.2. CAN/Cálculo Diferencial (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 183
4.1.3. CAN/Antiderivadas y Técnicas de Integración (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 183

4.1.4. CAN/Integrales Impropias (Core Tier1: 1 hr, Core Tier2: 1 hr) 183
4.1.5. CAN/Sucesiones, Series y Series de Potencias (Core Tier1: 1 hr, Core Tier2: 1 hr)184
4.1.6. CAN/Aplicaciones de la Integral Definida (Core Tier1: 1 hr, Core Tier2: 1 hr) 184
4.1.7. CAN/Geometría en el Espacio (Core Tier1: 1 hr, Core Tier2: 1 hr) 184
4.1.8. CAN/Superficies Cuádricas (Core Tier1: 1 hr, Core Tier2: 1 hr) 185
4.1.9. CAN/Ecuaciones Paramétricas y Coordenadas Polares (Core Tier1: 1 hr, Core Tier2: 1 hr) 185
4.1.10. CAN/Cálculo Diferencial Multivariable (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 185
4.1.11. CAN/Integrales Múltiples (Core Tier1: 1 hr, Core Tier2: 1 hr) 186
4.1.12. CAN/Integrales de Línea y Campos Vectoriales (Core Tier1: 1 hr, Core Tier2: 1 hr) 186
4.1.13. CAN/Integrales de Superficie y Teoremas Fundamentales (Core Tier1: 1 hr, Core Tier2: 1 hr) 187
4.1.14. CAN/Funciones Vectoriales de una Variable Real (Core Tier1: 1 hr, Core Tier2: 1 hr) 187
4.1.15. CAN/Transformaciones Multivariables (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 187
4.1.16. CAN/Análisis Real e Integración de Lebesgue (Core Tier1: 1 hr, Core Tier2: 1 hr)188
4.1.17. CAN/Análisis Complejo y Aplicaciones Conformes (Core Tier1: 1 hr, Core Tier2: 1 hr) 188
4.1.18. CAN/Análisis Funcional y Espacios de Banach/Hilbert 188
4.1.19. CAN/Análisis Armónico y Teoría de Fourier 189
4.1.20. CAN/Cálculo de Variaciones 189
4.1.21. CAN/Teoría de la Medida (Core Tier1: 1 hr, Core Tier2: 1 hr) 189
4.2. álgebra y Teoría de Números (ANT) 190
4.2.1. ANT/Matrices y Sistemas de Ecuaciones Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr) 190
4.2.2. ANT/Espacios Vectoriales y Transformaciones Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr) 191
4.2.3. ANT/Valores Propios, Producto Interno y Formas Cuadráticas (Core Tier1: 1 hr, Core Tier2: 1 hr) 191
4.2.4. ANT/Teoría de Grupos (Core Tier1: 1 hr, Core Tier2: 1 hr) 191
4.2.5. ANT/Anillos, Cuerpos y Teoría de Galois (Core Tier1: 1 hr, Core Tier2: 1 hr) 192
4.2.6. ANT/Teoría de Módulos y álgebra Conmutativa 192
4.2.7. ANT/Teoría Analítica de Números 192
4.2.8. ANT/Teoría Algebraica de Números 193
4.2.9. ANT/Teoría de Representaciones 193
4.2.10. ANT/Criptografía y Teoría de la Codificación (Core Tier1: 1 hr, Core Tier2: 1 hr)193
4.3. Geometría y Topología (GTO) 194
4.3.1. GTO/Geometría Euclidiana y Proyectiva (Core Tier1: 1 hr, Core Tier2: 1 hr) . 194
4.3.2. GTO/Topología General (Core Tier1: 1 hr, Core Tier2: 1 hr) 194
4.3.3. GTO/Geometría Diferencial de Curvas y Superficies (Core Tier1: 1 hr, Core Tier2: 1 hr) 195
4.3.4. GTO/Variedades Diferenciables y Formas Diferenciales (Core Tier1: 1 hr, Core Tier2: 1 hr) 195
4.3.5. GTO/Geometría Riemanniana 196
4.3.6. GTO/Topología Algebraica 196
4.3.7. GTO/Geometría Simpléctica 196
4.3.8. GTO/Teoría Geométrica de Grupos 197
4.4. Matemáticas Discretas y Combinatoria (DMC) 197
4.4.1. DMC/Combinatoria Enumerativa (Core Tier1: 1 hr, Core Tier2: 1 hr) 197
4.4.2. DMC/Teoría de Grafos (Core Tier1: 1 hr, Core Tier2: 1 hr) 198
4.4.3. DMC/Combinatoria Extremal y Probabilística 198
4.4.4. DMC/Optimización Discreta (Core Tier1: 1 hr, Core Tier2: 1 hr) 198
4.4.5. DMC/Teoría de Diseños y Códigos Combinatorios 199
4.4.6. DMC/Lógica Matemática y Teoría de Conjuntos (Core Tier1: 1 hr, Core Tier2: 1 hr) 199

4.5. Probabilidad y Estadística (PST) . . . 200
4.5.1. PST/Axiomas de Probabilidad y Distribuciones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 200
4.5.2. PST/Teoremas Límite (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 200
4.5.3. PST/Procesos Estocásticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 201
4.5.4. PST/Teoría de la Estimación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 201
4.5.5. PST/Contraste de Hipótesis (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 201
4.5.6. PST/Inferencia Bayesiana (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 202
4.5.7. PST/Estadística de Alta Dimensión . . . 202
4.5.8. PST/Análisis de Series Temporales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 202
4.5.9. PST/Métodos No Paramétricos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 203
4.6. Análisis Numérico y Científico (NSA) . . . 203
4.6.1. NSA/Análisis de Error y Aritmética de Punto Flotante (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 204
4.6.2. NSA/Búsqueda de Raíces de Ecuaciones No Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 204
4.6.3. NSA/álgebra Lineal Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 204
4.6.4. NSA/Teoría de Aproximación e Interpolación (Core Tier1: 1 hr, Core Tier2: 1 hr) 205
4.6.5. NSA/Integración Numérica y Cuadratura (Core Tier1: 1 hr, Core Tier2: 1 hr) . 205
4.6.6. NSA/Métodos Numéricos para Ecuaciones Diferenciales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 206
4.6.7. NSA/Método de Elementos Finitos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 206
4.6.8. NSA/Algoritmos de Optimización (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 206
4.6.9. NSA/Computación Paralela y de Alto Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 207
4.7. Modelado y Simulación Matemática (MMS) . . . 207
4.7.1. MMS/Ecuaciones Diferenciales Ordinarias de Primer Orden (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 208
4.7.2. MMS/Ecuaciones Diferenciales de Orden Superior y Sistemas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 208
4.7.3. MMS/Ecuaciones en Derivadas Parciales (Core Tier1: 1 hr, Core Tier2: 1 hr) . 208
4.7.4. MMS/Sistemas Dinámicos y Caos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 209
4.7.5. MMS/Análisis Asintótico y Métodos de Perturbación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 209
4.7.6. MMS/Dinámica de Fluidos Matemática . . . 209
4.7.7. MMS/Teoría de Control Matemático (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 210
4.7.8. MMS/Ecuaciones Diferenciales Estocásticas (Core Tier1: 1 hr, Core Tier2: 1 hr) 210
4.7.9. MMS/Modelado Basado en Agentes y Sistemas Complejos . . . 211
4.8. Fundamentos Matemáticos de la Ciencia de Datos (FDS) . . . 211
4.8.1. FDS/Teoría del Aprendizaje Estadístico (Core Tier1: 1 hr, Core Tier2: 1 hr) . 211
4.8.2. FDS/Optimización para Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 212
4.8.3. FDS/Análisis Topológico de Datos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 212
4.8.4. FDS/Reducción de Dimensionalidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 212
4.8.5. FDS/Métodos de Kernel y Espacios de Hilbert con Núcleo Reproductor (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 213
4.8.6. FDS/Teoría de la Información (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 213
4.8.7. FDS/Fundamentos Matemáticos del Aprendizaje Profundo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 214
4.9. Geometría y Topología Computacional (CGT) . . . 214
4.9.1. CGT/Diagramas de Voronoi y Triangulaciones de Delaunay (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 214
4.9.2. CGT/Envolventes Convexas y Politopos (Core Tier1: 1 hr, Core Tier2: 1 hr) . 215
4.9.3. CGT/Topología Digital y Análisis de Imágenes (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 215

4.9.4. CGT/Homología Persistente y Topología Aplicada (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 216
4.9.5. CGT/Generación de Mallas y Algoritmos Geométricos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 216
4.10. Información y Computación Cuántica (QIC) . . . 216
4.10.1. QIC/Fundamentos de Mecánica Cuántica para Computación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 217
4.10.2. QIC/Circuitos y Puertas Cuánticas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 217
4.10.3. QIC/Algoritmos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 218
4.10.4. QIC/Entrelazamiento y Desigualdades de Bell (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 218
4.10.5. QIC/Corrección de Errores Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 218
4.10.6. QIC/Redes de Tensores y Simulación Cuántica . . . 219
4.11. Biología Matemática y Ciencias de la Vida (MBL) . . . 219
4.11.1. MBL/Dinámica de Poblaciones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 219
4.11.2. MBL/Epidemiología Matemática (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 220
4.11.3. MBL/Biología de Sistemas y Dinámica de Redes (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 220
4.11.4. MBL/Algoritmos de Bioinformática (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 221
4.11.5. MBL/Neurociencia Matemática (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 221
4.12. Matemáticas Financieras y Actuariales (FAM) . . . 222
4.12.1. FAM/Teoría del Interés y Valor del Dinero en el Tiempo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 222
4.12.2. FAM/Valoración Libre de Arbitraje y Modelos de Mercado (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 222
4.12.3. FAM/Modelo de Black-Scholes y Valoración de Opciones (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 223
4.12.4. FAM/Medidas de Riesgo y Teoría de Portafolios (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 223
4.12.5. FAM/Modelos Actuariales y Tablas de Mortalidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 224
4.12.6. FAM/Finanzas Computacionales y Métodos de Monte Carlo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 224

5. Cuerpo del conocimiento de Física **225**
5.1. Mecánica Clásica (CME) . . . 226
5.1.1. CME/Cinemática y Dinámica de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr) 227
5.1.2. CME/Trabajo, Energía y Fuerzas Conservativas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 227
5.1.3. CME/Sistemas de Partículas y Momento Lineal (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 228
5.1.4. CME/Movimiento Rotacional y Momento Angular (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 229
5.1.5. CME/Dinámica del Cuerpo Rígido en 3D (Core Tier1: 1 hr, Core Tier2: 1 hr) . 229
5.1.6. CME/Marcos de Referencia no Inerciales y Fuerzas Ficticias (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 230
5.1.7. CME/Mecánica Lagrangiana y Hamiltoniana (Core Tier1: 1 hr, Core Tier2: 1 hr)230
5.1.8. CME/Movimiento de Fuerzas Centrales y Mecánica Orbital (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 231
5.1.9. CME/Teoría de Pequeñas Oscilaciones y Osciladores Acoplados (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 232
5.1.10. CME/Fundamentos de la Mecánica de Medios Continuos . . . 232
5.1.11. CME/Elasticidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 233
5.1.12. CME/Fluidos: Hidrostática, Hidrodinámica y Viscosidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 233
5.1.13. CME/Vibraciones y Ondas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 234

5.1.14. CME/Fenómeno de Transporte (Core Tier1: 1 hr, Core Tier2: 1 hr) 234
5.2. Electromagnetismo (ELM) 235
5.2.1. ELM/Electrostática: Campo Eléctrico y Ley de Gauss (Core Tier1: 1 hr, Core Tier2: 1 hr) 235
5.2.2. ELM/Potencial Eléctrico y Ecuaciones de Poisson/Laplace (Core Tier1: 1 hr, Core Tier2: 1 hr) 236
5.2.3. ELM/Energía Electrostática y Dieléctricos (Core Tier1: 1 hr, Core Tier2: 1 hr) 236
5.2.4. ELM/Magnetostática: Leyes de Biot-Savart y Ampère (Core Tier1: 1 hr, Core Tier2: 1 hr) 237
5.2.5. ELM/Propiedades Magnéticas de la Materia (Core Tier1: 1 hr, Core Tier2: 1 hr) 237
5.2.6. ELM/Electrodinámica: Ley de Faraday e Inducción (Core Tier1: 1 hr, Core Tier2: 1 hr) 238
5.2.7. ELM/Ecuaciones de Maxwell y Transformaciones de Norma (Core Tier1: 1 hr, Core Tier2: 1 hr) 238
5.2.8. ELM/Ondas Electromagnéticas y Propagación en Medios (Core Tier1: 1 hr, Core Tier2: 1 hr) 239
5.2.9. ELM/Teoría de la Radiación y Radiación Dipolar (Core Tier1: 1 hr, Core Tier2: 1 hr) 239
5.2.10. ELM/Relatividad Especial en Electromagnetismo (Core Tier1: 1 hr, Core Tier2: 1 hr) 240
5.2.11. ELM/Corriente Eléctrica, Circuitos y Leyes de Kirchhoff (Core Tier1: 1 hr, Core Tier2: 1 hr) 240
5.3. Termodinámica y Mecánica Estadística (TSM) 240
5.3.1. TSM/Leyes de la Termodinámica y Funciones de Estado (Core Tier1: 1 hr, Core Tier2: 1 hr) 241
5.3.2. TSM/Teoría Cinética de los Gases (Core Tier1: 1 hr, Core Tier2: 1 hr) 241
5.3.3. TSM/Máquinas Térmicas, Entropía y la Segunda Ley (Core Tier1: 1 hr, Core Tier2: 1 hr) 242
5.3.4. TSM/Potenciales Termodinámicos y Relaciones de Maxwell (Core Tier1: 1 hr, Core Tier2: 1 hr) 242
5.3.5. TSM/Transiciones de Fase y Fenómenos Críticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 242
5.3.6. TSM/Colectivos Microcanónico, Canónico y Gran Canónico (Core Tier1: 1 hr, Core Tier2: 1 hr) 243
5.3.7. TSM/Estadística Cuántica: Fermi-Dirac y Bose-Einstein (Core Tier1: 1 hr, Core Tier2: 1 hr) 243
5.3.8. TSM/Radiación de Cuerpo Negro y Fonones (Core Tier1: 1 hr, Core Tier2: 1 hr) 244
5.3.9. TSM/Mecánica Estadística de No-Equilibrio (Core Tier1: 1 hr, Core Tier2: 1 hr) 244
5.3.10. TSM/Aplicaciones a Sistemas Biológicos y Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr) 244
5.4. Mecánica Cuántica (QME) 245
5.4.1. QME/Orígenes de la Teoría Cuántica y Dualidad Onda-Partícula (Core Tier1: 1 hr, Core Tier2: 1 hr) 245
5.4.2. QME/La Ecuación de Schrödinger y Funciones de Onda (Core Tier1: 1 hr, Core Tier2: 1 hr) 245
5.4.3. QME/Formalismo: Espacio de Hilbert y Notación de Dirac (Core Tier1: 1 hr, Core Tier2: 1 hr) 246
5.4.4. QME/Potenciales Unidimensionales (Pozos, Barreras, Tunelamiento) (Core Tier1: 1 hr, Core Tier2: 1 hr) 246
5.4.5. QME/El Oscilador Armónico Cuántico (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 247
5.4.6. QME/Momento Angular y Física del Espín (Core Tier1: 1 hr, Core Tier2: 1 hr) 247
5.4.7. QME/El átomo de Hidrógeno y Potenciales Centrales (Core Tier1: 1 hr, Core Tier2: 1 hr) 247
5.4.8. QME/Teoría de Perturbaciones Independiente y Dependiente del Tiempo (Core Tier1: 1 hr, Core Tier2: 1 hr) 248

5.4.9. QME/Partículas Idénticas y Simetría de Intercambio (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 248
5.4.10. QME/Teoría de la Dispersión y Secciones Eficaces (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 248
5.5. Relatividad y Cosmología (RCO) . . . 249
5.5.1. RCO/Relatividad Especial: Transformaciones de Lorentz y Cuadrivectores (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 249
5.5.2. RCO/Dinámica Relativista y Energía-Momento (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 250
5.5.3. RCO/Fundamentos de la Relatividad General: Principio de Equivalencia (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 250
5.5.4. RCO/Curvatura del Espacio-Tiempo y Ecuaciones de Campo de Einstein (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 250
5.5.5. RCO/Geometría de Schwarzschild y Agujeros Negros (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 251
5.5.6. RCO/Ondas Gravitacionales (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 251
5.5.7. RCO/Teoría del Big Bang y Ecuaciones de Friedmann (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 251
5.5.8. RCO/Fondo Cósmico de Microondas (CMB) (Core Tier1: 1 hr, Core Tier2: 1 hr) 252
5.5.9. RCO/Materia Oscura y Energía Oscura (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 252
5.5.10. RCO/Inflación del Universo Temprano y Nucleosíntesis (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 252
5.6. Física de la Materia Condensada (SSP) . . . 253
5.6.1. SSP/Estructuras Cristalinas y Redes Recíprocas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 253
5.6.2. SSP/Difracción de Rayos X y Dinámica de Red (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 253
5.6.3. SSP/Modelo de Electrones Libres y Teoría de Bandas de Sólidos (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 254
5.6.4. SSP/Semiconductores, Aislantes y Metales (Core Tier1: 1 hr, Core Tier2: 1 hr) 254
5.6.5. SSP/Propiedades Magnéticas: Para-, Dia- y Ferromagnetismo (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 254
5.6.6. SSP/Superconductividad: Teoría BCS y Alta Tc (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 255
5.6.7. SSP/Física de la Materia Blanda y Polímeros (Core Tier1: 1 hr, Core Tier2: 1 hr) 255
5.6.8. SSP/Nanoestructuras y Sistemas de Baja Dimensionalidad (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 255
5.6.9. SSP/Espintrónica y Aislantes Topológicos (Core Tier1: 1 hr, Core Tier2: 1 hr) 256
5.6.10. SSP/Sistemas de Electrones Fuertemente Correlacionados (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 256
5.7. Física Nuclear y de Partículas (NPP) . . . 256
5.7.1. NPP/Estructura Nuclear, Energía de Enlace y Modelo de Capas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 257
5.7.2. NPP/Radiactividad y Desintegraciones Nucleares (Alfa, Beta, Gamma) (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 257
5.7.3. NPP/Reacciones Nucleares, Fisión y Fusión (Core Tier1: 1 hr, Core Tier2: 1 hr) 258
5.7.4. NPP/Aceleradores y Detectores de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr) 258
5.7.5. NPP/El Modelo Estándar de la Física de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 258
5.7.6. NPP/Quarks, Leptones y Bosones de Norma (Core Tier1: 1 hr, Core Tier2: 1 hr) 259
5.7.7. NPP/Simetrías y Leyes de Conservación (CPT) (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 259
5.7.8. NPP/Interacciones Fundamentales y Diagramas de Feynman (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 259
5.7.9. NPP/Mecanismo de Higgs y más allá del Modelo Estándar (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 260

5.7.10. NPP/Física de Neutrinos y Oscilación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 260
5.8. Información y Computación Cuántica (QIC) 260
5.8.1. QIC/Qubits, Superposición y Entrelazamiento (Core Tier1: 1 hr, Core Tier2: 1 hr) 261
5.8.2. QIC/Comuertas y Circuitos Lógicos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 261
5.8.3. QIC/Algoritmos Cuánticos (Shor, Grover) (Core Tier1: 1 hr, Core Tier2: 1 hr) 262
5.8.4. QIC/Teletransportación y Criptografía Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr) 262
5.8.5. QIC/Decoherencia y Corrección de Errores Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 262
5.8.6. QIC/Realizaciones Físicas de Computadoras Cuánticas (Core Tier1: 1 hr, Core Tier2: 1 hr) 263
5.8.7. QIC/Sensores Cuánticos y Metrología (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 263
5.8.8. QIC/Redes Cuánticas y Comunicación (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 263
5.9. Óptica y Fotónica (OPH) 264
5.9.1. OPH/Óptica Geométrica e Instrumentos Ópticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 264
5.9.2. OPH/Óptica Ondulatoria: Interferencia y Difracción (Core Tier1: 1 hr, Core Tier2: 1 hr) 265
5.9.3. OPH/Polarización y Óptica de Cristales (Core Tier1: 1 hr, Core Tier2: 1 hr) . 265
5.9.4. OPH/Física de Láseres: Principios y Aplicaciones (Core Tier1: 1 hr, Core Tier2: 1 hr) 265
5.9.5. OPH/Óptica No Lineal y Conversión de Frecuencia (Core Tier1: 1 hr, Core Tier2: 1 hr) 266
5.9.6. OPH/Fibra Óptica y Cristales Fotónicos (Core Tier1: 1 hr, Core Tier2: 1 hr) . 266
5.9.7. OPH/Óptica Ultrarrápida y Espectroscopía (Core Tier1: 1 hr, Core Tier2: 1 hr) 266
5.9.8. OPH/Nano-óptica y Plasmónica (Core Tier1: 1 hr, Core Tier2: 1 hr) 267
5.10. Física Computacional y Basada en Datos (CDP) 267
5.10.1. CDP/Métodos Numéricos para Ecuaciones Diferenciales en Física (Core Tier1: 1 hr, Core Tier2: 1 hr) 268
5.10.2. CDP/Métodos y Simulaciones de Monte Carlo (Core Tier1: 1 hr, Core Tier2: 1 hr) 268
5.10.3. CDP/Computación de Alto Rendimiento (HPC) para Simulaciones a Gran Escala (Core Tier1: 1 hr, Core Tier2: 1 hr) 268
5.10.4. CDP/Aprendizaje Automático para el Descubrimiento Físico (Core Tier1: 1 hr, Core Tier2: 1 hr) 269
5.10.5. CDP/Análisis de Datos y Cuantificación de la Incertidumbre (Core Tier1: 1 hr, Core Tier2: 1 hr) 269
5.10.6. CDP/Modelado de Sistemas Adaptativos Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr) 269
5.10.7. CDP/Redes Neuronales para Sistemas Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr) 270
5.10.8. CDP/Redes Neuronales Informadas por la Física (PINNs) (Core Tier1: 1 hr, Core Tier2: 1 hr) 270
5.11. Biofísica y Física Médica (BMP) 271
5.11.1. BMP/Física de las Macromoléculas Biológicas (Core Tier1: 1 hr, Core Tier2: 1 hr) 271
5.11.2. BMP/Biomecánica y Fuerzas Celulares (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 271
5.11.3. BMP/Física Neuronal y Procesamiento de Información en el Cerebro (Core Tier1: 1 hr, Core Tier2: 1 hr) 272
5.11.4. BMP/Física de Imágenes (MRI, CT, PET, Ultrasonido) (Core Tier1: 1 hr, Core Tier2: 1 hr) 272
5.11.5. BMP/Radioterapia y Dosimetría (Core Tier1: 1 hr, Core Tier2: 1 hr) 272

5.11.6. BMP/Mecánica Estadística de Redes Biológicas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 273
5.11.7. BMP/Microfluídica y Lab-on-a-Chip (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 273
5.12. Astrofísica y Ciencias del Espacio (ASS) . . . 273
5.12.1. ASS/Estructura y Evolución Estelar (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 274
5.12.2. ASS/Ciencia Planetaria y Exoplanetas (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 274
5.12.3. ASS/Dinámica y Estructura Galáctica (Core Tier1: 1 hr, Core Tier2: 1 hr) . . 274
5.12.4. ASS/Astrofísica de Altas Energías (Rayos X, Rayos Gamma) (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 275
5.12.5. ASS/Técnicas Observacionales y Telescopios (Core Tier1: 1 hr, Core Tier2: 1 hr) 275
5.12.6. ASS/Física Solar y Clima Espacial (Core Tier1: 1 hr, Core Tier2: 1 hr) 275
5.12.7. ASS/Astronomía de Multimensajeros (Ondas Gravitacionales + Neutrinos) (Core Tier1: 1 hr, Core Tier2: 1 hr) . . . 276

6. Plan de estudios **277**
6.1. Codificación de los cursos . . . 277
6.2. Estructura Curricular . . . 278
6.3. Distribución de cursos en la carrera . . . 283

7. Contenido detallado por curso **287**
Primer Semestre . . . 287
CS111 Introducción a la Programación (Obligatorio) . . . 287
MA100 Matemática Básica (Obligatorio) . . . 292
FG101 Lenguaje (Obligatorio) . . . 295
FG105 Género y Sociedad (Obligatorio) . . . 298
FG109 Ambiente y Desarrollo Sostenible (Obligatorio) . . . 301
FG112 Métodos de gestión personal universitaria (Obligatorio) . . . 304
FG140 Fundamentos de Investigación Científica (Obligatorio) . . . 307
Segundo Semestre . . . 310
CS100 Introducción a la Ciencia de la Computación (Obligatorio) . . . 310
CS112 Programación Orientada a Objetos I (Obligatorio) . . . 321
AI161 IA Aplicada (Obligatorio) . . . 326
MA111 Cálculo I (Obligatorio) . . . 331
FI101 Física General (Obligatorio) . . . 333
Tercer Semestre . . . 336
CS113 Programación Orientada a Objetos II (Obligatorio) . . . 336
CS1D1 Estructuras Discretas I (Obligatorio) . . . 342
CS2B1 Desarrollo Basado en Plataformas (Obligatorio) . . . 345
MA112 Cálculo II (Obligatorio) . . . 350
MA121 Algebra Lineal I (Obligatorio) . . . 352
ID101 Inglés I (Obligatorio) . . . 355
Cuarto Semestre . . . 359
CS1D2 Estructuras Discretas II (Obligatorio) . . . 359
CS210 Algoritmos y Estructuras de Datos (Obligatorio) . . . 361
CS271 Bases de Datos I (Obligatorio) . . . 366
MA122 Algebra Lineal II (Obligatorio) . . . 371
MA210 Cálculo III (Obligatorio) . . . 374
ID102 Inglés II (Obligatorio) . . . 377
Quinto Semestre . . . 381
CS251 Computación Gráfica (Obligatorio) . . . 381
MA211 Cálculo IV (Obligatorio) . . . 384
MA221 Análisis Numérico I (Obligatorio) . . . 387
MA222 Análisis Real I (Obligatorio) . . . 390
MA251 Introducción a la Estadística (Obligatorio) . . . 393
MA281 Teoría de números (Obligatorio) . . . 395
ID201 Inglés III (Obligatorio) . . . 399

Sexto Semestre . . . 402
MA212 Ecuaciones Diferenciales Ordinarias (Obligatorio) . . . 402
MA325 Análisis Numérico II (Obligatorio) . . . 405
MA351 Estadística y Probabilidades (Obligatorio) . . . 408
MA371 Análisis funcional I (Obligatorio) . . . 414
ID202 Inglés IV (Obligatorio) . . . 417
Séptimo Semestre . . . 421
CS261 Inteligencia Artificial (Obligatorio) . . . 421
CS3P1 Computación Paralela y Distribuída (Obligatorio) . . . 428
MA213 Ecuaciones Diferenciales Parciales (Obligatorio) . . . 438
MA326 Análisis Numérico III (Obligatorio) . . . 441
MA361 Modelado Matemático (Obligatorio) . . . 444
ID203 Inglés V (Obligatorio) . . . 448
Octavo Semestre . . . 451
AI263 Introducción al Aprendizaje de Máquina (Obligatorio) . . . 451
MA401 Proyecto de final de carrera I (Obligatorio) . . . 454
CS2S1 Sistemas Operativos (Electivo) . . . 457
CS3I1 Seguridad en Computación (Electivo) . . . 462
QC251 Mecánica Cuántica (Electivo) . . . 468
CS253 Análisis exploratorio de datos (Electivo) . . . 472
CS370 Big Data y Analítica de Datos (Electivo) . . . 476
STA252 Estadística Inferencial (Electivo) . . . 480
MA327 Análisis Numérico IV (Electivo) . . . 483
MA352 Procesos estocásticos (Electivo) . . . 486
MA372 Análisis funcional II (Electivo) . . . 489
Noveno Semestre . . . 492
AI264 Aprendizaje Profundo (Obligatorio) . . . 492
MA402 Proyecto de final de carrera II (Obligatorio) . . . 495
CS231 Redes y Comunicación (Electivo) . . . 499
MA381 Estructuras Algebraicas (Electivo) . . . 504
AI262 Procesamiento de Lenguaje Natural (Electivo) . . . 508
CS273 Análisis de series temporales (Electivo) . . . 511
MA341 Optimización I (Electivo) . . . 515
MA355 Teoría de Aprendizaje Estadístico (Electivo) . . . 518
Décimo Semestre . . . 521
CS400 Prácticas Pre-profesionales (Obligatorio) . . . 521
AI268 Visión Computacional (Obligatorio) . . . 523
AI367 Tópicos en Inteligencia Artificial (Obligatorio) . . . 529
MA403 Proyecto de final de carrera III (Obligatorio) . . . 536
QC381 Introduction a la Computación Cuántica (Electivo) . . . 540
CS254 Visualización de Datos Avanzada (Electivo) . . . 544

8. Profesores & Cursos **549**
Cursos asignados por profesor . . . 549
Profesor asignado por curso . . . 549

9. Equivalencias con otros planes curriculares **553**
Equivalencia del Plan 2026 al 2018 . . . 553
Equivalencia del Plan 2018 al 2026 . . . 557

10. Laboratorios **561**

Índice de figuras

6.1. Esquema de codificación para los cursos. 277
6.2. Créditos por área por semestre 284
6.3. Distribución de cursos por áreas considerando creditaje. 285
6.4. Distribución de créditos por niveles de cursos. 285

7.1. Mapa de Conexión. CS111 Introducción a la Programación 287
7.2. Mapa de Conexión. MA100 Matemática Básica 292
7.3. Mapa de Conexión. FG101 Lenguaje 295
7.4. Mapa de Conexión. FG105 Género y Sociedad 298
7.5. Mapa de Conexión. FG109 Ambiente y Desarrollo Sostenible 301
7.6. Mapa de Conexión. FG112 Métodos de gestión personal universitaria 304
7.7. Mapa de Conexión. FG140 Fundamentos de Investigación Científica 307
7.8. Mapa de Conexión. CS100 Introducción a la Ciencia de la Computación 310
7.9. Mapa de Conexión. CS112 Programación Orientada a Objetos I 321
7.10. Mapa de Conexión. AI161 IA Aplicada 326
7.11. Mapa de Conexión. MA111 Cálculo I 331
7.12. Mapa de Conexión. FI101 Física General 333
7.13. Mapa de Conexión. CS113 Programación Orientada a Objetos II 336
7.14. Mapa de Conexión. CS1D1 Estructuras Discretas I 342
7.15. Mapa de Conexión. CS2B1 Desarrollo Basado en Plataformas 345
7.16. Mapa de Conexión. MA112 Cálculo II 350
7.17. Mapa de Conexión. MA121 Algebra Lineal I 352
7.18. Mapa de Conexión. ID101 Inglés I 355
7.19. Mapa de Conexión. CS1D2 Estructuras Discretas II 359
7.20. Mapa de Conexión. CS210 Algoritmos y Estructuras de Datos 361
7.21. Mapa de Conexión. CS271 Bases de Datos I 366
7.22. Mapa de Conexión. MA122 Algebra Lineal II 371
7.23. Mapa de Conexión. MA210 Cálculo III 374
7.24. Mapa de Conexión. ID102 Inglés II 377
7.25. Mapa de Conexión. CS251 Computación Gráfica 381
7.26. Mapa de Conexión. MA211 Cálculo IV 384
7.27. Mapa de Conexión. MA221 Análisis Numérico I 387
7.28. Mapa de Conexión. MA222 Análisis Real I 390
7.29. Mapa de Conexión. MA251 Introducción a la Estadística 393
7.30. Mapa de Conexión. MA281 Teoría de números 395
7.31. Mapa de Conexión. ID201 Inglés III 399
7.32. Mapa de Conexión. MA212 Ecuaciones Diferenciales Ordinarias 402
7.33. Mapa de Conexión. MA325 Análisis Numérico II 405
7.34. Mapa de Conexión. MA351 Estadística y Probabilidades 408
7.35. Mapa de Conexión. MA371 Análisis funcional I 414
7.36. Mapa de Conexión. ID202 Inglés IV 417
7.37. Mapa de Conexión. CS261 Inteligencia Artificial 421
7.38. Mapa de Conexión. CS3P1 Computación Paralela y Distribuída 428
7.39. Mapa de Conexión. MA213 Ecuaciones Diferenciales Parciales 438

7.40. Mapa de Conexión. MA326 Análisis Numérico III 441
7.41. Mapa de Conexión. MA361 Modelado Matemático 444
7.42. Mapa de Conexión. ID203 Inglés V 448
7.43. Mapa de Conexión. AI263 Introducción al Aprendizaje de Máquina 451
7.44. Mapa de Conexión. MA401 Proyecto de final de carrera I 454
7.45. Mapa de Conexión. CS2S1 Sistemas Operativos 457
7.46. Mapa de Conexión. CS3I1 Seguridad en Computación 462
7.47. Mapa de Conexión. QC251 Mecánica Cuántica 468
7.48. Mapa de Conexión. CS253 Análisis exploratorio de datos 472
7.49. Mapa de Conexión. CS370 Big Data y Analítica de Datos 476
7.50. Mapa de Conexión. STA252 Estadística Inferencial 480
7.51. Mapa de Conexión. MA327 Análisis Numérico IV 483
7.52. Mapa de Conexión. MA352 Procesos estocásticos 486
7.53. Mapa de Conexión. MA372 Análisis funcional II 489
7.54. Mapa de Conexión. AI264 Aprendizaje Profundo 492
7.55. Mapa de Conexión. MA402 Proyecto de final de carrera II 495
7.56. Mapa de Conexión. CS231 Redes y Comunicación 499
7.57. Mapa de Conexión. MA381 Estructuras Algebraicas 504
7.58. Mapa de Conexión. AI262 Procesamiento de Lenguaje Natural 508
7.59. Mapa de Conexión. CS273 Análisis de series temporales 511
7.60. Mapa de Conexión. MA341 Optimización I 515
7.61. Mapa de Conexión. MA355 Teoría de Aprendizaje Estadístico 518
7.62. Mapa de Conexión. CS400 Prácticas Pre-profesionales 521
7.63. Mapa de Conexión. AI268 Visión Computacional 523
7.64. Mapa de Conexión. AI367 Tópicos en Inteligencia Artificial 529
7.65. Mapa de Conexión. MA403 Proyecto de final de carrera III 536
7.66. Mapa de Conexión. QC381 Introduction a la Computación Cuántica 540
7.67. Mapa de Conexión. CS254 Visualización de Datos Avanzada 544

Índice de cuadros

2.1. Lista de KUs del área de Análisis Numérico y Científico. 10
2.2. Lista de KUs del área de Modelado Matemático y Simulación. 12
2.3. Lista de KUs del área de Probabilidad y Computación Estadística. 14
2.4. Lista de KUs del área de Estructuras de Datos y Algoritmos. 16
2.5. Lista de KUs del área de Computación de Alto Rendimiento. 18
2.6. Lista de KUs del área de Optimización y Aprendizaje Automático. 20
2.7. Lista de KUs del área de Matemáticas para Ciencia de Datos y Aprendizaje Automático. 21
2.8. Lista de KUs del área de Matemáticas para Inteligencia Artificial. 23
2.9. Lista de KUs del área de Computación Simbólica y Discreta. 25
2.10. Lista de KUs del área de Señales, Imágenes y Visualización. 26
2.11. Lista de KUs del área de Computación Cuántica y Algoritmos. 28
2.12. Lista de KUs del área de Ciberseguridad y Privacidad. 29
2.13. Lista de KUs del área de Matemáticas Computacionales para Dominios Específicos. . 31
2.14. Lista de KUs del área de Gobernanza y Gestión de Datos. 33

3.1. Lista de KUs del área de Inteligencia Artificial. 38
3.2. Lista de KUs del área de Fundamentos Algorítmicos. 51
3.3. Lista de KUs del área de Arquitectura y Organización. 58
3.4. Lista de KUs del área de Gestión de Datos. 65
3.5. Lista de KUs del área de Fundamentos de los Lenguajes de Programación. 72
3.6. Lista de KUs del área de Gráficos y Técnicas Interactivas. 87
3.7. Lista de KUs del área de Interacción Humano-Computadora. 91
3.8. Lista de KUs del área de Fundamentos Matemáticos y Estadísticos. 94
3.9. Lista de KUs del área de Redes y Comunicaciones. 97
3.10. Lista de KUs del área de Sistemas Operativos. 101
3.11. Lista de KUs del área de Computación Paralela y Distribuida. 110
3.12. Lista de KUs del área de Fundamentos de Desarrollo de Software. 119
3.13. Lista de KUs del área de Ingeniería de Software. 123
3.14. Lista de KUs del área de Seguridad. 135
3.15. Lista de KUs del área de Sociedad, ética y la Profesión. 145
3.16. Lista de KUs del área de Fundamentos de Sistemas. 159
3.17. Lista de KUs del área de Desarrollo de Plataformas Especializadas. 167

4.1. Lista de KUs del área de Cálculo y Análisis. 182
4.2. Lista de KUs del área de álgebra y Teoría de Números. 190
4.3. Lista de KUs del área de Geometría y Topología. 194
4.4. Lista de KUs del área de Matemáticas Discretas y Combinatoria. 197
4.5. Lista de KUs del área de Probabilidad y Estadística. 200
4.6. Lista de KUs del área de Análisis Numérico y Científico. 203
4.7. Lista de KUs del área de Modelado y Simulación Matemática. 207
4.8. Lista de KUs del área de Fundamentos Matemáticos de la Ciencia de Datos. 211
4.9. Lista de KUs del área de Geometría y Topología Computacional. 214
4.10. Lista de KUs del área de Información y Computación Cuántica. 217
4.11. Lista de KUs del área de Biología Matemática y Ciencias de la Vida. 219

4.12. Lista de KUs del área de Matemáticas Financieras y Actuariales. 222

5.1. Lista de KUs del área de Mecánica Clásica. 226
5.2. Lista de KUs del área de Electromagnetismo. 235
5.3. Lista de KUs del área de Termodinámica y Mecánica Estadística. 241
5.4. Lista de KUs del área de Mecánica Cuántica. 245
5.5. Lista de KUs del área de Relatividad y Cosmología. 249
5.6. Lista de KUs del área de Física de la Materia Condensada. 253
5.7. Lista de KUs del área de Física Nuclear y de Partículas. 257
5.8. Lista de KUs del área de Información y Computación Cuántica. 261
5.9. Lista de KUs del área de Óptica y Fotónica. 264
5.10. Lista de KUs del área de Física Computacional y Basada en Datos. 267
5.11. Lista de KUs del área de Biofísica y Física Médica. 271
5.12. Lista de KUs del área de Astrofísica y Ciencias del Espacio. 274

6.1. Distribución de créditos por área por semestre 283

Agradecimientos

Además de los autores directos de este documento, también deseamos dejar manifiesto de nuestro agradecimiento a otros colegas de diversas universidades del país y del mundo que gentilmente han aportado parte de su tiempo a darnos sus sugerencias. Entre ellos debemos mencionar a:

- Todo de equipo de ACM/ IEEE-CS Computing Curricula (CC2020, CS2023 y CS2013).
- Los docentes que participaron en las Jornadas Curriculares (JC).
- A todo el personal administrativo por la organización las JC.
- A los estudiantes que participaron en las JC.
- A los colegas de otras universidades que nos han dado sus sugerencias.
- Sociedad Peruana de Computación por apoyarnos y facilitarnos su documento de referencia.

Todo este equipo de trabajo asumió como premisa que el centro de nuestro esfuerzo, es la formación académica y humana de los estudiantes para formar agentes de cambio positivo y disruptivo en la sociedad.

A todos ellos deseamos agradecerles por su aporte que ha permitido generar este documento, único en su género en nuestro país, que servirá para sentar las bases de una carrera más sólida en esta fantástica área que nos ha tocado estudiar y de la cual nos sentimos orgullosos de formar parte.

Abreviaturas

.

Capítulo 1

Introducción

La computación ha pasado por un desarrollo impresionante en las últimas décadas, convirtiéndose en el motor del desarrollo científico, tecnológico, industrial, social, económico y cultural, transformando de manera significativa nuestro diario accionar.

El surgimiento del computador ha marcado una nueva era en la historia de la humanidad que era imposible de imaginar varias décadas atrás. La gran cantidad de aplicaciones que se han desarrollado en los últimos años están transformando el desarrollo de todas las disciplinas del saber, la comercialización en el ámbito globalizado en que vivimos, la manera en que nos comunicamos, los procesos de enseñanza-aprendizaje y hasta en la manera como nos entretenemos.

Para darnos una idea de la relevancia e importancia, que en nuestro país ha alcanzado esta disciplina, basta mencionar que actualmente se ofrecen aproximadamente más de 100 carreras de Computación a nivel nacional. Esto sin considerar los programas de nivel Técnico Superior No Universitario que se ofertan.

Todas estas carreras existentes tienen como centro de su estudio a la computación pero lo hacen con más de 50 nombres distintos como: Ingeniería de Sistemas, Ingeniería de Computación, Ingeniería de Computación y Sistemas, entre otros. A pesar de que todas ellas apuntan al mismo mercado de trabajo, resulta por lo menos sorprendente que no sea posible encontrar por lo menos dos que compartan la misma curricula aunque sea en el primer año.

Comúnmente, durante la década de los setenta, la Computación se desarrolló dentro de las Facultades de Ciencias en la mayoría de las universidades estadounidenses, británicas y de otros países. Durante la década de los ochenta, los grupos de computación en las universidades se esforzaron por lograr una legitimidad académica en su ámbito local. Frecuentemente, se transformaron en departamentos de Matemáticas y Computación, hasta finalmente dividirse en dos departamentos de Matemáticas y de Computación, en la década de los noventa. Es en esta década en que un número creciente de instituciones reconocieron la influencia penetrante de la Computación, creando unidades independientes como departamentos, escuelas o institutos dedicados a tal área de estudio, un cambio que ha demostrado tanto perspicacia como previsión.

En Perú, un número cada vez mayor de instituciones de educación superior han tratado de seguir el desarrollo de las universidades extranjeras (aunque no siempre en forma muy seria o exitosa), reconociendo a la Computación como un área de estudio en sí misma, así como su importancia estratégica en la educación, y creando departamentos, escuelas o institutos dedicados a su estudio. La Facultad de Ciencias Matemáticas no puede ser la excepción a este cambio, en el que ya se tiene un retraso relativo con muchas de las instituciones educativas dentro y fuera de Perú.

1.1. Definiciones básicas

1.2. Campo y mercado ocupacional

Nuestro egresado podrá prestar sus servicios profesionales en empresas e instituciones públicas y privadas que requieran sus capacidades en función del desarrollo que oferta, entre ellas:

- Empresas dedicadas a la producción de software con calidad internacional.

- Empresas, instituciones y organizaciones que requieran software de calidad para mejorar sus actividades y/o servicios ofertados.

Nuestro egresado puede desempeñarse en el mercado laboral sin ningún problema ya que, en general, la exigencia del mercado y campo ocupacional está mucho más orientada al uso de herramientas. Sin embargo, es poco común que los propios profesionales de esta carrera se pregunten: ¿qué tipo de formación debería tener si yo quisiera crear esas herramientas además de saber usarlas?. Ambos perfiles (usuario y creador) son bastante diferentes pues no sería posible usar algo que todavía no fue creado. En otras palabras, los creadores de tecnología son los que dan origen a nuevos puestos de trabajo y abren la posibilidad de que otros puedan usar esa tecnología.

Debido a la formación basada en la investigación, nuestro profesional debe siempre ser un innovador donde trabaje. Esta misma formación permite que el egresado piense también en crear su propia empresa de desarrollo de software. Considerando que países en vias de desarrollo tienen un costo de vida mucho menor que Norte América ó Europa, una posibilidad que se muestra interesante es la exportación de software pero eso requiere que la calidad del producto sea al mismo nivel de lo ofrecido a nivel internacional.

Este perfil profesional también posibilita que los egresados se queden en el país; producir software en nuestro país y venderlo fuera es más rentable que salir al extranjero y comercializarlo allá.

El campo ocupacional de un egresado es amplio y está en continua expansión y cambio. Prácticamente toda empresa u organización hace uso de servicios de computación de algún tipo, y la buena formación básica de nuestros egresados hace que puedan responder a los requerimientos de las mismas exitosamente. Este egresado, no sólo podrá dar soluciones a los problemas existentes sino que deberá proponer innovaciones tecnológicas que impulsen la empresa hacia un progreso constante.

A medida que la informatización básica de las empresas del país avanza, la necesidad de personas capacitadas para resolver los problemas de mayor complejidad aumenta y el plan de estudios que hemos desarrollado tiene como objetivo satisfacer esta demanda considerandola a mediano y largo plazo. El campo para las tareas de investigación y desarrollo de problemas complejos en computación es también muy amplio y está creciendo día a día a nivel mundial.

Debido a la capacidad innovadora de nuestro egresado, existe una mayor la probabilidad de registrar patentes con un alto nivel inventivo lo cual es especialmente importante en nuestros países.

1.3. Importancia de la carrera en la sociedad

Uno de los caminos que se espera que siga un profesional del área de Computación Científica es que el se dedique a producir software o que se integre a las empresas productoras de software. En el ámbito de la computación, es común observar que los países cuentas con Asociaciones de Productores de Software cuyas políticas están orientadas a la exportación. Siendo así, no tendría sentido preparar a nuestros alumnos sólo para el mercado local o nacional. Nuestros egresados deben estar preparados para desenvolverse en el mundo globalizado que nos ha tocado vivir.

Nuestros futuros profesionales deben estar orientados a crear nuevas empresas de base tecnológica que puedan incrementar las exportaciones de software peruano. Este nuevo perfil está orientado a generar industria innovadora. Si nosotros somos capaces de exportar software competitivo también estaremos en condiciones de atraer nuevas inversiones. Las nuevas inversiones generarían más puestos de empleo bien remunerados y con un costo bajo en relación a otros tipos de industria. Bajo esta perspectiva, podemos afirmar que esta carrera será un motor que impulsará al desarrollo del país de forma decisiva con una inversión muy baja en relación a otros campos.

Es necesario recordar que la mayor innovación de productos comerciales de versiones recientes utiliza tecnología que se conocía en el mundo académico hace varios años, décadas o más. Un ejemplo claro son las bases de datos que soportan datos y consultas espaciales desde hace muy pocos años. Sin embargo, utilizan estructuras de datos que ya existían hace algunas décadas. Es lógico pensar que la gente del área académica no se dedique a estudiar en profundidad la última versión de un determinado software cuando esa tecnología ya la conocían hace mucho tiempo. Por esa misma razón es raro en el mundo observar que una universidad tenga convenios con una transnacional de software para dictar solamente esa tecnología pues, nuestra función es generar esa tecnología **y no sólo saber usarla**.

Tampoco debemos olvidar que los alumnos que ingresan hoy saldrán al mercado dentro de 5 años aproximadamente y, en un mundo que cambia tan rápido, no podemos ni debemos enseñarles tomando

en cuenta solamente el mercado actual. Nuestros profesionales deben estar preparados para resolver los problemas que habrá dentro de 10 o 15 años y eso sólo es posible a través de la investigación.

1.4. Información institutional

1.4.1. Misión

Por lo antes mencionado, nuestra misión es:

Ser el referente nacional integral de alto nivel en Matemática, Estadística, Computación Científica e Investigación Operativa, en los niveles de Posgrado y Pregrado, aportando a la preservacion de la diversidad cultural del Perú y a su progreso científico y tecnológico. Sus miembros se distinguen por la aplicación del metodo científico para el desarrollo del conocimiento y la solución de los problemas del país, respetando y protegiendo su medio ambiente.

1.4.2. Visión

- Queremos ser una carrera profesional acreditada con estándares internacionales que cuente con el reconocimiento en función de la calidad y competitividad de sus docentes y egresados.
- Queremos ser una carrera que trascienda por la relevancia y pertinencia de sus proyectos de investigación básica y aplicada.
- Buscamos ser una carrera que promueva el desarrollo de la industria del software relacionada a Computación Científica a nivel internacional, incorporando a sus egresados a la industria ya establecida o generando nuevas empresas en este mismo rubro.
- Queremos ser una carrera que comparta y difunda el conocimiento con todos los sectores de la población y contribuya a la solución de los problemas estratégicos de nuestra sociedad.

1.5. Información de acreditacion

1.5.1. Resultados de la carrera *Outcomes*

Al finalizar esta carrera, el(la) egresado(a), habrá logrado los siguientes resultados (*Outcomes*):

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina.

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto.

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones.

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias.

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales.

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre.

Las competencias y/o resultados propuestas en esta malla curricular corresponden a las competencias y/o resultados propuestas por ABET.

1.5.2. Objetivos educacionales

Después de cinco años de egresado de la carrera profesional de Computación Científica, nuestros profesionales deben ser capaces de:

1. Cumplir y superar las expectativas de trabajo definidas por el entorno laboral.
2. Desempeñarse como miembro, o líder, de un equipo de trabajo tanto especializado como multidisciplinario.
3. Proponer soluciones al contexto laboral, donde se desenvuelve, basadas en la implementación o mejora del estado del arte en Computación Científica.
4. Comunicar de forma efectiva propuestas tecnológicas a personas de distintos niveles de conocimiento y de diferentes ámbitos sociales.
5. Actualizarse y adaptarse a nuevos conocimientos en Computación Científica y a diferentes ámbitos laborales, de forma autónoma o mediante estudios complementarios.
6. Demostrar un claro entendimiento de las consecuencias que surgen a partir de creaciones tecnológicas relacionadas a la Computación Científica en aspectos tales como lo social, ético, humano, moral, legal, ambiental, económico, entre otros.

1.5.3. Aprendizaje esperado (*Learning Outcomes*)

Cada KU dentro de un KA enumera tanto un conjunto de temas como los resultados de aprendizaje (*Learning Outcomes*) que los estudiantes deben alcanzar en lo que respecta a los temas especificados. Los resultados de aprendizaje no son de igual tamaño y no tienen una asignación uniforme de horas curriculares; temas con el mismo número de horas pueden tener muy diferentes números de resultados de aprendizaje asociados.

Cada resultado de aprendizaje tiene un nivel asociado de dominio. En la definición de los diferentes niveles tomamos como referencia otros enfoques curriculares, especialmente la taxonomía de Bloom, que ha sido bien explorada dentro de la Computación Científica. En este documento no se aplicaron directamente los niveles de Bloom en parte porque varios de ellos están determinados por el contexto pedagógico, lo que introduciría demasiada pluralidad en un documento de este tipo; y en parte porque los niveles de dominio pretenden ser indicativos y no imponer restricciones teóricas sobre los usuarios de este documento.

Nosotros usamos tres niveles de dominio esperados, que son:

Nivel 1 Familiarizarse(*Familiarity*) : El estudiante **comprende** un concepto básicamente. Responde: **¿Qué sabe sobre esto?**

Nivel 2 Usar(*Usage*) : El estudiante **aplica** conceptos en situaciones prácticas (ej: programación). Responde: **¿Cómo lo haría?**

Nivel 3 Evaluar(*Assessment*) : El estudiante **evalúa y justifica** enfoques. Responde: **¿Por qué este método?**

Por ejemplo, para evaluar los niveles de dominio, consideremos la noción de derivada en el cálculo. En el plano de la "familiaridad", se espera que un estudiante conozca la definición de derivada, entienda su significado como tasa de cambio y sepa por qué es una herramienta fundamental en ciencias básicas.

Con el fin de mostrar el dominio del nivel "Uso", el estudiante debe ser capaz de calcular derivadas de funciones elementales, aplicar reglas de derivación y utilizarlas para resolver problemas sencillos de optimización o de razones de cambio.

En el nivel de "Evaluación", se requeriría que el estudiante comprenda las múltiples interpretaciones de la derivada (geométrica, física, económica, etc.), analice sus limitaciones y sea capaz de seleccionar el enfoque más adecuado para modelar y resolver fenómenos complejos en distintos contextos científicos.

1.6. Perfiles

1.6.1. Perfil del estudiante

El aspirante a ingresar a la Escuela Profesional de Computación Científica de la Universidad Nacional Mayor de San Marcos debe tener:

Conocimientos de:

1. Conceptos fundamentales de álgebra, geometría, trigonometría y análisis matemático elemental.
2. Fenómenos físicos y químicos básicos de su entorno cotidiano.
3. Nociones elementales de probabilidad y estadística descriptiva.

Habilidades para:

1. Identificar relaciones causa-efecto en sistemas naturales y abstractos, y predecir sus comportamientos.
2. Distinguir entre observaciones empíricas y modelos teóricos, reconociendo sus límites y aplicaciones.
3. Percibir y describir patrones cuantitativos y cualitativos en datos y fenómenos.
4. Plantear hipótesis, diseñar procedimientos de verificación e interpretar resultados de manera rigurosa.
5. Comunicar argumentos y conclusiones de forma clara, tanto oralmente como por escrito, utilizando lenguaje técnico apropiado.
6. Mantener la concentración en tareas analíticas que demanden esfuerzo sostenido.
7. Sintetizar información compleja y extraer conceptos esenciales.
8. Adquirir y aplicar metodologías de estudio sistemático y autorregulado.

Actitudes de:

1. Interés genuino y curiosidad por las ciencias exactas y naturales, y sus aplicaciones.
2. Disposición para el trabajo colaborativo en laboratorios, talleres y proyectos académicos.
3. Iniciativa para explorar problemas más allá del currículo y perseverancia ante desafíos intelectuales.

1.6.2. Perfil del graduado

El graduado del Programa Profesional de Computación Científica es capaz de:

1. Aplicar fundamentos teóricos de matemática pura (álgebra lineal, análisis matemático, teoría de números) para modelar problemas complejos en ciencia e ingeniería.
2. Diseñar e implementar algoritmos numéricos y simbólicos eficientes para resolver ecuaciones diferenciales, sistemas de optimización y problemas de álgebra computacional.
3. Desarrollar software matemático utilizando lenguajes de programación científica (Python, Julia, MATLAB, C++) y bibliotecas especializadas para computación de alto rendimiento.
4. Modelar y simular fenómenos utilizando métodos Monte Carlo, análisis de estabilidad, teoría de grafos y optimización combinatoria en aplicaciones reales.
5. Aplicar técnicas de machine learning, deep learning y análisis de datos masivos para extraer patrones, construir predictores y resolver problemas de clasificación y clustering.
6. Colaborar en proyectos interdisciplinarios de bioinformática, criptografía, finanzas cuantitativas, física computacional e inteligencia artificial, aportando rigor matemático.
7. Analizar la complejidad computacional de algoritmos, implementar técnicas de paralelización y optimizar el uso de recursos en clusters y arquitecturas GPU.
8. Comunicar resultados matemáticos y computacionales mediante publicaciones científicas, visualización de datos interactiva y documentación técnica rigurosa para diversas audiencias.
9. Adaptar herramientas matemáticas para resolver problemas industriales en sectores como telecomunicaciones, logística, ciberseguridad, minería de datos y desarrollo de videojuegos.
10. Continuar estudios de posgrado en matemática aplicada, ciencia de datos o computación científica, contribuyendo a la generación de conocimiento en áreas emergentes.

1.6.3. Perfil Profesional

El profesional en Computación Científica tiene las siguientes características:

- Aplicar teorías fundamentales de matemática pura y aplicada para modelar y resolver problemas complejos en ciencia e ingeniería.
- Diseñar e implementar algoritmos numéricos y simbólicos eficientes usando lenguajes científicos (Python, Julia, MATLAB, C++).
- Desarrollar modelos computacionales para simular fenómenos, resolver ecuaciones diferenciales y optimizar sistemas mediante métodos matemáticos avanzados.

- Aplicar técnicas de machine learning, deep learning y análisis de datos masivos para extraer patrones y construir predictores robustos.
- Trabajar con arquitecturas de computación de alto rendimiento (clusters, GPU) para resolver problemas de gran escala en tiempo razonable.
- Colaborar en proyectos interdisciplinarios en bioinformática, criptografía, finanzas cuantitativas, física computacional e inteligencia artificial.
- Comunicar resultados matemáticos y computacionales con rigor mediante publicaciones, visualizaciones interactivas y documentación técnica especializada.
- Analizar la complejidad computacional de algoritmos, implementar técnicas de paralelización y optimizar uso de recursos computacionales.
- Adaptar herramientas matemáticas para resolver desafíos en sectores como telecomunicaciones, minería, logística, ciberseguridad y desarrollo tecnológico.
- Continuar estudios de posgrado en matemática aplicada, ciencia de datos o computación científica, generando conocimiento original de impacto.

1.6.4. Perfil del docente

El docente de la carrera de Computación Científica es una persona que:

- Promueve la formación integral del estudiante mediante la enseñanza, investigación y extensión, transmitiendo conocimientos sólidos con claridad y simplicidad.
- Fomenta la investigación y el aprendizaje continuo para que los estudiantes se adapten a los avances del área y busquen el bien común de individuos, sociedad y organizaciones.
- Mantiene formación académica avanzada de calidad, con estudios a tiempo completo, buscando la excelencia académica a nivel internacional.
- Publica investigaciones en eventos internacionales y revistas indexadas de alto impacto, contribuyendo al avance del conocimiento y manteniéndose actualizado.
- Establece redes de cooperación con instituciones de nivel internacional, enriqueciendo la experiencia educativa.
- Participa activamente en la formación e integración del estudiante a la comunidad académica , promoviendo la exigencia continua y rigor académico.

1.7. Grados y Títulos

Cumplida la aprobación de los créditos académicos propuestos en la malla curricular y luego de haber cumplido con los requisitos estipulados en el reglamento de grados y títulos de la carrera, el egresado recibirá:

Grado Académico: Bachiller en Computación Científica y

Titulo Profesional: Licenciado en Computación Científica

1.8. Recursos para dictado de clases

Un profesional innovador debe estar al tanto de los últimos avances de su área siempre. Los últimos avances de esta área no son presentados en los libros necesariamente. Debemos utilizar publicaciones de revistas indexadas de circulación mundial.

Por esa razón, tomamos como base la suscripción institucional a la ACM y a la IEEE-CS. Es recomendado que el docente use este material para discutir en clase las tendencias en todas las áreas.

Los laboratorios de cómputo también deben ser renovados de acuerdo a la rapidez propia de esta área y esto implica renovaciones constantes para poder garantizar que los estudiantes estén actualizados.

El recurso físico en cuanto a laboratorios para cada curso está detallado en el Capítulo 10.

1.9. Organización del documento

El resto de este documento está organizado de la siguiente forma: el Capítulo 1, define y explica el campo de acción de la Computación Científica, además se hace una muy breve explicación de las distintas carreras relacionadas a esta disciplina. .

En el Capítulo 2, presentamos las áreas de Conocimiento de Matemática Computacional, indicando los tópicos y objetivos de aprendizaje de cada temas,

En el Capítulo 3, presentamos las áreas de Conocimiento de Ciencia de la Computación–*Computer Science*, indicando los tópicos y objetivos de aprendizaje de cada temas,

En el Capítulo 4, presentamos las áreas de Conocimiento de Matemática, indicando los tópicos y objetivos de aprendizaje de cada temas,

En el Capítulo 5, presentamos las áreas de Conocimiento de Física, indicando los tópicos y objetivos de aprendizaje de cada temas,

El Capítulo 6 contiene la distribución por semestres, por áreas, por niveles, visión gráfica de la malla curricular, comparación con las diversas propuestas internacionales, distribución de tópicos por curso así como la distribución de habilidades por materia.

El Capítulo 7 contiene información detallada para cada uno de los cursos incluyendo las habilidades con las cuales contribuye, bibliografía por cada unidad así como el número de horas mínimas por cada unidad.

En el Capítulo 8 podemos observar un listado de cursos por profesor y de profesor por curso.

En el Capítulo 9 se presentan las tablas de equivalencias con otros planes curriculares.

En el Capítulo 10 se presenta una sugerencia de los laboratorios requeridos para el dictado de clases las mismas que podrían variar de acuerdo al volumen de alumnos que se tenga.

Capítulo 2

Cuerpo del conocimiento de Matemática Computacional

Las 14 áreas de conocimiento en Matemática Computacional son:

2.1 Análisis Numérico y Científico (NSA) (Pág. 10)
- 2.1.1 Aritmética Computacional y Análisis de Error (Pág. 10)
- 2.1.2 álgebra Lineal Numérica (Pág. 10)
- 2.1.3 Búsqueda de Raíces y Optimización Numérica Básica (Pág. 11)
- 2.1.4 Interpolación y Aproximación de Funciones (Pág. 11)
- 2.1.5 Diferenciación e Integración Numérica (Pág. 11)
- 2.1.6 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (Pág. 12)

2.2 Modelado Matemático y Simulación (MMS) (Pág. 12)
- 2.2.1 Formulación de Modelos Matemáticos (Pág. 12)
- 2.2.2 Simulación de Sistemas Discretos (Pág. 13)
- 2.2.3 Simulación de Sistemas Continuos (Pág. 13)
- 2.2.4 Análisis de Sensibilidad y Robustez (Pág. 13)
- 2.2.5 Validación y Verificación de Modelos (Pág. 14)

2.3 Probabilidad y Computación Estadística (PSC) (Pág. 14)
- 2.3.1 Probabilidad Avanzada y Procesos Estocásticos (Pág. 14)
- 2.3.2 Inferencia Estadística Computacional (Pág. 15)
- 2.3.3 Diseño de Experimentos Computacionales (Pág. 15)
- 2.3.4 Métodos de Remuestreo y Validación (Pág. 15)

2.4 Estructuras de Datos y Algoritmos (DSA) (Pág. 16)
- 2.4.1 Complejidad Algorítmica y Análisis Asintótico (Pág. 16)
- 2.4.2 Estructuras de Datos Fundamentales y Avanzadas (Pág. 16)
- 2.4.3 Algoritmos de Búsqueda y Ordenamiento (Pág. 17)
- 2.4.4 Algoritmos de Grafos y Optimización Combinatoria (Pág. 17)
- 2.4.5 Algoritmos Paralelos y Distribuidos (Pág. 17)

2.5 Computación de Alto Rendimiento (HPC) (Pág. 17)
- 2.5.1 Arquitecturas de Computadoras Paralelas (Pág. 18)
- 2.5.2 Programación Paralela de Memoria Compartida (Pág. 18)
- 2.5.3 Programación Paralela de Memoria Distribuida (Pág. 18)
- 2.5.4 Programación de Aceleradores (Pág. 19)
- 2.5.5 Métodos Numéricos Paralelos (Pág. 19)

2.6 Optimización y Aprendizaje Automático (OML) (Pág. 19)
- 2.6.1 Optimización Convexa y No Convexa (Pág. 20)
- 2.6.2 Optimización con Restricciones (Pág. 20)
- 2.6.3 Optimización para Aprendizaje Automático (Pág. 20)
- 2.6.4 Optimización a Gran Escala y Distribuida (Pág. 21)
- 2.6.5 Metaheurísticas y Optimización Bioinspirada (Pág. 21)

2.7 Matemáticas para Ciencia de Datos y Aprendizaje Automático (DSM) (Pág. 21)
- 2.7.1 Fundamentos del Aprendizaje Estadístico (Pág. 22)
- 2.7.2 Métodos de Aprendizaje Supervisado (Pág. 22)
- 2.7.3 Métodos de Aprendizaje No Supervisado (Pág. 22)
- 2.7.4 Evaluación y Selección de Modelos (Pág. 22)

2.8 Matemáticas para Inteligencia Artificial (AIM) (Pág. 23)
- 2.8.1 Teoría del Aprendizaje Profundo (Pág. 23)
- 2.8.2 Arquitecturas Neuronales Avanzadas (Pág. 23)
- 2.8.3 Aprendizaje por Refuerzo (Pág. 24)
- 2.8.4 Procesamiento de Lenguaje Natural con Transformers (Pág. 24)
- 2.8.5 Modelos Generativos y GANs (Pág. 24)

2.9 Computación Simbólica y Discreta (SDC) (Pág. 25)
- 2.9.1 Teoría de Grupos y Algoritmos Algebraicos (Pág. 25)
- 2.9.2 Teoría de Números Computacional y Criptografía (Pág. 25)
- 2.9.3 Sistemas de Computación Simbólica (Pág. 25)
- 2.9.4 Lógica Computacional y Verificación Formal (Pág. 26)

2.10 Señales, Imágenes y Visualización (SIV) (Pág. 26)
- 2.10.1 Análisis de Fourier y Wavelets (Pág. 26)
- 2.10.2 Procesamiento Digital de Imágenes y Visión Computacional (Pág. 27)
- 2.10.3 Visualización Científica y Computación Visual (Pág. 27)
- 2.10.4 Gráficos por Computadora y Geometría Computacional (Pág. 27)

2.11 Computación Cuántica y Algoritmos (QCA) (Pág. 28)
- 2.11.1 Fundamentos de Mecánica Cuántica para Computación (Pág. 28)
- 2.11.2 Qubits y Circuitos Cuánticos (Pág. 28)
- 2.11.3 Algoritmos Cuánticos (Pág. 28)
- 2.11.4 Criptografía Post-Cuántica y Simulación Cuántica (Pág. 29)

2.12 Ciberseguridad y Privacidad (CSP) (Pág. 29)
- 2.12.1 Criptografía Simétrica (Pág. 29)
- 2.12.2 Criptografía Asimétrica (Pág. 30)
- 2.12.3 Computación con Preservación de la Privacidad (Pág. 30)
- 2.12.4 Teoría de la Información y Compresión de Datos (Pág. 30)

2.13 Matemáticas Computacionales para Dominios Específicos (CDM) (Pág. 31)
- 2.13.1 Bioinformática y Biología Computacional (Pág. 31)
- 2.13.2 Finanzas Computacionales y Ciencia del Riesgo (Pág. 31)
- 2.13.3 Física Computacional y Química Cuántica (Pág. 32)
- 2.13.4 Ciencia de Materiales Computacional (Pág. 32)
- 2.13.5 Modelado Climático y Sistemas Complejos (Pág. 32)

2.14 Gobernanza y Gestión de Datos (DGM) (Pág. 32)

- 2.14.1 ética en IA y Algoritmos Responsables (Pág. 33)
- 2.14.2 Gestión de Proyectos de Datos y MLOps (Pág. 33)
- 2.14.3 Gobernanza de Datos y Cumplimiento Normativo (Pág. 33)

2.1. Análisis Numérico y Científico (NSA)

Esta área cubre los métodos numéricos fundamentales y su análisis que forman la base de las matemáticas computacionales. Se enfoca en el desarrollo, implementación y análisis de algoritmos para resolver problemas matemáticos numéricamente, con un fuerte énfasis en comprender y controlar los errores que surgen de la aritmética de precisión finita y las aproximaciones discretas. Este conocimiento es esencial para todas las áreas posteriores en matemáticas computacionales, ya que proporciona las herramientas para garantizar que las soluciones calculadas sean precisas, estables y confiables.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.1.1 Aritmética Computacional y Análisis de Error (Pág. 10)	1	1
⭐ 2.1.2 álgebra Lineal Numérica (Pág. 10)	1	1
⭐ 2.1.3 Búsqueda de Raíces y Optimización Numérica Básica (Pág. 11)	1	1
⭐ 2.1.4 Interpolación y Aproximación de Funciones (Pág. 11)	1	1
⭐ 2.1.5 Diferenciación e Integración Numérica (Pág. 11)	1	1
⭐ 2.1.6 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (Pág. 12)	1	1

Cuadro 2.1: Lista de KUs del área de Análisis Numérico y Científico.

2.1.1. NSA/Aritmética Computacional y Análisis de Error (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Aritmética de punto flotante y el estándar IEEE
- Fuentes de error: truncamiento, redondeo y discretización
- Propagación de error y número de condición
- Estabilidad de algoritmos numéricos
- Error absoluto, relativo y directo/inverso

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse]
2. Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse]
3. Calcular el error absoluto y relativo de una aproximación [Usar]
4. Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar]
5. Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar]
6. Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

2.1.2. NSA/álgebra Lineal Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Descomposición LU y sus aplicaciones
- Descomposición QR y problemas de mínimos cuadrados
- Descomposición en Valores Singulares (SVD) y sus aplicaciones
- Número de condición y estabilidad numérica de sistemas lineales
- Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar]
2. Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse]
3. Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar]
4. Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar]
5. Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar]

2.1.3. NSA/Búsqueda de Raíces y Optimización Numérica Básica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Método de bisección e iteración de punto fijo
- Método de Newton y método de la secante para búsqueda de raíces
- Análisis de convergencia y orden de convergencia
- Descenso de gradiente para optimización univariada
- Tasa de convergencia y criterios de parada

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar]
2. Clasificar el orden de convergencia de métodos iterativos [Evaluar]
3. Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar]
4. Diseñar criterios de parada apropiados para métodos iterativos [Usar]

2.1.4. NSA/Interpolación y Aproximación de Funciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Interpolación polinomial (formas de Lagrange, Newton)
- Interpolación por splines (splines cúbicos)
- Aproximación por mínimos cuadrados
- Fenómeno de Runge y nodos de Chebyshev
- Análisis de error para interpolación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Construir polinomios interpoladores usando diferentes métodos [Usar]
2. Implementar interpolación por splines cúbicos [Usar]
3. Aplicar el método de mínimos cuadrados para ajuste de datos [Usar]
4. Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse]
5. Analizar el error de un esquema de interpolación [Evaluar]

2.1.5. NSA/Diferenciación e Integración Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas)
- Extrapolación de Richardson
- Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson)
- Cuadratura gaussiana
- Análisis de error para integración numérica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar]
2. Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar]
3. Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar]
4. Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar]
5. Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

2.1.6. NSA/Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Método de Euler y sus variantes (explícito, implícito)
- Métodos de Runge-Kutta
- Métodos multipaso (Adams-Bashforth, Adams-Moulton)
- Análisis de estabilidad y convergencia para solucionadores de EDOs
- Sistemas de EDOs y EDOs de orden superior

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar]
2. Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar]
3. Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar]
4. Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar]

2.2. Modelado Matemático y Simulación (MMS)

Esta área se enfoca en el proceso de crear, analizar y validar modelos matemáticos para representar sistemas del mundo real. Cubre tanto paradigmas de modelado continuo como discreto, así como enfoques deterministas y estocásticos. Los estudiantes aprenden a formular modelos, implementar simulaciones y evaluar críticamente los resultados mediante análisis de sensibilidad y cuantificación de incertidumbre. Este conocimiento es esencial para aplicar matemáticas computacionales a problemas complejos en ciencia, ingeniería y otros dominios.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.2.1 Formulación de Modelos Matemáticos (Pág. 12)	1	1
⭐ 2.2.2 Simulación de Sistemas Discretos (Pág. 13)	1	1
⭐ 2.2.3 Simulación de Sistemas Continuos (Pág. 13)	1	1
⭐ 2.2.4 Análisis de Sensibilidad y Robustez (Pág. 13)	1	1
⭐ 2.2.5 Validación y Verificación de Modelos (Pág. 14)	1	1

Cuadro 2.2: Lista de KUs del área de Modelado Matemático y Simulación.

2.2.1. MMS/Formulación de Modelos Matemáticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Proceso y ciclo de modelado
- Modelos continuos vs. discretos
- Modelos deterministas vs. estocásticos

- Identificación de supuestos y técnicas de simplificación
- Análisis dimensional y escalado

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir el ciclo completo de modelado desde la definición del problema hasta la interpretación de la solución [Familiarizarse]
2. Clasificar modelos como continuos/discretos y deterministas/estocásticos [Evaluar]
3. Identificar y justificar supuestos apropiados para simplificación de modelos [Evaluar]
4. Aplicar análisis dimensional para reducir la complejidad de modelos físicos [Usar]

2.2.2. MMS/Simulación de Sistemas Discretos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Teoría de colas y modelos
- Autómatas celulares y modelos basados en agentes
- Métodos de simulación Monte Carlo
- Cadenas y procesos de Markov
- Simulación de eventos discretos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Formular y analizar modelos simples de colas [Usar]
2. Implementar autómatas celulares y simulaciones basadas en agentes [Usar]
3. Aplicar métodos Monte Carlo para simulación probabilística [Usar]
4. Analizar cadenas de Markov para transiciones de estado y comportamiento en estado estacionario [Evaluar]
5. Diseñar simulaciones de eventos discretos para análisis de sistemas [Usar]

2.2.3. MMS/Simulación de Sistemas Continuos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelos de Ecuaciones Diferenciales Ordinarias (EDOs)
- Modelos de Ecuaciones Diferenciales Parciales (EDPs)
- Métodos de diferencias finitas para EDPs
- Selección e implementación de solucionadores
- Sistemas stiff y métodos adaptativos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Formular modelos de EDOs y EDPs para sistemas físicos [Usar]
2. Implementar métodos de diferencias finitas para EDPs simples [Usar]
3. Seleccionar solucionadores numéricos apropiados para diferentes tipos de EDOs/EDPs [Evaluar]
4. Manejar sistemas stiff usando métodos numéricos apropiados [Usar]

2.2.4. MMS/Análisis de Sensibilidad y Robustez (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Análisis de sensibilidad local (derivadas parciales)
- Análisis de sensibilidad global (índices de Sobol)
- Método de Morris y técnicas de screening
- Análisis de robustez y escenarios del peor caso
- Propagación de incertidumbre

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Realizar análisis de sensibilidad local usando métodos basados en derivadas [Usar]
2. Aplicar técnicas de análisis de sensibilidad global (ej., índices de Sobol) [Usar]
3. Usar el método de Morris para screening de factores [Usar]
4. Evaluar la robustez del modelo bajo variaciones de parámetros [Evaluar]
5. Cuantificar la propagación de incertidumbre a través de modelos [Evaluar]

2.2.5. MMS/Validación y Verificación de Modelos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Verificación: corrección del código numérico
- Validación: comparación con datos experimentales
- Cuantificación de Incertidumbre (UQ) en modelos
- Calibración de modelos y estimación de parámetros
- Pruebas y métricas de bondad de ajuste

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Distinguir entre verificación y validación en modelado [Familiarizarse]
2. Aplicar técnicas de verificación para garantizar la corrección del código [Usar]
3. Validar modelos usando datos experimentales y pruebas estadísticas [Usar]
4. Calibrar parámetros del modelo usando técnicas de optimización [Usar]
5. Aplicar métodos de cuantificación de incertidumbre para evaluar la confiabilidad del modelo [Usar]

2.3. Probabilidad y Computación Estadística (PSC)

Esta área conecta los fundamentos teóricos de probabilidad y estadística con su implementación computacional. Cubre modelado probabilístico avanzado, métodos de inferencia computacional y el diseño de experimentos in silico. Los estudiantes aprenden a aplicar procesos estocásticos, métodos bayesianos y técnicas de remuestreo para analizar sistemas complejos y tomar decisiones basadas en datos bajo incertidumbre. Este conocimiento es crucial para el análisis de datos moderno, el aprendizaje automático y la computación científica.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.3.1 Probabilidad Avanzada y Procesos Estocásticos (Pág. 14)	1	1
⭐ 2.3.2 Inferencia Estadística Computacional (Pág. 15)	1	1
⭐ 2.3.3 Diseño de Experimentos Computacionales (Pág. 15)	1	1
⭐ 2.3.4 Métodos de Remuestreo y Validación (Pág. 15)	1	1

Cuadro 2.3: Lista de KUs del área de Probabilidad y Computación Estadística.

2.3.1. PSC/Probabilidad Avanzada y Procesos Estocásticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Fundamentos de probabilidad medida-teórica
- Variables aleatorias y distribuciones
- Caracterización de procesos estocásticos
- Cadenas y procesos de Markov
- Movimiento browniano y procesos de difusión

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar conceptos medida-teóricos a espacios de probabilidad [Familiarizarse]

2. Caracterizar diferentes tipos de procesos estocásticos [Evaluar]
3. Analizar cadenas de Markov para comportamiento transitorio y en estado estacionario [Usar]
4. Modelar sistemas usando procesos estocásticos en tiempo continuo [Usar]

2.3.2. PSC/Inferencia Estadística Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Estimación de Máxima Verosimilitud (MLE)
- Métodos de inferencia bayesiana
- Métodos de Monte Carlo mediante Cadenas de Markov (MCMC)
- Algoritmo Expectación-Maximización (EM)
- Remuestreo bootstrap y jackknife

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar Estimación de Máxima Verosimilitud para varias distribuciones [Usar]
2. Aplicar métodos bayesianos para estimación de parámetros [Usar]
3. Implementar algoritmos MCMC para muestreo posterior [Usar]
4. Usar el algoritmo EM para modelos con variables latentes [Usar]
5. Aplicar métodos bootstrap para intervalos de confianza [Usar]

2.3.3. PSC/Diseño de Experimentos Computacionales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Diseños factoriales y factoriales fraccionados
- Metodología de superficie de respuesta
- Diseños de llenado de espacio (hipercubo latino)
- Diseño óptimo de experimentos
- Experimentos computacionales y modelado sustituto

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar experimentos factoriales para screening de factores [Usar]
2. Aplicar metodología de superficie de respuesta para optimización [Usar]
3. Implementar diseños de llenado de espacio para experimentos computacionales [Usar]
4. Seleccionar diseños óptimos para objetivos específicos [Evaluar]
5. Construir modelos sustitutos para simulaciones costosas [Usar]

2.3.4. PSC/Métodos de Remuestreo y Validación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Estimación bootstrap e intervalos de confianza
- Técnicas de validación cruzada
- Pruebas de permutación y aleatorización
- Selección de modelos y criterios de información
- Métodos de ensamblado y bagging

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar bootstrap para estimación de error estándar [Usar]
2. Aplicar validación cruzada para evaluación de modelos [Usar]
3. Usar pruebas de permutación para contraste de hipótesis [Usar]
4. Comparar modelos usando criterios de información (AIC, BIC) [Evaluar]

5. Implementar bagging para reducción de varianza [Usar]

2.4. Estructuras de Datos y Algoritmos (DSA)

Esta área proporciona el conjunto de herramientas fundamental para diseñar y analizar soluciones computacionales eficientes. Cubre los fundamentos teóricos de la complejidad algorítmica, la implementación práctica de estructuras de datos fundamentales y avanzadas, y los principios de diseño detrás de algoritmos eficientes para búsqueda, ordenamiento, procesamiento de grafos y optimización combinatoria. Este conocimiento forma el núcleo computacional que permite a los estudiantes desarrollar soluciones eficientes en todas las demás áreas de las matemáticas computacionales.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.4.1 Complejidad Algorítmica y Análisis Asintótico (Pág. 16)	1	1
⭐ 2.4.2 Estructuras de Datos Fundamentales y Avanzadas (Pág. 16)	1	1
⭐ 2.4.3 Algoritmos de Búsqueda y Ordenamiento (Pág. 17)	1	1
⭐ 2.4.4 Algoritmos de Grafos y Optimización Combinatoria (Pág. 17)	1	1
⭐ 2.4.5 Algoritmos Paralelos y Distribuidos (Pág. 17)	1	1

Cuadro 2.4: Lista de KUs del área de Estructuras de Datos y Algoritmos.

2.4.1. DSA/Complejidad Algorítmica y Análisis Asintótico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Medidas de complejidad y notación asintótica
- Análisis del mejor caso, caso promedio y peor caso
- Relaciones de recurrencia y técnicas de resolución
- Análisis empírico de algoritmos
- Complejidad espacial y compensaciones

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar algoritmos usando notación asintótica (Big O, Omega, Theta) [Usar]
2. Diferenciar entre complejidades del mejor caso, caso promedio y peor caso [Evaluar]
3. Resolver relaciones de recurrencia para análisis de algoritmos [Usar]
4. Comparar algoritmos mediante mediciones empíricas [Evaluar]

2.4.2. DSA/Estructuras de Datos Fundamentales y Avanzadas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Arreglos, listas enlazadas y arreglos dinámicos
- Pilas, colas y deques
- Tablas hash y resolución de colisiones
- árboles y árboles de búsqueda binaria
- Montículos y colas de prioridad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar estructuras de datos básicas desde cero [Usar]
2. Analizar complejidad temporal/espacial de operaciones de estructuras de datos [Evaluar]
3. Seleccionar estructuras de datos apropiadas para problemas dados [Evaluar]
4. Manejar colisiones hash usando diferentes estrategias [Usar]

2.4.3. DSA/Algoritmos de Búsqueda y Ordenamiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Búsqueda lineal y binaria
- Ordenamiento basado en comparación (QuickSort, MergeSort, HeapSort)
- Ordenamiento sin comparación (RadixSort, CountingSort)
- Algoritmos de selección y estadísticas de orden
- Ordenamiento externo y procesamiento a gran escala

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar y comparar algoritmos de búsqueda [Usar]
2. Analizar complejidad temporal/espacial de algoritmos de ordenamiento [Evaluar]
3. Seleccionar algoritmo de ordenamiento apropiado para características de datos [Evaluar]
4. Implementar algoritmos de estadísticas de orden [Usar]

2.4.4. DSA/Algoritmos de Grafos y Optimización Combinatoria (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Representación de grafos y recorrido (BFS, DFS)
- Algoritmos de camino más corto (Dijkstra, Bellman-Ford)
- árboles de expansión mínima (Prim, Kruskal)
- Flujo en redes y emparejamiento
- NP-completitud y algoritmos de aproximación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de recorrido de grafos [Usar]
2. Resolver problemas de camino más corto usando algoritmos apropiados [Usar]
3. Construir árboles de expansión mínima [Usar]
4. Analizar complejidad de algoritmos de grafos [Evaluar]
5. Diseñar algoritmos de aproximación para problemas NP-difíciles [Usar]

2.4.5. DSA/Algoritmos Paralelos y Distribuidos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelos de computación paralela (PRAM)
- Algoritmos paralelos básicos (reducción, scan)
- Algoritmos distribuidos y consenso
- Sincronización y balanceo de carga
- MapReduce y procesamiento distribuido de datos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar algoritmos paralelos usando el modelo trabajo-span [Evaluar]
2. Implementar operaciones de reducción y scan paralelas [Usar]
3. Diseñar algoritmos distribuidos para consenso [Usar]
4. Aplicar el paradigma MapReduce al procesamiento de datos [Usar]

2.5. Computación de Alto Rendimiento (HPC)

Esta área se enfoca en los principios y prácticas de desarrollar soluciones computacionales que aprovechan los recursos de computación paralela y distribuida para resolver problemas científicos y

de ingeniería a gran escala. Cubre arquitecturas de computadoras, modelos de programación paralela, técnicas de optimización para hardware moderno y la implementación de métodos numéricos en sistemas de alto rendimiento. Este conocimiento permite a los estudiantes abordar problemas computacionalmente intensivos que son inviables en sistemas de un solo procesador.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.5.1 Arquitecturas de Computadoras Paralelas (Pág. 18)	1	1
⭐ 2.5.2 Programación Paralela de Memoria Compartida (Pág. 18)	1	1
⭐ 2.5.3 Programación Paralela de Memoria Distribuida (Pág. 18)	1	1
⭐ 2.5.4 Programación de Aceleradores (Pág. 19)	1	1
⭐ 2.5.5 Métodos Numéricos Paralelos (Pág. 19)	1	1

Cuadro 2.5: Lista de KUs del área de Computación de Alto Rendimiento.

2.5.1. HPC/Arquitecturas de Computadoras Paralelas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Taxonomía de Flynn y arquitecturas paralelas
- Jerarquías de memoria y coherencia de caché
- Procesadores multi-núcleo y memoria compartida
- Sistemas de memoria distribuida y clusters
- Aceleradores (GPUs, FPGAs) y computación heterogénea

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Clasificar arquitecturas paralelas usando la taxonomía de Flynn [Familiarizarse]
2. Analizar efectos de la jerarquía de memoria en el rendimiento [Evaluar]
3. Comparar sistemas multi-núcleo y de memoria distribuida [Evaluar]
4. Seleccionar aceleradores apropiados para tareas computacionales específicas [Evaluar]

2.5.2. HPC/Programación Paralela de Memoria Compartida (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelo de programación OpenMP
- Creación y gestión de hilos
- Primitivas de sincronización (cerrojos, barreras)
- Compartición de datos y alcance
- Optimización de rendimiento y balanceo de carga

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos paralelos usando OpenMP [Usar]
2. Gestionar hilos y paralelismo de tareas [Usar]
3. Aplicar mecanismos de sincronización correctamente [Usar]
4. Optimizar el rendimiento de memoria compartida [Usar]

2.5.3. HPC/Programación Paralela de Memoria Distribuida (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelo de programación MPI
- Comunicación punto a punto

- Operaciones de comunicación colectiva
- Topologías de procesos y grupos
- Análisis de escalabilidad y optimización

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos distribuidos usando MPI [Usar]
2. Manejar comunicaciones punto a punto y colectivas [Usar]
3. Diseñar topologías de procesos eficientes [Usar]
4. Analizar escalabilidad de aplicaciones distribuidas [Evaluar]

2.5.4. HPC/Programación de Aceleradores (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Arquitectura GPU y modelos de programación
- Modelo de programación CUDA
- Directivas OpenACC
- Diseño y optimización de kernels
- Gestión de memoria en aceleradores

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Programar GPUs usando CUDA [Usar]
2. Aplicar directivas OpenACC para programación de aceleradores [Usar]
3. Diseñar kernels GPU eficientes [Usar]
4. Gestionar la jerarquía de memoria GPU efectivamente [Usar]

2.5.5. HPC/Métodos Numéricos Paralelos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Algoritmos de álgebra lineal paralela
- Solucionadores paralelos de EDPs
- Métodos de descomposición de dominio
- Balanceo de carga para computaciones numéricas
- Afinamiento de rendimiento y profiling

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos numéricos paralelos [Usar]
2. Aplicar descomposición de dominio para EDPs [Usar]
3. Balancear carga computacional entre procesadores [Usar]
4. Afinar el rendimiento de códigos numéricos paralelos [Usar]

2.6. Optimización y Aprendizaje Automático (OML)

Esta área cubre los fundamentos matemáticos y métodos computacionales para optimización, con especial énfasis en técnicas relevantes para el aprendizaje automático. Incluye optimización convexa y no convexa, optimización con restricciones, métodos de optimización a gran escala y metaheurísticas. El área conecta la teoría de optimización tradicional con aplicaciones modernas de aprendizaje automático, proporcionando las herramientas necesarias para entrenar modelos complejos y resolver problemas de optimización a gran escala eficientemente.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.6.1 Optimización Convexa y No Convexa (Pág. 20)	1	1
⭐ 2.6.2 Optimización con Restricciones (Pág. 20)	1	1
⭐ 2.6.3 Optimización para Aprendizaje Automático (Pág. 20)	1	1
⭐ 2.6.4 Optimización a Gran Escala y Distribuida (Pág. 21)	1	1
⭐ 2.6.5 Metaheurísticas y Optimización Bioinspirada (Pág. 21)	1	1

Cuadro 2.6: Lista de KUs del área de Optimización y Aprendizaje Automático.

2.6.1. OML/Optimización Convexa y No Convexa (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Conjuntos y funciones convexas
- Condiciones de optimalidad
- Métodos de gradiente para optimización convexa
- Desafíos de optimización no convexa
- Análisis de convergencia

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Caracterizar problemas de optimización convexa [Evaluar]
2. Aplicar condiciones de optimalidad para verificar soluciones [Usar]
3. Implementar métodos basados en gradiente para problemas convexos [Usar]
4. Analizar desafíos en optimización no convexa [Evaluar]

2.6.2. OML/Optimización con Restricciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Programación lineal (método simplex)
- Programación entera
- Multiplicadores de Lagrange y condiciones KKT
- Métodos de penalización y barrera
- Teoría de dualidad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Resolver problemas de programación lineal [Usar]
2. Formular y resolver problemas de programación entera [Usar]
3. Aplicar condiciones KKT a problemas con restricciones [Usar]
4. Usar métodos de penalización para optimización con restricciones [Usar]

2.6.3. OML/Optimización para Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Descenso de gradiente estocástico
- Momentum y métodos adaptativos (Adam, RMSProp)
- Regularización y dispersidad
- Optimización distribuida para AA
- Métodos de segundo orden para AA

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar variantes de descenso de gradiente estocástico [Usar]

2. Aplicar métodos de optimización adaptativos [Usar]
3. Usar regularización para generalización de modelos [Usar]
4. Diseñar estrategias de optimización distribuidas [Usar]

2.6.4. OML/Optimización a Gran Escala y Distribuida (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Métodos de primer orden para problemas a gran escala
- Descenso coordenado y métodos por bloques
- Consenso y optimización distribuida
- Optimización de aprendizaje federado
- Optimización distribuida eficiente en comunicación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar métodos de primer orden a problemas a gran escala [Usar]
2. Implementar algoritmos de descenso coordenado [Usar]
3. Diseñar algoritmos distribuidos basados en consenso [Usar]
4. Optimizar sistemas de aprendizaje federado [Usar]

2.6.5. OML/Metaheurísticas y Optimización Bioinspirada (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Algoritmos genéticos y estrategias evolutivas
- Inteligencia de enjambre (PSO, colonia de hormigas)
- Temple simulado
- Búsqueda tabú
- Metaheurísticas híbridas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos genéticos [Usar]
2. Aplicar métodos de inteligencia de enjambre [Usar]
3. Usar temple simulado para optimización global [Usar]
4. Diseñar enfoques metaheurísticos híbridos [Usar]

2.7. Matemáticas para Ciencia de Datos y Aprendizaje Automático (DSM)

Esta área proporciona los fundamentos matemáticos subyacentes a la ciencia de datos moderna y el aprendizaje automático. Cubre la teoría de aprendizaje estadístico, métodos de aprendizaje supervisado y no supervisado, técnicas de evaluación de modelos y los principios matemáticos que gobiernan el comportamiento de los algoritmos de aprendizaje. El enfoque está en comprender por qué funcionan los métodos de aprendizaje automático, sus garantías teóricas y sus limitaciones, en lugar de solo su aplicación.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.7.1 Fundamentos del Aprendizaje Estadístico (Pág. 22)	1	1
⭐ 2.7.2 Métodos de Aprendizaje Supervisado (Pág. 22)	1	1
⭐ 2.7.3 Métodos de Aprendizaje No Supervisado (Pág. 22)	1	1
⭐ 2.7.4 Evaluación y Selección de Modelos (Pág. 22)	1	1

Cuadro 2.7: Lista de KUs del área de Matemáticas para Ciencia de Datos y Aprendizaje Automático.

2.7.1. DSM/Fundamentos del Aprendizaje Estadístico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Compensación sesgo-varianza
- Teoría de Vapnik-Chervonenkis (VC)
- Minimización de riesgo empírico
- Cotass de generalización
- Fundamentos de teoría de aprendizaje

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar la compensación sesgo-varianza en selección de modelos [Evaluar]
2. Aplicar teoría VC para analizar complejidad de modelos [Usar]
3. Implementar minimización de riesgo empírico [Usar]
4. Interpretar cotas de generalización [Familiarizarse]

2.7.2. DSM/Métodos de Aprendizaje Supervisado (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelos lineales y extensiones
- Máquinas de Vectores Soporte (SVM)
- árboles de decisión y métodos de ensamblado
- Métodos de kernel
- Aprendizaje supervisado probabilístico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar y analizar modelos lineales [Usar]
2. Aplicar SVM para clasificación y regresión [Usar]
3. Construir métodos de ensamblado (bagging, boosting) [Usar]
4. Usar métodos de kernel para problemas no lineales [Usar]

2.7.3. DSM/Métodos de Aprendizaje No Supervisado (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Algoritmos de clustering (k-means, jerárquico)
- Reducción de dimensionalidad (PCA, t-SNE)
- Estimación de densidad
- Aprendizaje de variedades
- Autoencoders y aprendizaje profundo no supervisado

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar algoritmos de clustering para exploración de datos [Usar]
2. Implementar PCA y otros métodos de reducción de dimensionalidad [Usar]
3. Usar estimación de densidad para aprendizaje no supervisado [Usar]
4. Analizar técnicas de aprendizaje de variedades [Evaluar]

2.7.4. DSM/Evaluación y Selección de Modelos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Estrategias de validación cruzada
- Métricas de rendimiento para clasificación y regresión
- Comparación de modelos y pruebas estadísticas
- Optimización de hiperparámetros
- Curvas de aprendizaje y herramientas de diagnóstico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar validación cruzada para evaluación de modelos [Usar]
2. Seleccionar métricas de rendimiento apropiadas [Evaluar]
3. Comparar modelos usando pruebas estadísticas [Usar]
4. Optimizar hiperparámetros usando métodos sistemáticos [Usar]

2.8. Matemáticas para Inteligencia Artificial (AIM)

Esta área cubre los fundamentos matemáticos avanzados de la inteligencia artificial moderna, con énfasis en la teoría del aprendizaje profundo, arquitecturas neuronales avanzadas, aprendizaje por refuerzo y modelos generativos. Proporciona la comprensión teórica necesaria para desarrollar arquitecturas de IA novedosas, analizar sus propiedades y entender sus limitaciones. El enfoque está en los principios matemáticos que permiten a las máquinas aprender representaciones complejas y tomar decisiones inteligentes.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.8.1 Teoría del Aprendizaje Profundo (Pág. 23)	1	1
⭐ 2.8.2 Arquitecturas Neuronales Avanzadas (Pág. 23)	1	1
⭐ 2.8.3 Aprendizaje por Refuerzo (Pág. 24)	1	1
⭐ 2.8.4 Procesamiento de Lenguaje Natural con Transformers (Pág. 24)	1	1
⭐ 2.8.5 Modelos Generativos y GANs (Pág. 24)	1	1

Cuadro 2.8: Lista de KUs del área de Matemáticas para Inteligencia Artificial.

2.8.1. AIM/Teoría del Aprendizaje Profundo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Teoremas de aproximación universal
- Paisaje de optimización de redes profundas
- Generalización en aprendizaje profundo
- Teoría de aprendizaje de representaciones
- Dinámicas de entrenamiento y flujo de gradiente

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar los teoremas de aproximación universal y sus implicaciones [Familiarizarse]
2. Analizar el paisaje de optimización de redes profundas [Evaluar]
3. Estudiar fenómenos de generalización en aprendizaje profundo [Evaluar]
4. Analizar dinámicas de entrenamiento de redes neuronales [Evaluar]

2.8.2. AIM/Arquitecturas Neuronales Avanzadas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Redes Neuronales Convolucionales (CNNs)
- Redes Neuronales Recurrentes (RNNs) y LSTMs
- Arquitecturas Transformer
- Mecanismos de atención

- Redes Neuronales de Grafos (GNNs)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar arquitecturas CNN para tareas visuales [Usar]
2. Implementar RNNs para modelado de secuencias [Usar]
3. Aplicar arquitecturas Transformer a varios dominios [Usar]
4. Usar mecanismos de atención en redes neuronales [Usar]

2.8.3. AIM/Aprendizaje por Refuerzo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Procesos de Decisión de Markov (MDPs)
- Iteración de valor e iteración de política
- Q-learning y aprendizaje por diferencias temporales
- Métodos de gradiente de política
- Aprendizaje por refuerzo profundo

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Formular problemas como Procesos de Decisión de Markov [Usar]
2. Implementar iteración de valor y de política [Usar]
3. Aplicar Q-learning para RL libre de modelo [Usar]
4. Usar métodos de gradiente de política [Usar]

2.8.4. AIM/Procesamiento de Lenguaje Natural con Transformers (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Arquitectura Transformer y auto-atención
- BERT y modelos de lenguaje pre-entrenados
- Modelos GPT y generación autoregresiva
- Transformers multimodales
- Variantes eficientes de transformers

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar modelos transformer desde cero [Usar]
2. Afinar modelos de lenguaje pre-entrenados [Usar]
3. Aplicar modelos GPT para generación de texto [Usar]
4. Usar transformers multimodales para tareas cross-modales [Usar]

2.8.5. AIM/Modelos Generativos y GANs (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Redes Generativas Antagónicas (GANs)
- Autoencoders Variacionales (VAEs)
- Flujos normalizantes
- Modelos de difusión
- Evaluación de modelos generativos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar GANs para generación de imágenes [Usar]
2. Aplicar VAEs para aprendizaje de representaciones latentes [Usar]
3. Usar modelos de difusión para generación [Usar]
4. Evaluar la calidad de modelos generativos [Evaluar]

2.9. Computación Simbólica y Discreta (SDC)

Esta área conecta disciplinas matemáticas clásicas con métodos computacionales, enfocándose en computación simbólica, algoritmos algebraicos, aplicaciones de teoría de números y lógica computacional. Cubre la manipulación de expresiones matemáticas en forma exacta, los aspectos computacionales de las matemáticas discretas y la aplicación de estas técnicas a criptografía y verificación formal. Este conocimiento es esencial para desarrollar sistemas de álgebra computacional, protocolos criptográficos y software formalmente verificado.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.9.1 Teoría de Grupos y Algoritmos Algebraicos (Pág. 25)	1	1
⭐ 2.9.2 Teoría de Números Computacional y Criptografía (Pág. 25)	1	1
⭐ 2.9.3 Sistemas de Computación Simbólica (Pág. 25)	1	1
⭐ 2.9.4 Lógica Computacional y Verificación Formal (Pág. 26)	1	1

Cuadro 2.9: Lista de KUs del área de Computación Simbólica y Discreta.

2.9.1. SDC/Teoría de Grupos y Algoritmos Algebraicos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Fundamentos de teoría de grupos y representación computacional
- Algoritmos algebraicos para operaciones de grupos
- Grupos de permutación y simetría computacional
- Aplicaciones en criptografía y teoría de códigos
- Complejidad de algoritmos algebraicos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar operaciones fundamentales de grupos computacionalmente [Usar]
2. Analizar algoritmos algebraicos para teoría de grupos [Evaluar]
3. Aplicar grupos de permutación a problemas de simetría [Usar]
4. Usar teoría de grupos en aplicaciones criptográficas [Usar]

2.9.2. SDC/Teoría de Números Computacional y Criptografía (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Aritmética modular y algoritmos
- Pruebas de primalidad y factorización
- Criptografía de curva elíptica
- Problema del logaritmo discreto
- Fundamentos de criptografía post-cuántica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de aritmética modular [Usar]
2. Aplicar métodos de prueba de primalidad [Usar]
3. Usar curvas elípticas en protocolos criptográficos [Usar]
4. Analizar sistemas basados en logaritmo discreto [Evaluar]

2.9.3. SDC/Sistemas de Computación Simbólica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Sistemas de álgebra computacional (Mathematica, SymPy)
- Integración y diferenciación simbólica
- Resolución de ecuaciones y simplificación
- Demostración de teoremas y razonamiento automatizado
- Aplicaciones en computación científica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Usar sistemas de álgebra computacional para computación matemática [Usar]
2. Implementar algoritmos de computación simbólica [Usar]
3. Aplicar técnicas de demostración de teoremas [Usar]
4. Conectar computación simbólica y numérica [Evaluar]

2.9.4. SDC/Lógica Computacional y Verificación Formal (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Lógica proposicional y de predicados
- Solucionadores SAT y satisfacibilidad
- Verificación de modelos y lógica temporal
- Verificación de programas y lógica de Hoare
- Demostradores de teoremas interactivos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar lógica computacional a resolución de problemas [Usar]
2. Usar solucionadores SAT para satisfacción de restricciones [Usar]
3. Implementar algoritmos de verificación de modelos [Usar]
4. Verificar programas usando métodos formales [Usar]

2.10. Señales, Imágenes y Visualización (SIV)

Esta área cubre los fundamentos matemáticos y computacionales para procesar, analizar y visualizar señales, imágenes y datos. Incluye análisis de Fourier y wavelets, procesamiento digital de imágenes, algoritmos de visión computacional, técnicas de visualización científica y gráficos por computadora. El enfoque está en desarrollar la comprensión matemática y las habilidades computacionales necesarias para extraer información de datos sensoriales y crear representaciones visuales efectivas de información compleja.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.10.1 Análisis de Fourier y Wavelets (Pág. 26)	1	1
⭐ 2.10.2 Procesamiento Digital de Imágenes y Visión Computacional (Pág. 27)	1	1
⭐ 2.10.3 Visualización Científica y Computación Visual (Pág. 27)	1	1
⭐ 2.10.4 Gráficos por Computadora y Geometría Computacional (Pág. 27)	1	1

Cuadro 2.10: Lista de KUs del área de Señales, Imágenes y Visualización.

2.10.1. SIV/Análisis de Fourier y Wavelets (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Series y transformadas de Fourier
- Transformada de Fourier Discreta (DFT) y FFT
- Transformadas wavelet y análisis multirresolución
- Diseño de filtros y procesamiento de señales
- Aplicaciones en análisis de señales e imágenes

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos FFT para procesamiento de señales [Usar]
2. Aplicar transformadas wavelet para análisis multiescala [Usar]
3. Diseñar filtros digitales para procesamiento de señales [Usar]
4. Analizar señales en el dominio de la frecuencia [Evaluar]

2.10.2. SIV/Procesamiento Digital de Imágenes y Visión Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Representación de imágenes y espacios de color
- Filtrado y mejora de imágenes
- Segmentación de imágenes y detección de características
- Procesamiento morfológico de imágenes
- Algoritmos de visión computacional

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de filtrado de imágenes [Usar]
2. Aplicar técnicas de segmentación de imágenes [Usar]
3. Usar operaciones morfológicas para análisis de imágenes [Usar]
4. Desarrollar aplicaciones de visión computacional [Usar]

2.10.3. SIV/Visualización Científica y Computación Visual (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Principios de visualización y percepción
- Visualización de flujo y campos vectoriales
- Renderizado volumétrico e isosuperficies
- Visualización de información
- Analítica visual y exploración interactiva

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar visualizaciones científicas efectivas [Usar]
2. Implementar técnicas de visualización de flujo [Usar]
3. Aplicar renderizado volumétrico a datos científicos [Usar]
4. Crear herramientas de analítica visual interactiva [Usar]

2.10.4. SIV/Gráficos por Computadora y Geometría Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Pipeline de renderizado y transformaciones
- Algoritmos geométricos y estructuras de datos
- Representación de curvas y superficies
- Detección de colisiones y geometría computacional
- Gráficos en tiempo real y programación GPU

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos básicos de renderizado [Usar]
2. Aplicar algoritmos de geometría computacional [Usar]
3. Diseñar curvas y superficies usando representaciones matemáticas [Usar]

4. Desarrollar aplicaciones gráficas aceleradas por GPU [Usar]

2.11. Computación Cuántica y Algoritmos (QCA)

Esta área introduce los principios fundamentales de la computación cuántica y los algoritmos cuánticos, tendiendo un puente entre la mecánica cuántica y las matemáticas computacionales. Cubre bits cuánticos, compuertas cuánticas, circuitos cuánticos y el desarrollo de algoritmos cuánticos que ofrecen posibles aceleraciones sobre enfoques clásicos. El enfoque está en comprender los fundamentos matemáticos de la computación cuántica e implementar algoritmos cuánticos usando herramientas de simulación clásica.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.11.1 Fundamentos de Mecánica Cuántica para Computación (Pág. 28)	1	1
⭐ 2.11.2 Qubits y Circuitos Cuánticos (Pág. 28)	1	1
⭐ 2.11.3 Algoritmos Cuánticos (Pág. 28)	1	1
⭐ 2.11.4 Criptografía Post-Cuántica y Simulación Cuántica (Pág. 29)	1	1

Cuadro 2.11: Lista de KUs del área de Computación Cuántica y Algoritmos.

2.11.1. QCA/Fundamentos de Mecánica Cuántica para Computación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Estados cuánticos y vectores de estado
- Superposición y entrelazamiento
- Medición cuántica y observables
- Espacios de Hilbert y notación de Dirac
- Evolución unitaria y dinámica cuántica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir estados cuánticos usando notación de Dirac [Familiarizarse]
2. Explicar fenómenos de superposición y entrelazamiento [Familiarizarse]
3. Aplicar principios de medición cuántica [Usar]
4. Analizar transformaciones unitarias en sistemas cuánticos [Evaluar]

2.11.2. QCA/Qubits y Circuitos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Representación de qubits y esfera de Bloch
- Compuertas y circuitos cuánticos
- Conjuntos universales de compuertas cuánticas
- Ruido cuántico y modelos de error
- Simulación de circuitos cuánticos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Representar qubits usando formalismo matemático [Usar]
2. Diseñar circuitos cuánticos usando compuertas estándar [Usar]
3. Implementar conjuntos universales de compuertas cuánticas [Usar]
4. Simular circuitos cuánticos en computadoras clásicas [Usar]

2.11.3. QCA/Algoritmos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Algoritmo de Deutsch-Jozsa
- Algoritmo de búsqueda de Grover
- Algoritmo de factorización de Shor
- Transformada Cuántica de Fourier
- Algoritmos de caminata cuántica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar el algoritmo de Deutsch-Jozsa [Usar]
2. Aplicar el algoritmo de Grover para problemas de búsqueda [Usar]
3. Analizar el algoritmo de factorización de Shor [Evaluar]
4. Usar la Transformada Cuántica de Fourier en algoritmos [Usar]

2.11.4. QCA/Criptografía Post-Cuántica y Simulación Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Amenaza cuántica a la criptografía clásica
- Criptografía basada en retículos
- Criptografía basada en códigos
- Simulación cuántica de sistemas físicos
- Corrección de errores cuánticos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar amenazas cuánticas a sistemas criptográficos [Evaluar]
2. Implementar esquemas criptográficos basados en retículos [Usar]
3. Simular sistemas cuánticos usando computadoras clásicas [Usar]
4. Aplicar códigos de corrección de errores cuánticos [Usar]

2.12. Ciberseguridad y Privacidad (CSP)

Esta área cubre los fundamentos matemáticos de las tecnologías modernas de ciberseguridad y preservación de la privacidad. Incluye protocolos criptográficos, análisis de seguridad, tecnologías de mejora de la privacidad y aplicaciones de teoría de la información. El enfoque está en desarrollar la comprensión matemática necesaria para diseñar, analizar e implementar sistemas seguros y algoritmos de preservación de la privacidad en el contexto de las matemáticas computacionales.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.12.1 Criptografía Simétrica (Pág. 29)	1	1
⭐ 2.12.2 Criptografía Asimétrica (Pág. 30)	1	1
⭐ 2.12.3 Computación con Preservación de la Privacidad (Pág. 30)	1	1
⭐ 2.12.4 Teoría de la Información y Compresión de Datos (Pág. 30)	1	1

Cuadro 2.12: Lista de KUs del área de Ciberseguridad y Privacidad.

2.12.1. CSP/Criptografía Simétrica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Cifrados de bloque y modos de operación
- Cifrados de flujo
- Funciones hash criptográficas
- Códigos de autenticación de mensajes
- Técnicas de criptoanálisis

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Implementar algoritmos de cifrado de bloque [Usar]
2. Aplicar modos de operación para cifrado [Usar]
3. Usar funciones hash criptográficas [Usar]
4. Analizar ataques de criptoanálisis [Evaluar]

2.12.2. CSP/Criptografía Asimétrica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Principios de criptografía de clave pública
- Criptosistema RSA
- Criptografía de curva elíptica
- Intercambio de clave Diffie-Hellman
- Firmas digitales

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Implementar cifrado y descifrado RSA [Usar]
2. Aplicar criptografía de curva elíptica [Usar]
3. Usar Diffie-Hellman para intercambio seguro de claves [Usar]
4. Implementar esquemas de firma digital [Usar]

2.12.3. CSP/Computación con Preservación de la Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Privacidad diferencial
- Cifrado homomórfico
- Computación multipartita segura
- Privacidad en aprendizaje federado
- Métricas y análisis de privacidad

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Aplicar mecanismos de privacidad diferencial [Usar]
2. Usar cifrado homomórfico para computación privada [Usar]
3. Implementar protocolos de computación multipartita segura [Usar]
4. Analizar garantías de privacidad de algoritmos [Evaluar]

2.12.4. CSP/Teoría de la Información y Compresión de Datos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Entropía y medidas de información
- Teorema de codificación de fuente
- Capacidad de canal y canales ruidosos
- Algoritmos de compresión
- Seguridad criptográfica y teoría de la información

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Calcular entropía e información mutua [Usar]
2. Aplicar codificación de fuente a compresión de datos [Usar]
3. Analizar capacidad de canal para comunicación [Evaluar]
4. Implementar algoritmos de compresión [Usar]

2.13. Matemáticas Computacionales para Dominios Específicos (CDM)

Esta área aplica métodos de matemáticas computacionales a dominios de aplicación específicos, demostrando cómo el modelado matemático, la simulación y las técnicas de análisis de datos se utilizan para resolver problemas del mundo real en varios campos científicos y de ingeniería. Cubre biología computacional, finanzas cuantitativas, física computacional, ciencia de materiales y modelado climático, proporcionando a los estudiantes conocimiento específico del dominio y habilidades interdisciplinarias de resolución de problemas.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.13.1 Bioinformática y Biología Computacional (Pág. 31)	1	1
⭐ 2.13.2 Finanzas Computacionales y Ciencia del Riesgo (Pág. 31)	1	1
⭐ 2.13.3 Física Computacional y Química Cuántica (Pág. 32)	1	1
⭐ 2.13.4 Ciencia de Materiales Computacional (Pág. 32)	1	1
⭐ 2.13.5 Modelado Climático y Sistemas Complejos (Pág. 32)	1	1

Cuadro 2.13: Lista de KUs del área de Matemáticas Computacionales para Dominios Específicos.

2.13.1. CDM/Bioinformática y Biología Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Alineamiento y análisis de secuencias
- Análisis de datos genómicos
- Predicción de estructura de proteínas
- Biología de sistemas y análisis de redes
- Construcción de árboles filogenéticos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de alineamiento de secuencias [Usar]
2. Analizar conjuntos de datos genómicos usando métodos estadísticos [Usar]
3. Modelar estructuras de proteínas usando métodos computacionales [Usar]
4. Aplicar enfoques de biología de sistemas [Usar]

2.13.2. CDM/Finanzas Computacionales y Ciencia del Riesgo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelos de valoración de opciones
- Optimización de portafolios
- Gestión de riesgos y valor en riesgo
- Cálculo estocástico en finanzas
- Estrategias de trading algorítmico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Valorar opciones usando métodos numéricos [Usar]
2. Optimizar portafolios de inversión [Usar]
3. Calcular medidas de riesgo y valor en riesgo [Usar]
4. Aplicar cálculo estocástico al modelado financiero [Usar]

2.13.3. CDM/Física Computacional y Química Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Simulación de dinámica molecular
- Cálculos de química cuántica
- Métodos Monte Carlo en física
- Simulación de mecánica de medios continuos
- Cálculo de campos electromagnéticos

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Simular dinámica molecular [Usar]
2. Implementar algoritmos de química cuántica [Usar]
3. Aplicar métodos Monte Carlo a sistemas físicos [Usar]
4. Modelar problemas de mecánica de medios continuos [Usar]

2.13.4. CDM/Ciencia de Materiales Computacional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Predicción de estructura cristalina
- Modelado de campo de fase
- Cálculos de Teoría del Funcional de la Densidad (DFT)
- Modelado multiescala de materiales
- Informática de materiales y aprendizaje automático

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Predecir estructuras cristalinas usando métodos computacionales [Usar]
2. Aplicar modelos de campo de fase para evolución de materiales [Usar]
3. Implementar cálculos DFT para estructura electrónica [Usar]
4. Usar enfoques de modelado multiescala [Usar]

2.13.5. CDM/Modelado Climático y Sistemas Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core
- Modelado del sistema climático
- Dinámica de fluidos y modelos atmosféricos
- Sistemas complejos y teoría de redes
- Modelado de ecosistemas
- Cuantificación de incertidumbre en modelos climáticos

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Modelar sistemas climáticos usando ecuaciones diferenciales [Usar]
2. Simular dinámica de fluidos para modelado atmosférico [Usar]
3. Analizar sistemas complejos usando teoría de redes [Evaluar]
4. Cuantificar incertidumbre en modelos ambientales [Usar]

2.14. Gobernanza y Gestión de Datos (DGM)

Esta área cubre los aspectos éticos, gerenciales y organizacionales de la ciencia de datos y las matemáticas computacionales. Aborda el desarrollo y despliegue responsable de sistemas de IA, metodologías de gestión de proyectos para proyectos intensivos en datos, marcos de gobernanza de datos y

cumplimiento regulatorio. El enfoque está en desarrollar las habilidades profesionales necesarias para liderar proyectos de matemáticas computacionales de manera responsable y efectiva en contextos del mundo real.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 2.14.1 ética en IA y Algoritmos Responsables (Pág. 33)	1	1
⭐ 2.14.2 Gestión de Proyectos de Datos y MLOps (Pág. 33)	1	1
⭐ 2.14.3 Gobernanza de Datos y Cumplimiento Normativo (Pág. 33)	1	1

Cuadro 2.14: Lista de KUs del área de Gobernanza y Gestión de Datos.

2.14.1. DGM/ética en IA y Algoritmos Responsables (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Equidad algorítmica y mitigación de sesgos
- Transparencia e IA explicable (XAI)
- Responsabilidad y rendición de cuentas en sistemas de IA
- Marcos éticos para el desarrollo de IA
- Principios FAT/ML (Equidad, Responsabilidad, Transparencia)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Detectar y mitigar sesgos algorítmicos [Usar]
2. Implementar técnicas de IA explicable [Usar]
3. Aplicar marcos éticos a proyectos de IA [Evaluar]
4. Evaluar sistemas usando principios FAT/ML [Evaluar]

2.14.2. DGM/Gestión de Proyectos de Datos y MLOps (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- MLOps y gestión del ciclo de vida de modelos
- Metodologías de gestión de proyectos de datos
- Control de versiones para modelos y datos
- Despliegue y monitoreo de modelos
- Infraestructura para ciencia de datos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar prácticas MLOps [Usar]
2. Gestionar proyectos de ciencia de datos efectivamente [Usar]
3. Desplegar y monitorear modelos de aprendizaje automático [Usar]
4. Configurar infraestructura para ciencia de datos [Usar]

2.14.3. DGM/Gobernanza de Datos y Cumplimiento Normativo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Marcos de gobernanza de datos
- GDPR y regulaciones de protección de datos
- Gestión de calidad de datos
- ética de datos y administración
- Monitoreo y auditoría de cumplimiento

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar marcos de gobernanza de datos [Usar]
2. Aplicar GDPR y regulaciones de privacidad [Usar]
3. Gestionar calidad de datos a lo largo del ciclo de vida [Usar]
4. Garantizar cumplimiento normativo en proyectos de datos [Usar]

Capítulo 3

Cuerpo del conocimiento de Ciencia de la Computación–*Computer Science*

Las 17 áreas de conocimiento en Ciencia de la Computación–*Computer Science* son:

3.1 Inteligencia Artificial (AI) (Pág. 37)
- 3.1.1 Introducción a la IA: Conceptos y Comportamiento Inteligente (Pág. 39)
- 3.1.2 Características del Problema y Naturaleza de los Agentes (Pág. 39)
- 3.1.3 Estado y Búsqueda No Informada (Pág. 39)
- 3.1.4 Búsqueda Heurística y CSP (Pág. 40)
- 3.1.5 Búsqueda Adversarial y Optimización Estocástica (Pág. 40)
- 3.1.6 Representación del Conocimiento y Razonamiento Probabilístico Básico (Pág. 41)
- 3.1.7 Independencia, Modelos de Markov y Toma de Decisiones (Pág. 41)
- 3.1.8 Fundamentos de Aprendizaje Automático (Pág. 41)
- 3.1.9 Aprendizaje Automático: Datos, Representaciones y Evaluación (Pág. 42)
- 3.1.10 Redes Neuronales y Ética (Pág. 43)
- 3.1.11 Formulación del Aprendizaje Automático y Evaluación (Pág. 43)
- 3.1.12 Aprendizaje Profundo, RL y Ética (Pág. 43)
- 3.1.13 Entrenamiento y Optimización de Redes Neuronales (Pág. 44)
- 3.1.14 Técnicas de Regularización para Redes Profundas (Pág. 44)
- 3.1.15 Arquitecturas de Redes Convolucionales y Residuales (Pág. 44)
- 3.1.16 Arquitecturas Transformer y Modelos Generativos (Pág. 44)
- 3.1.17 Aplicaciones e Impacto Social de la IA (Pág. 45)
- 3.1.18 Aplicaciones Adicionales e Impacto Social de la IA (Pág. 45)
- 3.1.19 Representación Lógica y Razonamiento (Pág. 46)
- 3.1.20 Representación Probabilística y Razonamiento (Pág. 46)
- 3.1.21 Planificación (Pág. 47)
- 3.1.22 Agentes y Sistemas Cognitivos (Pág. 47)
- 3.1.23 Ingeniería de Bucles de Agentes (Loop Engineering) (Pág. 48)
- 3.1.24 Procesamiento del Lenguaje Natural (Pág. 48)
- 3.1.25 Robótica (Pág. 49)
- 3.1.26 Percepción y Visión por Computadora (Pág. 50)

3.2 Fundamentos Algorítmicos (AL) (Pág. 51)
- 3.2.1 Estructuras de Datos Fundamentales (Pág. 52)
- 3.2.2 Algoritmos Fundamentales (Pág. 52)
- 3.2.3 Algoritmos Avanzados (Pág. 53)
- 3.2.4 Estrategias Algorítmicas (Pág. 53)
- 3.2.5 Marco de Análisis de Complejidad (Pág. 54)
- 3.2.6 Notación Asintótica y Clases de Complejidad (Pág. 54)
- 3.2.7 Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (Pág. 55)
- 3.2.8 Teoría de la Complejidad Computacional (Pág. 55)
- 3.2.9 Lenguajes Formales y Autómatas (Pág. 55)
- 3.2.10 Computabilidad y Decidibilidad (Pág. 56)
- 3.2.11 Sociedad, ética y la Profesión (Pág. 57)

3.3 Arquitectura y Organización (AR) (Pág. 57)
- 3.3.1 Lógica Digital y Sistemas Digitales (Pág. 58)
- 3.3.2 Representación de Datos a Nivel de Máquina (Pág. 58)
- 3.3.3 Organización de Máquina a Nivel de Ensamblador (Pág. 59)
- 3.3.4 Jerarquía de Memoria (Pág. 60)
- 3.3.5 Interconexión y Comunicación (Pág. 60)
- 3.3.6 Organización Funcional (Pág. 60)
- 3.3.7 Rendimiento y Eficiencia Energética (Pág. 61)
- 3.3.8 Arquitecturas Heterogéneas (Pág. 61)
- 3.3.9 Arquitecturas de Procesador Seguras (Pág. 62)
- 3.3.10 Arquitecturas Cuánticas (Pág. 62)
- 3.3.11 Problemas de Sostenibilidad (Pág. 63)

3.4 Gestión de Datos (DM) (Pág. 63)
- 3.4.1 El Rol de los Datos y el Ciclo de Vida de los Datos (Pág. 65)
- 3.4.2 Conceptos Fundamentales de Sistemas de Bases de Datos (Pág. 65)
- 3.4.3 Modelado de Datos (Pág. 66)
- 3.4.4 Bases de Datos Relacionales (Pág. 66)
- 3.4.5 Construcción de Consultas (Pág. 67)
- 3.4.6 Procesamiento de Consultas (Pág. 67)
- 3.4.7 Internos de DBMS (Pág. 68)
- 3.4.8 Sistemas NoSQL (Pág. 68)
- 3.4.9 Seguridad y Privacidad de Datos (Pág. 69)
- 3.4.10 Análisis de Datos (Pág. 69)
- 3.4.11 Bases de Datos Distribuidas/Computación en la Nube (Pág. 70)
- 3.4.12 Bases de Datos Semiestructuradas y No Estructuradas (Pág. 70)
- 3.4.13 Sociedad, ética y Profesión (Pág. 70)

3.5 Fundamentos de los Lenguajes de Programación (FPL) (Pág. 71)
- 3.5.1 Fundamentos de Programación Orientada a Objetos (Pág. 72)
- 3.5.2 Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (Pág. 73)
- 3.5.3 Programación Funcional (Pág. 73)
- 3.5.4 Programación Lógica (Pág. 74)
- 3.5.5 Scripting de Shell (Pág. 75)
- 3.5.6 Programación Dirigida por Eventos y Reactiva (Pág. 75)
- 3.5.7 Computación Paralela y Distribuida (Pág. 76)
- 3.5.8 Programación Orientada a Aspectos (Pág. 76)
- 3.5.9 Fundamentos de Sistemas de Tipos (Pág. 77)
- 3.5.10 Sistemas de Tipos: Polimorfismo y Tipado Complementario (Pág. 77)
- 3.5.11 Teoría de Tipos Avanzada (Pág. 78)
- 3.5.12 Modelo de Ejecución y Memoria de Sistemas (Pág. 78)
- 3.5.13 Gestión de Memoria (Pág. 79)

- 3.5.14 Traducción y Ejecución de Lenguajes (Pág. 79)
- 3.5.15 Abstracción y Representación de Programas (Pág. 80)
- 3.5.16 Análisis Sintáctico (Pág. 81)
- 3.5.17 Análisis Semántico del Compilador (Pág. 81)
- 3.5.18 Análisis de Programas y Analizadores (Pág. 82)
- 3.5.19 Generación de Código (Pág. 82)
- 3.5.20 Comportamiento en Tiempo de Ejecución y Sistemas (Pág. 83)
- 3.5.21 Construcciones de Programación Avanzadas (Pág. 83)
- 3.5.22 Pragmática del Lenguaje (Pág. 84)
- 3.5.23 Semántica Formal (Pág. 84)
- 3.5.24 Metodologías de Desarrollo Formal (Pág. 85)
- 3.5.25 Principios de Diseño de Lenguajes de Programación (Pág. 85)
- 3.5.26 Sociedad, ética y Profesión (Pág. 86)

3.6 Gráficos y Técnicas Interactivas (GIT) (Pág. 86)
- 3.6.1 Conceptos Fundamentales de Gráficos y Técnicas Interactivas (Pág. 87)
- 3.6.2 Visualización (Pág. 87)
- 3.6.3 Renderizado Aplicado y Técnicas (Pág. 88)
- 3.6.4 Modelado Geométrico (Pág. 88)
- 3.6.5 Sombreado y Renderizado Avanzado (Pág. 88)
- 3.6.6 Animación por Computadora (Pág. 89)
- 3.6.7 Simulación (Pág. 89)
- 3.6.8 Inmersión (Pág. 89)
- 3.6.9 Interacción (Pág. 89)
- 3.6.10 Procesamiento de Imágenes (Pág. 90)
- 3.6.11 Computación Tangible/Física (Pág. 90)
- 3.6.12 Sociedad, ética y Profesión (Pág. 90)

3.7 Interacción Humano-Computadora (HCI) (Pág. 91)
- 3.7.1 Comprender al Usuario: Objetivos individuales e interacciones con otros (Pág. 91)
- 3.7.2 Responsabilidad y Rendición de Cuentas en el Diseño (Pág. 92)
- 3.7.3 Accesibilidad y Diseño Inclusivo (Pág. 92)
- 3.7.4 Evaluación del Diseño (Pág. 92)
- 3.7.5 Diseño de Sistemas (Pág. 93)
- 3.7.6 Sociedad, ética y Profesión (Pág. 93)

3.8 Fundamentos Matemáticos y Estadísticos (MSF) (Pág. 93)
- 3.8.1 Matemáticas Discretas (Pág. 94)
- 3.8.2 Matemáticas Discretas Avanzadas (Pág. 94)
- 3.8.3 Probabilidad (Pág. 95)
- 3.8.4 Estadística (Pág. 95)
- 3.8.5 álgebra Lineal (Pág. 96)
- 3.8.6 Cálculo (Pág. 96)

3.9 Redes y Comunicaciones (NC) (Pág. 97)
- 3.9.1 Fundamentos de Redes y Comunicaciones (Pág. 97)
- 3.9.2 Aplicaciones en Red (Pág. 98)
- 3.9.3 Soporte de Fiabilidad (Pág. 98)
- 3.9.4 Enrutamiento y Reenvío (Pág. 99)
- 3.9.5 Comunicación de un solo salto (Pág. 99)
- 3.9.6 Seguridad de Red (Pág. 99)
- 3.9.7 Soporte de Movilidad (Pág. 100)
- 3.9.8 Temas Emergentes (Pág. 100)

3.10 Sistemas Operativos (OS) (Pág. 100)
- 3.10.1 Rol y Propósito de los Sistemas Operativos (Pág. 101)
- 3.10.2 Principios del Sistema Operativo (Pág. 102)
- 3.10.3 Concurrencia (Pág. 102)
- 3.10.4 Protección y Seguridad (Pág. 103)
- 3.10.5 Planificación (Pág. 104)
- 3.10.6 Modelo de Proceso (Pág. 104)
- 3.10.7 Gestión de Memoria (Pág. 105)
- 3.10.8 Gestión de Dispositivos (Pág. 105)
- 3.10.9 API de Sistemas de Archivos e Implementación (Pág. 106)
- 3.10.10 Sistemas de Archivos Avanzados (Pág. 106)
- 3.10.11 Virtualización (Pág. 107)
- 3.10.12 Sistemas en Tiempo Real y Embebidos (Pág. 108)
- 3.10.13 Tolerancia a fallos (Pág. 108)
- 3.10.14 Sociedad, ética y Profesión (Pág. 109)

3.11 Computación Paralela y Distribuida (PDC) (Pág. 109)
- 3.11.1 Programas: Paralelismo Declarativo (Pág. 110)
- 3.11.2 Programas: Inicio de Actividades (Pág. 111)
- 3.11.3 Programas: Propiedades de Ejecución (Pág. 111)
- 3.11.4 Programas: Distribución (Pág. 111)
- 3.11.5 Programas: Mapeos de Implementación (Pág. 112)
- 3.11.6 Programación GPU (Pág. 112)
- 3.11.7 Comunicación (Pág. 112)
- 3.11.8 Comunicación: Propiedades y Extensiones (Pág. 113)
- 3.11.9 Memoria y Consistencia (Pág. 114)
- 3.11.10 Coordinación (Pág. 114)
- 3.11.11 Coordinación: Sincronización y Atomicidad (Pág. 115)
- 3.11.12 Coordinación: Propiedades Avanzadas (Pág. 115)
- 3.11.13 Evaluación (Pág. 115)
- 3.11.14 Evaluación: Métodos Formales (Pág. 116)
- 3.11.15 Evaluación: Pruebas y Medición (Pág. 116)
- 3.11.16 Algoritmos (Pág. 117)
- 3.11.17 Algoritmos: Dominios de Aplicación (Pág. 117)
- 3.11.18 Algoritmos: Dominios Algorítmicos (Pág. 118)

3.12 Fundamentos de Desarrollo de Software (SDF) (Pág. 118)
- 3.12.1 Conceptos y Construcciones Básicas de Programación (Pág. 119)
- 3.12.2 Prácticas de Programación: Pruebas, Depuración y Documentación (Pág. 120)
- 3.12.3 Estructuras de Datos Fundamentales (Pág. 120)
- 3.12.4 Algoritmos (Pág. 121)
- 3.12.5 Prácticas de Desarrollo de Software (Pág. 121)
- 3.12.6 Sociedad, ética y la Profesión (Pág. 121)

3.13 Ingeniería de Software (SE) (Pág. 122)
- 3.13.1 Colaboración y Dinámica de Equipo (Pág. 123)
- 3.13.2 Diversidad, Stakeholders y Equipos Distribuidos (Pág. 124)
- 3.13.3 Control de Versiones y CI/CD (Pág. 124)
- 3.13.4 Entorno de Desarrollo e IDE (Pág. 125)
- 3.13.5 Ingeniería de Requisitos (Pág. 125)
- 3.13.6 Evolución y Estimación de Requisitos (Pág. 126)
- 3.13.7 Principios y Arquitectura de Software (Pág. 126)
- 3.13.8 Diseño de API y Modelado (Pág. 127)
- 3.13.9 Calidad y Evaluación del Diseño (Pág. 127)
- 3.13.10 Prácticas de Codificación (Pág. 128)
- 3.13.11 Construcción a Escala y Proceso (Pág. 128)
- 3.13.12 Conceptos de verificación y validación (Pág. 129)
- 3.13.13 Planificación y Tipos de Pruebas (Pág. 129)
- 3.13.14 Prácticas Avanzadas de Pruebas (Pág. 130)
- 3.13.15 Herramientas y Análisis de Pruebas (Pág. 130)
- 3.13.16 Pruebas de Rendimiento y Evaluación Comparativa (Pág. 131)
- 3.13.17 Compatibilidad y Versionado de API (Pág. 131)
- 3.13.18 Técnicas de Refactorización (Pág. 132)
- 3.13.19 Conceptos de Confiabilidad (Pág. 132)
- 3.13.20 Ingeniería de Confiabilidad (Pág. 133)
- 3.13.21 Métodos Formales (Pág. 133)

3.14 Seguridad (SEC) (Pág. 134)
- 3.14.1 Principios y Mentalidad de Seguridad (Pág. 135)
- 3.14.2 Control de Acceso y Aplicaciones (Pág. 136)
- 3.14.3 Seguridad en IA y Hardware (Pág. 136)
- 3.14.4 Ética, Ley y Sociedad en Seguridad (Pág. 137)
- 3.14.5 Privacidad y Seguridad por Diseño (Pág. 137)
- 3.14.6 Inyección y Validación de Entrada (Pág. 138)
- 3.14.7 Seguridad de Memoria y Tipos (Pág. 138)
- 3.14.8 Flujo de Información y No Interferencia (Pág. 139)
- 3.14.9 Análisis de Malware y Seguridad Avanzada (Pág. 139)
- 3.14.10 Fundamentos de Criptografía (Pág. 140)
- 3.14.11 Algoritmos Criptográficos (Pág. 140)
- 3.14.12 Protocolos, Aplicaciones y Criptografía Avanzada (Pág. 140)
- 3.14.13 Análisis de Amenazas e Ingeniería de Seguridad (Pág. 141)
- 3.14.14 Diseño e Ingeniería de Controles de Seguridad (Pág. 142)
- 3.14.15 Computación Confiable e Ingeniería de Privacidad (Pág. 142)
- 3.14.16 Técnicas de Informática Forense (Pág. 142)
- 3.14.17 Aspectos Legales y Manejo de Incidentes (Pág. 143)
- 3.14.18 Gobierno, Gestión y Política de Seguridad (Pág. 143)
- 3.14.19 Gestión de Riesgos y Datos (Pág. 144)

3.15 Sociedad, ética y la Profesión (SEP) (Pág. 144)
- 3.15.1 Impacto Social e IA (Pág. 145)
- 3.15.2 Equidad Digital y Accesibilidad (Pág. 146)
- 3.15.3 Fundamentos y Marcos del Análisis Ético (Pág. 146)
- 3.15.4 Métodos y Herramientas de Análisis Ético (Pág. 147)

- 3.15.5 Valores, Naturaleza y Actualización Profesional (Pág. 147)
- 3.15.6 Códigos de Ética y Responsabilidad Legal (Pág. 148)
- 3.15.7 Responsabilidad Profesional y Política Pública (Pág. 148)
- 3.15.8 Disidencia Ética y Contexto Cultural (Pág. 149)
- 3.15.9 Ética en el Lugar de Trabajo (Pág. 149)
- 3.15.10 Fundamentos de la Propiedad Intelectual (Pág. 149)
- 3.15.11 Licencias, Plagio y Autoría (Pág. 150)
- 3.15.12 Derechos de PI Digital y Derechos de Autor (Pág. 150)
- 3.15.13 Open Source, Piratería y DRM (Pág. 151)
- 3.15.14 Datos, Vigilancia y Tecnología de Privacidad (Pág. 151)
- 3.15.15 Fundamentos Filosóficos y Legislación de Privacidad (Pág. 151)
- 3.15.16 Libertades Civiles y Expresión (Pág. 152)
- 3.15.17 Comunicación Técnica y Profesional (Pág. 152)
- 3.15.18 Comunicación en Equipo y Cultura (Pág. 153)
- 3.15.19 Huella Ambiental de los Sistemas Informáticos (Pág. 153)
- 3.15.20 Efectos Sistémicos y Contexto Social (Pág. 154)
- 3.15.21 Diseño Sostenible y Computación Ubicua (Pág. 154)
- 3.15.22 Fundamentos e Historia Temprana de la Informática (Pág. 154)
- 3.15.23 Historia Moderna: PC, Internet e IA (Pág. 155)
- 3.15.24 Modelos y Estrategias Económicas (Pág. 155)
- 3.15.25 Automatización, IA y Ética Económica (Pág. 156)
- 3.15.26 Delitos y Ataques Informáticos (Pág. 156)
- 3.15.27 Leyes, Políticas y Responsabilidad en Seguridad (Pág. 157)
- 3.15.28 Identidad y Diversidad en la Informática (Pág. 157)
- 3.15.29 Contexto Histórico e Intercultural de DEIA (Pág. 157)
- 3.15.30 Inclusión y Prácticas en Informática (Pág. 158)
- 3.15.31 Diseño Accesible (Pág. 158)

3.16 Fundamentos de Sistemas (SF) (Pág. 159)

- 3.16.1 Componentes Hardware y Lógica (Pág. 159)
- 3.16.2 Capas, Abstracciones y Escalabilidad (Pág. 160)
- 3.16.3 Sistemas Digitales y Lógica (Pág. 160)
- 3.16.4 Máquinas de Estado (Pág. 161)
- 3.16.5 Procesamiento Secuencial y Paralelo (Pág. 161)
- 3.16.6 Gestión de Recursos (Pág. 162)
- 3.16.7 Latencia, Caché y Jerarquía de Memoria (Pág. 162)
- 3.16.8 Virtualización y Aislamiento (Pág. 162)
- 3.16.9 Métricas y Benchmarks de Rendimiento (Pág. 163)
- 3.16.10 Modelos Analíticos de Rendimiento (Pág. 163)
- 3.16.11 Eventos, Herramientas y Experimentación (Pág. 164)
- 3.16.12 Confiabilidad del Sistema (Pág. 164)
- 3.16.13 Seguridad del Sistema (Pág. 165)
- 3.16.14 Diseño del Sistema (Pág. 165)
- 3.16.15 Sociedad, ética y la Profesión (Pág. 165)

3.17 Desarrollo de Plataformas Especializadas (SPD) (Pág. 166)

- 3.17.1 Aspectos Comunes: Plataformas, APIs y Restricciones (Pág. 167)
- 3.17.2 Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (Pág. 168)
- 3.17.3 Frontend, Frameworks y Estándares Web (Pág. 168)
- 3.17.4 Arquitectura Cloud y Gestión de Datos (Pág. 168)
- 3.17.5 Plataformas Móviles: Fundamentos del Desarrollo (Pág. 169)
- 3.17.6 Plataformas Móviles: Arquitectura y Restricciones (Pág. 169)
- 3.17.7 Plataformas Móviles: Capacidades y Computación Asíncrona (Pág. 169)
- 3.17.8 Plataformas Robóticas (Pág. 170)
- 3.17.9 Características y Arquitecturas Embebidas (Pág. 170)
- 3.17.10 Sistemas, Programación y Sensores (Pág. 171)
- 3.17.11 Tiempo Real, Memoria y Verificación (Pág. 171)
- 3.17.12 Historia, Tipos y Características de Plataformas (Pág. 171)
- 3.17.13 Sensores, Controladores y Ecosistemas eSports (Pág. 172)
- 3.17.14 Arquitecturas y Renderizado (Pág. 172)
- 3.17.15 Algoritmos, Medios y Mundos Virtuales (Pág. 173)
- 3.17.16 Lenguajes, APIs y Herramientas de Desarrollo (Pág. 173)
- 3.17.17 Motores de Juego (Pág. 173)
- 3.17.18 Fundamentos del Diseño de Juegos (Pág. 174)
- 3.17.19 Géneros, Audiencias y Alcance Social (Pág. 174)
- 3.17.20 Plataformas Interactivas de Datos y Análisis (Pág. 175)
- 3.17.21 Programación con IA y Plataformas Cuánticas (Pág. 175)
- 3.17.22 SPD-SEP/Móvil (Pág. 175)
- 3.17.23 SPD-SEP Web (Pág. 176)
- 3.17.24 Propiedad Intelectual en los Juegos (Pág. 176)
- 3.17.25 Accesibilidad e Interfaz del Juego (Pág. 177)
- 3.17.26 Salud, Seguridad y Sostenibilidad (Pág. 177)
- 3.17.27 Representación, Inclusión e Igualdad (Pág. 178)
- 3.17.28 SPD-SEP/Robótica (Pág. 178)
- 3.17.29 SPD-SEP/Interactiva (Pág. 178)

3.1. Inteligencia Artificial (AI)

La inteligencia artificial (IA) estudia problemas que son difíciles o impracticables de resolver con enfoques algorítmicos tradicionales. Estos problemas a menudo recuerdan a aquellos que se consideran que requieren inteligencia humana, y las estrategias de solución de IA resultantes típicamente se generalizan sobre clases de problemas. Las técnicas de IA ahora son omnipresentes en la informática, apoyando aplicaciones cotidianas como el correo electrónico, las redes sociales, la fotografía, los mercados financieros y los asistentes virtuales inteligentes (por ejemplo, Siri, Alexa). Estas técnicas también se utilizan en el diseño y análisis de agentes autónomos que perciben su entorno e interactúan racionalmente con él, como vehículos autónomos y otros robots.

Tradicionalmente, la IA ha incluido una mezcla de enfoques simbólicos y subsimbólicos. Las soluciones que proporciona se basan en un amplio conjunto de esquemas de representación del conocimiento generales y especializados, mecanismos de resolución de problemas y técnicas de optimización. Estos enfoques abordan la percepción (por ejemplo, reconocimiento de voz, comprensión del lenguaje natural, visión por computadora), la resolución de problemas (por ejemplo, búsqueda, planificación, optimización), la generación (por ejemplo, narrativa, conversación, imágenes, modelos, recomendaciones), la actuación (por ejemplo, robótica, automatización de tareas, control) y las arquitecturas necesarias para soportarlas (por ejemplo, agentes únicos, sistemas multiagente). El aprendizaje automático puede

utilizarse dentro de cada uno de estos aspectos e incluso puede emplearse de extremo a extremo en todos ellos. El estudio de la Inteligencia Artificial prepara a los estudiantes para determinar cuándo un enfoque de IA es apropiado para un problema dado, identificar representaciones y mecanismos de razonamiento apropiados, implementarlos y evaluarlos tanto con respecto al rendimiento como a su impacto social más amplio.

Durante la última década, el término ”inteligencia artificial” se ha vuelto común en empresas, artículos de noticias y conversaciones cotidianas, impulsado en gran medida por una serie de aplicaciones de aprendizaje automático de alto impacto. Estos avances fueron posibles gracias a la amplia disponibilidad de grandes conjuntos de datos, el aumento de la potencia computacional y las mejoras algorítmicas. En particular, ha habido un cambio desde representaciones diseñadas manualmente hacia representaciones aprendidas automáticamente a través de la optimización sobre grandes conjuntos de datos. Los avances resultantes han puesto términos como ”redes neuronales” y ”aprendizaje profundo” en el vocabulario cotidiano. Las empresas ahora anuncian soluciones basadas en IA como valor añadido a sus servicios, de modo que ”inteligencia artificial” es ahora tanto un término técnico como una palabra de moda de marketing. Otras disciplinas, como la biología, el arte, la arquitectura y las finanzas, utilizan cada vez más técnicas de IA para resolver problemas dentro de sus disciplinas.

Por primera vez en nuestra historia, la población en general tiene acceso a herramientas sofisticadas impulsadas por IA, incluyendo herramientas para generar ensayos o poemas a partir de un indicador, obras de arte a partir de una descripción, y fotografías o videos falsos que representan personas reales. La tecnología de IA ahora se utiliza ampliamente en el comercio de acciones, la curación de nuestras noticias y feeds de redes sociales, la evaluación automatizada de solicitantes de empleo, la detección de condiciones médicas y la influencia en las sentencias penales a través de la predicción de reincidencia. En consecuencia, la tecnología de IA puede tener impactos sociales significativos y consideraciones éticas que deben entenderse y considerarse al desarrollarla y aplicarla.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.1.1 Introducción a la IA: Conceptos y Comportamiento Inteligente (Pág. 39)	2	1
⭐ 3.1.2 Características del Problema y Naturaleza de los Agentes (Pág. 39)	1	1
⭐ 3.1.3 Estado y Búsqueda No Informada (Pág. 39)	1	2
⭐ 3.1.4 Búsqueda Heurística y CSP (Pág. 40)	1	2
⭐ 3.1.5 Búsqueda Adversarial y Optimización Estocástica (Pág. 40)	1	3
⭐ 3.1.6 Representación del Conocimiento y Razonamiento Probabilístico Básico (Pág. 41)	1	1
⭐ 3.1.7 Independencia, Modelos de Markov y Toma de Decisiones (Pág. 41)	1	1
⭐ 3.1.8 Fundamentos de Aprendizaje Automático (Pág. 41)	1	2
⭐ 3.1.9 Aprendizaje Automático: Datos, Representaciones y Evaluación (Pág. 42)	1	2
⭐ 3.1.10 Redes Neuronales y Ética (Pág. 43)	1	1
⭐ 3.1.11 Formulación del Aprendizaje Automático y Evaluación (Pág. 43)	1	2
⭐ 3.1.12 Aprendizaje Profundo, RL y Ética (Pág. 43)	1	2
⭐ 3.1.13 Entrenamiento y Optimización de Redes Neuronales (Pág. 44)	1	1
⭐ 3.1.14 Técnicas de Regularización para Redes Profundas (Pág. 44)	1	1
⭐ 3.1.15 Arquitecturas de Redes Convolucionales y Residuales (Pág. 44)	1	1
⭐ 3.1.16 Arquitecturas Transformer y Modelos Generativos (Pág. 44)	1	1
⭐ 3.1.17 Aplicaciones e Impacto Social de la IA (Pág. 45)	2	2
⭐ 3.1.18 Aplicaciones Adicionales e Impacto Social de la IA (Pág. 45)	2	2
3.1.19 Representación Lógica y Razonamiento (Pág. 46)	Electivo	
3.1.20 Representación Probabilística y Razonamiento (Pág. 46)	Electivo	
3.1.21 Planificación (Pág. 47)	Electivo	
3.1.22 Agentes y Sistemas Cognitivos (Pág. 47)	Electivo	
3.1.23 Ingeniería de Bucles de Agentes (Loop Engineering) (Pág. 48)	Electivo	
3.1.24 Procesamiento del Lenguaje Natural (Pág. 48)	Electivo	
3.1.25 Robótica (Pág. 49)	Electivo	
3.1.26 Percepción y Visión por Computadora (Pág. 50)	Electivo	

Cuadro 3.1: Lista de KUs del área de Inteligencia Artificial.

3.1.1. AI/Introducción a la IA: Conceptos y Comportamiento Inteligente (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Visión general de los problemas de IA, Ejemplos de aplicaciones recientes exitosas de IA
- Definiciones de agentes con ejemplos (por ejemplo, reactivos, deliberativos)
- ¿Qué es el comportamiento inteligente?: 1. La prueba de Turing y sus defectos 2. Entrada y salida multimodal 3. Simulación del comportamiento inteligente 4. Razonamiento racional versus no racional
- Visión general de las Aplicaciones de IA, crecimiento e impacto (económico, social, ético)

Non Core

- Problemas filosóficos
- Historia de la IA

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la prueba de Turing y el experimento mental de la "Habitación China" [Explicar]
2. Diferenciar entre razonamiento/comportamiento óptimo y razonamiento/comportamiento similar al humano [Evaluar]
3. Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Explicar]

3.1.2. AI/Características del Problema y Naturaleza de los Agentes (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Características del problema: 1. Totalmente observable versus parcialmente observable 2. Agente único versus multiagente 3. Determinista versus estocástico 4. Estático versus dinámico 5. Discreto versus continuo
- Naturaleza de los agentes: 1. Autonomía autónoma, semiautónoma, de iniciativa mixta 2. Reflexivos, basados en metas y basados en utilidad 3. Toma de decisiones bajo incertidumbre y con información incompleta 4. La importancia de la percepción y las interacciones ambientales 5. Agentes basados en aprendizaje 6. Agentes corporeizados: *a*) sensores, dinámicas, efectores
- Practicar la identificación de características del problema en entornos de ejemplo
- Profundidad adicional sobre la naturaleza de los agentes con ejemplos
- Profundidad adicional sobre Aplicaciones de IA, Crecimiento e Impacto (económico, social, ético, seguridad)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar las características de un problema específico [Aplicar]

3.1.3. AI/Estado y Búsqueda No Informada (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Representación del espacio de estados de un problema: 1. Especificación de estados, metas y operadores 2. Factorización de estados en representaciones (espacios de hipótesis) 3. Resolución de problemas mediante búsqueda en grafos: *a*) por ejemplo, Grafos como un espacio y recorridos de árboles como exploración de ese espacio *b*) Construcción dinámica del grafo (no dada de antemano)
- Búsqueda en grafo no informada para resolución de problemas: 1. Búsqueda en amplitud 2. Búsqueda en profundidad: *a*) Con profundización iterativa 3. Búsqueda de costo uniforme
- Complejidades de espacio y tiempo de los algoritmos de búsqueda en grafos
- Búsqueda bidireccional

Non Core

- Comprensión del espacio de búsqueda: 1. Construcción de árboles de búsqueda 2. Espacios de búsqueda dinámicos 3. Explosión combinatoria del espacio de búsqueda 4. Topología del espacio de búsqueda (por ejemplo, crestas, puntos de silla, mínimos locales)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar la representación del espacio de estados para un rompecabezas (por ejemplo, N-reinas o problema de los 3 jarrones) [Aplicar]
2. Seleccionar e implementar un algoritmo de búsqueda no informada apropiado para un problema (por ejemplo, tres en raya), y caracterizar sus complejidades de tiempo y espacio [Aplicar]

3.1.4. AI/Búsqueda Heurística y CSP (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Búsqueda en grafo heurística para resolución de problemas: 1. Construcción heurística y admisibilidad 2. Ascenso de colinas 3. Mínimos locales y el paisaje de búsqueda: *a*) Soluciones locales versus globales 4. Búsqueda primero el mejor voraz 5. Búsqueda A*
- Búsqueda en haz
- Implementación de la búsqueda A*
- Satisfacción de restricciones

Non Core

- Variaciones de A* (IDA*, SMA*, RBFS)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Seleccionar e implementar un algoritmo de búsqueda informada apropiado para un problema después de diseñar una función heurística útil (por ejemplo, un robot navegando en un mundo de cuadrícula 2D) [Aplicar]
2. Evaluar si una heurística para un problema dado es admisible/puede garantizar una solución óptima [Evaluar]
3. Diseñar e implementar búsqueda A*/en haz para resolver un problema, y compararla con otros algoritmos de búsqueda en términos del costo de la solución, número de nodos expandidos, etc [Aplicar]
4. Comparar y contrastar varias búsquedas heurísticas con respecto a la aplicabilidad a un problema dado [Evaluar]
5. Modelar un rompecabezas lógico o Sudoku como un problema de satisfacción de restricciones, resolverlo con búsqueda con retroceso y determinar cuánto la consistencia de arcos puede reducir el espacio de búsqueda [Aplicar]

3.1.5. AI/Búsqueda Adversarial y Optimización Estocástica (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Juegos adversariales de dos jugadores: 1. Búsqueda minimax 2. Poda alfa-beta: *a*) Corte por profundidad

Non Core

- Búsqueda local
- Búsqueda tabú
- Juegos adversariales de dos jugadores: 1. El efecto horizonte 2. Libros de apertura/soluciones de finales 3. Qué significa ”resolver” un juego (por ejemplo, damas)
- Implementación de búsqueda minimax, búsqueda en haz
- Búsqueda expectimax (resolución MDP) y nodos de azar
- Búsqueda estocástica: 1. Recocido simulado 2. Algoritmos genéticos 3. Búsqueda en árbol Monte Carlo

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar búsqueda minimax en un juego adversarial de dos jugadores (por ejemplo, conecta 4), usando evaluación heurística en una profundidad particular para calcular las puntuaciones a propagar [Aplicar]
2. Aplicar búsqueda minimax con poda alfa-beta para podar el espacio de búsqueda en un juego adversarial de dos jugadores (por ejemplo, conecta 4) [Aplicar]
3. Comparar y contrastar algoritmos genéticos con técnicas de búsqueda clásicas, explicando cuándo es más apropiado usar un algoritmo genético para aprender un modelo versus otras formas de optimización (por ejemplo, descenso de gradiente) [Evaluar]

NonCore:

4. Diseñar e implementar una solución de algoritmo genético para un problema [Aplicar]
5. Diseñar e implementar un programa de recocido simulado para evitar mínimos locales en un problema [Aplicar]

3.1.6. AI/Representación del Conocimiento y Razonamiento Probabilístico Básico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Tipos de representaciones: 1. Simbólicas, lógicas: *a*) Crear una representación a partir de un enunciado de problema en lenguaje natural 2. Representaciones subsimbólicas aprendidas 3. Modelos gráficos (por ejemplo, Bayes ingenuo, red bayesiana)
- Repaso del razonamiento probabilístico, teorema de Bayes
- Razonamiento bayesiano: 1. Inferencia bayesiana
- Variables aleatorias y distribuciones de probabilidad: 1. Axiomas de probabilidad 2. Inferencia probabilística 3. Regla de Bayes (derivación) 4. Inferencia bayesiana (ejemplos más complejos)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Dado un enunciado de problema en lenguaje natural, codificarlo como una representación simbólica o lógica [Aplicar]
2. Calcular una inferencia probabilística en un problema del mundo real usando el teorema de Bayes para determinar la probabilidad de una hipótesis dada la evidencia [Aplicar]
3. Aplicar la regla de Bayes para determinar la probabilidad de una hipótesis dada la evidencia [Aplicar]
4. Calcular la probabilidad de resultados y probar si los resultados son independientes [Aplicar]

3.1.7. AI/Independencia, Modelos de Markov y Toma de Decisiones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Independencia
- Independencia condicional
- Cadenas de Markov y modelos de Markov
- Utilidad y toma de decisiones

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar cómo podemos tomar decisiones bajo incertidumbre, usando conceptos como el teorema de Bayes y la utilidad [Explicar]

3.1.8. AI/Fundamentos de Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:

Core

- Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: 1. Aprendizaje supervisado: *a*) Clasificación *b*) Regresión 2. Aprendizaje por refuerzo 3. Aprendizaje no supervisado: *a*) Agrupamiento
- Ideas fundamentales: 1. Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. 2. Fuentes de error e indecidibilidad en el aprendizaje automático
- Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: 1. Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente
- El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): 1. El compromiso entre sesgo (subajuste) - varianza (sobreajuste)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar]
2. Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar]
3. Formular una aplicación como un problema de clasificación, incluyendo las características de entrada disponibles y la salida a predecir (por ejemplo, identificar caracteres alfabéticos a partir de entrada de cuadrícula de píxeles) [Aplicar]
4. Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar]
5. Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

3.1.9. AI/Aprendizaje Automático: Datos, Representaciones y Evaluación (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Trabajar con Datos: 1. Preprocesamiento de datos: *a*) Importancia y peligros de las elecciones de preprocesamiento 2. Manejo de valores faltantes (imputación, marcar como faltante): *a*) Implicaciones de imputar versus marcar como faltante 3. Codificación de variables categóricas, codificación de datos de valor real 4. Normalización/estandarización 5. énfasis en datos reales, no ejemplos de libros de texto
- Representaciones: 1. Espacios de hipótesis y complejidad 2. Expansión de características de base simple, como elevar al cuadrado características univariadas 3. Representaciones de características aprendidas
- Evaluación del aprendizaje automático: 1. Separación de conjuntos de entrenamiento, validación y prueba 2. Métricas de rendimiento para clasificadores 3. Estimación del rendimiento en prueba en datos retenidos 4. Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación 5. Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar]
2. Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar]
3. Explicar los procedimientos adecuados de evaluación de AA, incluyendo las diferencias entre rendimiento en entrenamiento y prueba, y qué puede salir mal con el proceso de evaluación llevando a informes inexactos del rendimiento de AA [Explicar]
4. Comparar dos algoritmos de aprendizaje automático en un conjunto de datos, implementando la metodología de preprocesamiento de datos y evaluación (por ejemplo, métricas y manejo de divisiones entrenamiento/prueba) desde cero [Aplicar]

3.1.10. AI/Redes Neuronales y Ética (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Redes neuronales básicas: 1. Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos 2. Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes)
- ética para el Aprendizaje Automático: 1. Enfocarse en datos reales, escenarios reales y estudios de caso 2. Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]

3.1.11. AI/Formulación del Aprendizaje Automático y Evaluación (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: 1. Función objetivo 2. Descenso de gradiente 3. Regularización para evitar sobreajuste (formulación matemática)
- Conjuntos de modelos: 1. Combinación de mayoría ponderada simple
- Evaluación de rendimiento: 1. Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) 2. Métricas de rendimiento para regresores 3. Matriz de confusión 4. Validación cruzada: *a*) Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar y contrastar varias técnicas de aprendizaje (por ejemplo, árboles de decisión, regresión logística, Bayes ingenuo, redes neuronales y redes de creencia), proporcionando ejemplos de cuándo cada estrategia es superior [Comparar]
2. Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar]

NonCore:

3. Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar]
4. Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar]
5. Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]

3.1.12. AI/Aprendizaje Profundo, RL y Ética (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Aprendizaje profundo: 1. Redes feed-forward profundas (solo intuición, sin matemáticas) 2. Redes neuronales convolucionales (solo intuición, sin matemáticas) 3. Visualización de representaciones de características aprendidas de redes profundas 4. Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.)
- Visión general de los métodos de aprendizaje por refuerzo
- Dos o más aplicaciones de algoritmos de aprendizaje automático: 1. Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego
- ética para el Aprendizaje Automático: 1. Continuar enfocándose en datos reales, escenarios reales y estudios de caso 2. Privacidad 3. Equidad 4. Propiedad intelectual 5. Explicabilidad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Visualizar el progreso del entrenamiento de una red neuronal a través de curvas de aprendizaje en un kit de herramientas establecido (por ejemplo, TensorBoard) y visualizar las características aprendidas de la red [Aplicar]

NonCore:

2. Dada una aplicación real de aprendizaje automático, describir problemas éticos respecto a las elecciones de datos, pasos de preprocesamiento, selección de algoritmo y visualización/presentación de resultados [Aplicar]

3.1.13. AI/Entrenamiento y Optimización de Redes Neuronales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Backpropagation moderno: grafos computacionales y diferenciación automática
- Optimizadores avanzados: momentum, RMSProp, Adam, y programación de la tasa de aprendizaje

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar una red neuronal profunda desde cero, incluyendo su paso de backpropagation [Aplicar]
2. Seleccionar y ajustar un optimizador y su programación de tasa de aprendizaje para la estabilidad del entrenamiento [Aplicar]

3.1.14. AI/Técnicas de Regularización para Redes Profundas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Regularización por Dropout
- Normalización por lotes (Batch Normalization)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diagnosticar el sobreajuste en un entrenamiento usando diagnósticos de regularización [Analizar]
2. Aplicar Dropout y Batch Normalization para mejorar la generalización de un modelo [Aplicar]

3.1.15. AI/Arquitecturas de Redes Convolucionales y Residuales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Arquitecturas de redes neuronales convolucionales profundas
- Conexiones residuales y ResNets

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar una arquitectura CNN profunda para una tarea de visión dada [Aplicar]
2. Adaptar una arquitectura residual a un dominio de visión específico [Aplicar]

3.1.16. AI/Arquitecturas Transformer y Modelos Generativos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Transformers y el mecanismo de atención

- Modelos generativos: autoencoders variacionales (VAEs) y redes generativas adversariales (GANs)

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Entrenar un modelo transformer en un dataset real [Aplicar]
2. Analizar y contrastar arquitecturas de modelos generativos VAE y GAN [Analizar]

3.1.17. AI/Aplicaciones e Impacto Social de la IA (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core
- Al menos una aplicación de IA a un problema y campo específico, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir al menos uno para el Núcleo CS): 1. Formular y evaluar una aplicación específica como un problema de IA: *a*) Cómo lidiar con problemas mal especificados o mal planteados 2. Disponibilidad/escasez y limpieza de datos: *a*) Limpieza y preprocesamiento básico de datos *b*) Sesgo del conjunto de datos 3. Sesgo algorítmico 4. Sesgo de evaluación 5. Evaluación de las implicaciones sociales de la aplicación
- Modelos generativos profundos desplegados: 1. Visión general de alto nivel de modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion, etc.), sus usos y sus limitaciones/peligros. 2. Visión general de alto nivel de modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard, etc.), sus usos y sus limitaciones/peligros.
- Visión general del impacto social de la IA: 1. ética 2. Equidad 3. Confianza/explicabilidad 4. Privacidad y uso de datos de entrenamiento 5. Autonomía humana y supervisión/regulaciones/requisitos legales 6. Sostenibilidad

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Dado un dominio de aplicación y problema del mundo real, formular una solución de IA para ello, identificando datos/entrada apropiados, preprocesamiento, representaciones, técnicas de IA y métricas/metodología de evaluación [Aplicar]

3.1.18. AI/Aplicaciones Adicionales e Impacto Social de la IA (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core
- Una o más aplicaciones adicionales de IA a un amplio conjunto de problemas y diversos campos, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir un área diferente de la elegida para el Núcleo CS): 1. Formular y evaluar una aplicación específica como un problema de IA: *a*) Cómo lidiar con problemas mal especificados o mal planteados 2. Disponibilidad/escasez y limpieza de datos: *a*) Limpieza y preprocesamiento básico de datos *b*) Sesgo del conjunto de datos 3. Sesgo algorítmico 4. Sesgo de evaluación 5. Evaluación de las implicaciones sociales de la aplicación
- Profundidad adicional sobre modelos generativos profundos desplegados: 1. Introducción a cómo funcionan los modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion) incluyendo discusión de atención 2. Introducción a cómo funcionan los modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard) incluyendo discusión de atención 3. Idea de modelos fundamentales, cómo usarlos, y los beneficios/problemas con entrenarlos a partir de grandes datos
- Análisis y discusión del impacto social de la IA: 1. ética 2. Equidad 3. Confianza/explicabilidad 4. Privacidad y uso de datos de entrenamiento 5. Autonomía humana y supervisión/regulaciones/requisitos legales 6. Sostenibilidad

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Analizar el impacto social de una o más aplicaciones específicas de IA del mundo real, identificando problemas respecto a ética, equidad, sesgo, confianza y explicabilidad [Evaluar]

2. Describir algunos de los modos de falla de los modelos generativos profundos actuales para lenguaje o imágenes, y cómo esto podría afectar su uso en una aplicación [Describir]

3.1.19. AI/Representación Lógica y Razonamiento

Temas:
Non Core

- Repaso de lógica proposicional y de predicados
- Resolución y demostración de teoremas (solo lógica proposicional): 1. Encadenamiento hacia adelante, encadenamiento hacia atrás
- Problemas de representación del conocimiento: 1. Lógicas de descripción 2. Ingeniería de ontologías
- Web semántica
- Razonamiento no monótono (por ejemplo, lógicas no clásicas, razonamiento por defecto)
- Argumentación
- Razonamiento sobre acción y cambio (por ejemplo, cálculo de situaciones y eventos)
- Razonamiento temporal y espacial
- Programación lógica: 1. Prolog, Programación de conjuntos de respuestas
- Sistemas expertos basados en reglas
- Redes semánticas
- Razonamiento basado en modelos y basado en casos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Traducir una oración en lenguaje natural (por ejemplo, inglés) a una declaración de lógica de predicados [Aplicar]
2. Convertir una declaración lógica a forma causal [Aplicar]
3. Aplicar resolución a un conjunto de declaraciones lógicas para responder a una consulta [Aplicar]
4. Comparar y contrastar los modelos más comunes utilizados para la representación estructurada del conocimiento, destacando sus fortalezas y debilidades [Evaluar]
5. Identificar los componentes del razonamiento no monótono y su utilidad como mecanismo representacional para sistemas de creencias [Evaluar]
6. Comparar y contrastar las técnicas básicas para representar la incertidumbre [Evaluar]
7. Comparar y contrastar las técnicas básicas para la representación cualitativa [Evaluar]
8. Aplicar cálculo de situaciones y eventos a problemas de acción y cambio [Aplicar]
9. Explicar la distinción entre razonamiento temporal y espacial, y cómo se interrelacionan [Explicar]
10. Explicar la diferencia entre técnicas de razonamiento basadas en reglas, basadas en casos y basadas en modelos [Explicar]
11. Definir el concepto de un sistema de planificación y cómo difiere de las técnicas de búsqueda clásicas [Definir]
12. Describir las diferencias entre planificación como búsqueda, planificación basada en operadores y planificación proposicional, proporcionando ejemplos de dominios donde cada una es más aplicable [Describir]
13. Explicar la distinción entre inferencia monótona y no monótona [Explicar]

3.1.20. AI/Representación Probabilística y Razonamiento

Temas:
Non Core

- Repaso de independencia condicional
- Representaciones del conocimiento: 1. Redes bayesianas: *a*) Inferencia exacta y su complejidad *b*) Mantas de Markov y d-separación *c*) Métodos de muestreo aleatorizado (Monte Carlo) (por ejemplo, muestreo de Gibbs) 2. Redes de Markov 3. Modelos de probabilidad relacional 4. Modelos ocultos de Markov
- Teoría de la decisión: 1. Preferencias y funciones de utilidad 2. Maximización de la utilidad esperada 3. Teoría de juegos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Calcular la probabilidad de una hipótesis dada la evidencia en una red bayesiana [Aplicar]
2. Explicar cómo las afirmaciones de independencia condicional permiten una mayor eficiencia de los sistemas probabilísticos [Explicar]
3. Identificar ejemplos de representaciones del conocimiento para razonar bajo incertidumbre [Listar/Enumerar]
4. Establecer la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Enunciar/Establecer]
5. Diseñar e implementar al menos una representación del conocimiento para razonar bajo incertidumbre [Aplicar]
6. Describir las complejidades del razonamiento probabilístico temporal [Describir]
7. Diseñar e implementar un HMM como un ejemplo de sistema probabilístico temporal [Aplicar]
8. Describir la relación entre preferencias y funciones de utilidad [Describir]
9. Explicar cómo las funciones de utilidad y el razonamiento probabilístico pueden combinarse para tomar decisiones racionales [Explicar]

3.1.21. AI/Planificación

Temas:
Non Core

- Repaso de lógica proposicional y de primer orden
- Operadores de planificación y representaciones de estado
- Planificación de orden total
- Planificación de orden parcial
- Grafos de plan y GraphPlan
- Planificación jerárquica
- Lenguajes y representaciones de planificación: 1. PDDL
- Planificación multiagente
- Planificación basada en MDP
- Interconexión de planificación, ejecución y replanificación dinámica: 1. Planificación condicional 2. Planificación continua 3. Planificación probabilística

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Construir la representación de estado, meta y operadores para un problema de planificación dado [Aplicar]
2. Codificar un problema de planificación en PDDL y usar un planificador para resolverlo [Aplicar]
3. Dado un conjunto de operadores, estado inicial y estado meta, dibujar el grafo de planificación de orden parcial e incluir restricciones de orden para resolver todos los conflictos [Aplicar]
4. Construir el grafo de planificación completo para que GraphPlan resuelva un problema dado [Aplicar]

3.1.22. AI/Agentes y Sistemas Cognitivos

Temas:
Non Core

- Arquitecturas de agentes (por ejemplo, reactivas, en capas, cognitivas)
- Teoría de agentes (incluyendo formalismos matemáticos)
- Racionalidad, Teoría de juegos: 1. Agentes decisorios-teóricos 2. Procesos de decisión de Markov (MDP) 3. Algoritmos de bandidos
- Agentes de software, asistentes personales y acceso a información: 1. Agentes colaborativos 2. Agentes de recolección de información 3. Agentes creíbles (personajes sintéticos, modelado de emociones en agentes)
- Agentes de aprendizaje
- Sistemas cognitivos: 1. Arquitecturas cognitivas (por ejemplo, ACT-R, SOAR, ICARUS, FORR) 2. Capacidades (por ejemplo, percepción, toma de decisiones, predicción, mantenimiento del co-

nocimiento) 3. Representación, organización, utilización, adquisición y refinamiento del conocimiento 4. Aplicaciones y evaluación de sistemas cognitivos
- Sistemas multiagente: 1. Agentes colaboradores 2. Equipos de agentes 3. Agentes competitivos (por ejemplo, subastas, votación) 4. Sistemas de enjambre y modelos biológicamente inspirados 5. Aprendizaje multiagente
- Interacción humano-agente: 1. Metodologías de comunicación (verbal y no verbal) 2. Problemas prácticos 3. Aplicaciones: *a*) Agentes comerciales, gestión de cadena de suministro *b*) Problemas éticos de las interacciones de IA con humanos *c*) Regulación y requisitos legales de los sistemas de IA para interactuar con humanos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:
1. Caracterizar y contrastar las arquitecturas estándar de agentes [Evaluar]
2. Describir las aplicaciones de la teoría de agentes a dominios como agentes de software, asistentes personales y agentes creíbles, y discutir las implicaciones éticas asociadas [Describir]
3. Describir los paradigmas principales utilizados por los agentes de aprendizaje [Describir]
4. Demostrar usando ejemplos apropiados cómo los sistemas multiagente soportan la interacción de agentes [Demostrar]
5. Construir un agente inteligente usando una arquitectura cognitiva establecida (ACT-R, SOAR) para resolver un problema específico [Aplicar]

3.1.23. AI/Ingeniería de Bucles de Agentes (Loop Engineering)

La disciplina de diseñar el ciclo iterativo observar-actuar-verificar que ejecuta un agente de IA, en contraste con la ingeniería de un único prompt. Incluye patrones de bucle con múltiples agentes especializados (constructor/crítico) que refinan un resultado contra un estándar de calidad predefinido hasta su aprobación.

Temas:
Non Core
- La ingeniería de bucles como disciplina: diseñar el ciclo iterativo observar-actuar-verificar que ejecuta un agente, en contraste con escribir un único prompt
- El patrón Gauntlet Loop (Matt Shumer, 2026): un agente líder descompone un objetivo y un estándar de calidad en las partes más pequeñas y evaluables posibles, y despliega agentes constructores y agentes críticos con contexto limpio
- Separación constructor/crítico: un agente que crea un resultado suele ser peor juzgando objetivamente su propia calidad que explicándola, de ahí la necesidad de un crítico independiente
- Comparación ciega A/B del resultado contra un estándar de calidad predefinido, y criterios de parada del bucle (aprobación del crítico o límite de iteraciones)
- Relación con otros patrones de bucles agénticos: ReAct (razonar-actuar), auto-reflexión y debate multiagente

Aprendizaje esperado (*Learning Outcomes*):
NonCore:
1. Explicar en qué se diferencia la ingeniería de bucles de la ingeniería de un único prompt [Explicar]
2. Diseñar un bucle de tipo Gauntlet Loop (agente constructor + agente crítico independiente + comparación ciega contra un estándar de calidad) para refinar iterativamente un resultado generado por IA [Diseñar]
3. Justificar por qué separar los roles de constructor y crítico mejora la calidad de la evaluación frente a que un mismo agente juzgue su propio resultado [Justificar]
4. Definir criterios de parada apropiados para un bucle agéntico iterativo dado un estándar de calidad y un presupuesto de iteraciones [Definir]

3.1.24. AI/Procesamiento del Lenguaje Natural

Temas:
Non Core
- Gramáticas deterministas y estocásticas

- Algoritmos de análisis sintáctico: 1. CFGs y analizadores de tabla (por ejemplo, CYK) 2. CFGs probabilísticos y CYK ponderado
- Representación del significado/Semántica: 1. Representaciones del conocimiento basadas en lógica 2. Roles semánticos 3. Representaciones temporales 4. Creencias, deseos e intenciones
- Métodos basados en corpus
- N-gramas y HMMs
- Suavizado y retroceso
- Ejemplos de uso: Etiquetado POS y morfología
- Recuperación de información: 1. Modelo de espacio vectorial: *a*) TF & IDF 2. Precisión y exhaustividad
- Extracción de información
- Traducción de lenguaje
- Clasificación de texto, categorización: 1. Modelo de bolsa de palabras
- Aprendizaje profundo para PLN: 1. RNNs 2. Transformers 3. Incrustaciones multimodales (por ejemplo, imágenes + texto) 4. Modelos de lenguaje generativos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Definir y contrastar gramáticas deterministas y estocásticas, proporcionando ejemplos para mostrar la adecuación de cada una [Definir]
2. Simular, aplicar o implementar algoritmos clásicos y estocásticos para analizar lenguaje natural [Aplicar]
3. Identificar los desafíos de representar el significado [Analizar]
4. Listar las ventajas de usar corpus estándar. Identificar ejemplos de corpus actuales para una variedad de tareas de PLN [Listar/Enumerar]
5. Identificar técnicas para recuperación de información, traducción de lenguaje y clasificación de texto [Analizar]
6. Implementar una transformación TF/IDF, usarla para extraer características de un corpus, y entrenar un algoritmo de aprendizaje automático estándar usando esas características para hacer clasificación de texto [Aplicar]

3.1.25. AI/Robótica

Temas:
Non Core

- Visión general: problemas y progreso: 1. Sistemas robóticos de última generación, incluyendo sus sensores y una visión general de su procesamiento sensorial 2. Arquitecturas de control de robots, por ejemplo, control deliberativo versus reactivo y vehículos de Braitenberg 3. Modelado del mundo y modelos del mundo 4. Incertidumbre inherente en la percepción y en el control
- Sensores y efectores: 1. Sensores: por ejemplo, LIDAR, sonar, visión, profundidad, estereoscópico, cámaras de eventos, micrófonos, háptica 2. Efectores: por ejemplo, ruedas, brazos, pinzas
- Marcos de coordenadas, traslación y rotación (2D y 3D)
- Espacio de configuración y mapas ambientales
- Interpretación de datos sensoriales inciertos
- Localización y mapeo
- Navegación y control
- Cinemática directa e inversa
- Planificación de trayectorias de movimiento y optimización de trayectorias
- Manipulación y agarre
- Control y dinámica de articulaciones
- Control basado en visión
- Coordinación y colaboración múltiple de robots
- Interacción humano-robot: 1. Espacios de trabajo compartidos 2. Equipos humano-robot e HRI físico 3. Robots de asistencia social 4. Predicción de movimiento/tarea/meta 5. Colaboración y comunicación (explícita vs implícita, verbal o simbólica vs no verbal o visual) 6. Confianza
- Aplicaciones y Problemas Sociales, Económicos y éticos: 1. Implicaciones sociales, económicas, de derecho al trabajo 2. Implicaciones éticas y de privacidad de las aplicaciones robóticas 3. Res-

ponsabilidad en robótica autónoma 4. Armas autónomas y ética 5. Supervisión y control humano

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Listar capacidades y limitaciones de los sistemas robóticos de última generación actuales, incluyendo sus sensores y el procesamiento sensorial crucial que informa a esos sistemas [Listar/Enumerar]
2. Integrar sensores, actuadores y software en un robot diseñado para emprender una tarea específica [Aplicar]
3. Programar un robot para realizar tareas simples usando arquitecturas de control deliberativas, reactivas y/o híbridas [Aplicar]
4. Implementar algoritmos fundamentales de planificación de movimiento dentro de un espacio de configuración de robot [Aplicar]
5. Caracterizar las incertidumbres asociadas con sensores y actuadores robóticos comunes; articular estrategias para mitigar estas incertidumbres [Caracterizar]
6. Listar las diferencias entre las representaciones del entorno externo de los robots, incluyendo sus fortalezas y limitaciones [Listar/Enumerar]
7. Comparar y contrastar al menos tres estrategias para la navegación de robots dentro de entornos conocidos y/o desconocidos, incluyendo sus fortalezas y limitaciones [Comparar]
8. Describir al menos un enfoque para coordinar las acciones y percepción de varios robots para realizar una única tarea [Describir]
9. Comparar y contrastar un enfoque de coordinación multirobot y un enfoque de colaboración humano-robot y atribuir sus diferencias a diferencias entre los entornos del problema [Comparar]
10. Analizar los problemas sociales, económicos y éticos de una aplicación robótica del mundo real [Analizar]

3.1.26. AI/Percepción y Visión por Computadora

Temas:
Non Core

- Visión por computadora: 1. Adquisición, representación, procesamiento y propiedades de imágenes 2. Representación de formas, reconocimiento de objetos y segmentación 3. Análisis de movimiento 4. Modelos generativos
- Audio y reconocimiento de voz
- Tacto y propiocepción
- Otras modalidades (por ejemplo, olfato)
- Modularidad en el reconocimiento
- Enfoques para el reconocimiento de patrones: 1. Algoritmos de clasificación y medidas de calidad de clasificación 2. Técnicas estadísticas 3. Técnicas de aprendizaje profundo

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir]
2. Listar al menos tres enfoques de segmentación de imágenes, como umbralización, algoritmos basados en bordes y basados en regiones, junto con sus características definitorias, fortalezas y debilidades [Listar/Enumerar]
3. Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar]
4. Distinguir los objetivos del reconocimiento de sonido, reconocimiento de voz y reconocimiento de hablante, e identificar cómo se manejará diferente la señal de audio cruda en cada uno de estos casos [Distinguir]
5. Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, por ejemplo, datos táctiles interpretados como imágenes 2D de banda única [Generar]
6. Implementar un algoritmo de extracción de características en datos reales, por ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier que describen

un segmento corto de señal de audio [Aplicar]
7. Implementar un algoritmo que combine características en percepciones de nivel superior, por ejemplo, un contorno o polígono a partir de primitivas visuales o hipótesis de fonemas a partir de una señal de audio [Aplicar]
8. Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar]
9. Evaluar el rendimiento de la extracción de características subyacente, relativo a al menos un enfoque alternativo posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8), anterior [Evaluar]
10. Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir]
11. Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]

3.2. Fundamentos Algorítmicos (AL)

Los algoritmos y las estructuras de datos son fundamentales para la informática, ya que todo cálculo teórico y programa aplicado consiste en algoritmos que operan sobre elementos de datos que poseen alguna estructura subyacente. Seleccionar soluciones computacionales apropiadas para problemas del mundo real se beneficia de comprender las capacidades y limitaciones teóricas y prácticas de los algoritmos y paradigmas disponibles, incluyendo su impacto en el medio ambiente y la sociedad. Además, esta comprensión proporciona una visión de la naturaleza intrínseca de la computación, los problemas computacionales y la resolución de problemas computacionales, así como de las posibles técnicas de solución independientes del lenguaje de programación, el paradigma de programación, el hardware de la computadora u otros aspectos de implementación.

Esta área de conocimiento se centra en la naturaleza de la computación, incluyendo los conceptos y habilidades necesarios para diseñar y analizar algoritmos para resolver problemas computacionales del mundo real. Complementa la implementación de algoritmos y estructuras de datos que se encuentran en el área de conocimiento de Fundamentos de Desarrollo de Software (SDF). Como los algoritmos y las estructuras de datos son esenciales en todas las áreas avanzadas de la informática, esta área proporciona los fundamentos algorítmicos que se espera que todo graduado en informática conozca. La exposición a la amplitud de estos temas fundamentales de AL está diseñada para proporcionar a los estudiantes la base para estudiar estos temas en mayor profundidad, para estudiar temas adicionales de computación y algoritmos, y para aprender algoritmos avanzados en una variedad de áreas de conocimiento de CS y disciplinas CS+X.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.2.1 Estructuras de Datos Fundamentales (Pág. 52)	5	3
⭐ 3.2.2 Algoritmos Fundamentales (Pág. 52)	4	2
⭐ 3.2.3 Algoritmos Avanzados (Pág. 53)	3	2
⭐ 3.2.4 Estrategias Algorítmicas (Pág. 53)	6	
⭐ 3.2.5 Marco de Análisis de Complejidad (Pág. 54)	2	1
⭐ 3.2.6 Notación Asintótica y Clases de Complejidad (Pág. 54)	1	1
⭐ 3.2.7 Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (Pág. 55)	2	1
⭐ 3.2.8 Teoría de la Complejidad Computacional (Pág. 55)	2	1
⭐ 3.2.9 Lenguajes Formales y Autómatas (Pág. 55)	4	10
⭐ 3.2.10 Computabilidad y Decidibilidad (Pág. 56)	6	14
3.2.11 Sociedad, ética y la Profesión (Pág. 57)	Electivo	

Cuadro 3.2: Lista de KUs del área de Fundamentos Algorítmicos.

3.2.1. AL/Estructuras de Datos Fundamentales (Core Tier1: 5 hrs, Core Tier2: 3 hrs)

Temas:
Core

- Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: 1. Operaciones de diccionario (insertar, eliminar, encontrar)
- Arreglos: 1. Numéricos vs no numéricos, cadenas de caracteres 2. Unidimensionales (vector) vs multidimensionales (matriz)
- Registros/Estructuras/Tuplas y Objetos
- Listas enlazadas (por razones históricas): 1. Simples vs Dobles y Lineales vs Circulares
- Pilas
- Colas y deques: 1. Cola de prioridad basada en montículo
- Tablas/mapas hash: 1. Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash)
- árboles: 1. Binarios, n-arios y árboles de búsqueda 2. Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo)
- Conjuntos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Para cada ADT/Estructura de Datos en esta unidad: *a*) Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. *b*) Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Explicar]
2. Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Explicar]
3. Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Explicar]

3.2.2. AL/Algoritmos Fundamentales (Core Tier1: 4 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Grafos (por ejemplo, [no]dirigidos, [a]cíclicos, [no]conexos y [no]ponderados): 1. Representación de grafos: lista de adyacencia vs matriz
- Algoritmos de búsqueda: 1. Complejidad $O(n)$ (por ejemplo, búsqueda lineal/secuencial en arreglo/lista) 2. Complejidad $O(\log_2 n)$ (por ejemplo, búsqueda binaria) 3. Complejidad $O(\log_b n)$ (por ejemplo, búsqueda en árbol no informada en profundidad/amplitud)
- Algoritmos de ordenamiento (por ejemplo, estables, inestables): 1. Complejidad $O(n^2)$ (por ejemplo, inserción, selección) 2. Complejidad $O(n \log n)$ (por ejemplo, quicksort, merge, timesort)
- Algoritmos de grafos: 1. Camino más corto (por ejemplo, Dijkstra, Floyd) 2. árbol de expansión mínima (por ejemplo, Prim, Kruskal)
- Algoritmos de ordenamiento: 1. Complejidad $O(n \log n)$ heapsort 2. Pseudo $O(n)$ complejidad (por ejemplo, bucket, counting, radix)
- Algoritmos de grafos: 1. Clausura transitiva (por ejemplo, Warshall) 2. Ordenamiento topológico
- Emparejamiento: 1. Emparejamiento eficiente de cadenas (por ejemplo, Boyer-Moore, Knuth-Morris-Pratt) 2. Emparejamiento de subsecuencia común más larga 3. Emparejamiento de expresiones regulares

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar]
2. Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar]
3. Dados los requisitos para un problema, desarrollar múltiples soluciones usando varias estructuras de datos y algoritmos. Posteriormente, evaluar la idoneidad, fortalezas y debilidades seleccionando un enfoque que satisfaga mejor los requisitos [Crear]

4. Explicar factores más allá de la eficiencia computacional que influyen en la elección de algoritmos, como el tiempo de programación, la mantenibilidad y el uso de patrones específicos de la aplicación en los datos de entrada [Explicar]
5. Para cada uno de los algoritmos y enfoques algorítmicos en los temas del Núcleo KA: *a*) Explicar un ejemplo prototípico del algoritmo, y *b*) Explicar paso a paso cómo opera el algoritmo. [Explicar]

3.2.3. AL/Algoritmos Avanzados (Core Tier1: 3 hrs, Core Tier2: 2 hrs)

Temas:
Non Core

- Algoritmos de criptografía (por ejemplo, SHA-256)
- Algoritmos paralelos
- Algoritmos de consenso (por ejemplo, Blockchain): 1. Prueba de trabajo vs prueba de participación
- Algoritmos de computación cuántica: 1. Basados en oráculo (por ejemplo, Deutsch-Jozsa, Bernstein-Vazirani, Simon) 2. Aceleración superpolinomial mediante QFT (por ejemplo, Shor) 3. Aceleración polinomial mediante amplificación de amplitud (por ejemplo, Grover)
- Algoritmo de Transformada Rápida de Fourier (FFT)
- Algoritmo de evolución diferencial

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Una apreciación de la computación cuántica y su aplicación a ciertos problemas [Explicar]

3.2.4. AL/Estrategias Algorítmicas (Core Tier1: 6 hrs)

Temas:
Core

- Paradigmas: 1. Fuerza Bruta (por ejemplo, búsqueda lineal, ordenamiento por selección, viajante de comercio, mochila) 2. Disminuir y Vencer: *a*) Por una Constante (por ejemplo, ordenamiento por inserción, ordenamiento topológico) *b*) Por un Factor Constante (por ejemplo, búsqueda binaria) *c*) Por un Tamaño Variable (por ejemplo, Euclides) 3. Dividir y Vencer (por ejemplo, búsqueda binaria, quicksort, mergesort, Strassen) 4. Voraz (por ejemplo, Dijkstra, Kruskal, Mochila) 5. Transformar y Vencer: *a*) Simplificación de instancia (por ejemplo, encontrar duplicados mediante preordenamiento de lista) *b*) Cambio de representación (por ejemplo, heapsort) *c*) Reducción de problema (por ejemplo, mínimo común múltiplo, programación lineal) *d*) Programación dinámica (por ejemplo, Floyd, Marshall, Bellman-Ford) 6. Compromisos espacio vs tiempo (por ejemplo, hash)
- Manejo del crecimiento exponencial (por ejemplo, heurística A*, ramificación y poda, retroceso)
- Iteración vs recursión (por ejemplo, factorial, búsqueda en árbol)
- Paradigmas: 1. Algoritmos de aproximación 2. Mejora iterativa (por ejemplo, Ford-Fulkerson, simplex) 3. Algoritmos aleatorizados/estocásticos (por ejemplo, corte máximo, bolas y cubos)

Non Core

- Computación cuántica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Para cada uno de los paradigmas en esta unidad: *a*) Explicar sus características definitorias *b*) Explicar un ejemplo que demuestre el paradigma incluyendo cómo este ejemplo satisface las características del paradigma. [Explicar]
2. Para cada uno de los algoritmos en la unidad Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, explicar el paradigma utilizado por el algoritmo y cómo ejemplifica este paradigma [Explicar]
3. Dado un algoritmo, explicar el paradigma utilizado por el algoritmo y cómo ejemplifica este paradigma [Explicar]

4. Dar un problema del mundo real, evaluar paradigmas algorítmicos apropiados y algoritmos de estos paradigmas que aborden el problema incluyendo evaluar los compromisos entre los paradigmas y algoritmos seleccionados [Evaluar]
5. Dar ejemplos de algoritmos iterativos y recursivos que resuelvan el mismo problema, explicar los beneficios y desventajas de cada enfoque [Explicar]
6. Evaluar si un enfoque voraz conduce a una solución óptima [Evaluar]
7. Explicar varios enfoques para abordar problemas computacionales cuyas soluciones algorítmicas son exponenciales [Explicar]

3.2.5. AL/Marco de Análisis de Complejidad (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Marco de Análisis de Complejidad: 1. Rendimiento de un algoritmo en el mejor caso, caso promedio y peor caso 2. Mediciones empíricas y relativas (Orden de Crecimiento) 3. Tamaño de entrada y operaciones primitivas 4. Eficiencia de tiempo y espacio
- Mediciones empíricas de rendimiento
- Compromisos tiempo-espacio en algoritmos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Preparar una presentación que explique a estudiantes de primer año los conceptos básicos de complejidad algorítmica incluyendo comportamiento de algoritmo en mejor caso, caso promedio y peor caso, notaciones Big-O, Omega y Theta, clases de complejidad, compromisos tiempo-espacio, medición empírica e impacto en problemas prácticos [Explicar]
2. Para cada algoritmo en la unidad Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, explicar su clase de complejidad de tiempo de ejecución y por qué pertenece a esta clase [Explicar]
3. Evaluar informalmente la clase de complejidad fundamental de algoritmos simples [Evaluar]
4. Desarrollar estudios empíricos para determinar y validar hipótesis sobre la complejidad de tiempo de ejecución de varios algoritmos ejecutando algoritmos con entradas de varios tamaños y comparando el rendimiento real con el análisis teórico [Crear]
5. Explicar ejemplos que ilustren los compromisos tiempo-espacio de algoritmos [Explicar]
6. Explicar cómo el balance del árbol afecta la eficiencia de las operaciones de árbol de búsqueda binaria [Explicar]

3.2.6. AL/Notación Asintótica y Clases de Complejidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Análisis de complejidad asintótica (cotas promedio y peor caso): 1. Notaciones formales Big-O, Big-Omega y Big-Theta 2. Clases de Complejidad Fundamentales y Ejemplos/Problemas Representativos: *a*) $O(1)$ Constante (por ejemplo, acceso a arreglo) *b*) $O(\log_2 n)$ Logarítmica (por ejemplo, búsqueda binaria) *c*) $O(n)$ Lineal (por ejemplo, búsqueda lineal) *d*) $O(n \log_2 n)$ Log Lineal (por ejemplo, mergesort) *e*) $O(n^2)$ Cuadrática (por ejemplo, ordenamiento por selección) *f*) $O(n^c)$ Polinomial (por ejemplo, $O(n^3)$ eliminación gaussiana) *g*) $O(2^n)$ Exponencial (por ejemplo, Mochila, Satisfactibilidad (SAT), Viajante de Comercio (TSP), todos los subconjuntos) *h*) $O(n!)$ Factorial (por ejemplo, circuito hamiltoniano, todas las permutaciones)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Usando ejemplos, explicar cada una de las clases de complejidad fundamentales en esta unidad [Explicar]
2. Para cada clase de complejidad fundamental en esta unidad, explicar un algoritmo que demuestre la complejidad de tiempo de ejecución asociada [Explicar]
3. Explicar a una audiencia no técnica la importancia de los algoritmos tratables versus intratables usando una explicación intuitiva de la complejidad Big-O [Explicar]

4. Aplicar la notación Big-O para dar cotas superiores en la complejidad tiempo/espacio de algoritmos [Aplicar]

3.2.7. AL/Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Notaciones Little-o, Little-Omega y Little Theta
- Análisis recursivo formal
- Análisis amortizado

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Dado un problema para programar para el cual puede haber varios enfoques algorítmicos, evaluarlos y determinar cuáles son factibles, y seleccionar uno que sea óptimo en implementación y comportamiento de tiempo de ejecución [Evaluar]
2. Usar relaciones de recurrencia para evaluar la complejidad de tiempo de algoritmos definidos recursivamente [Aplicar]
3. Aplicar relaciones de recurrencia elementales usando una forma del Teorema Maestro [Aplicar]

3.2.8. AL/Teoría de la Complejidad Computacional (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Tractabilidad e intractabilidad: 1. Clases de Complejidad P, NP y NP-Completo 2. Problemas NP-Completos (por ejemplo, SAT, Mochila, TSP) 3. Reducciones
- Modelos de complejidad basados en Máquinas de Turing: 1. Complejidad de tiempo: *a*) Clases P, NP, NP-C y EXP *b*) Teorema de Cook-Levin 2. Complejidad de espacio: *a*) NSpace y PSpace *b*) Teorema de Savitch

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la importancia de la NP-Completitud [Explicar]
2. Explicar cómo NP-Duro es una cota inferior y NP es una cota superior para la NP-Completitud [Explicar]
3. Explicar ejemplos de problemas NP-completos [Explicar]
4. Explicar el Teorema de Cook-Levin y la NP-Completitud de SAT [Explicar]
5. Explicar las clases P y NP [Explicar]
6. Demostrar que un problema es NP-Completo reduciendo un problema NP-C clásico conocido a él (por ejemplo, 3SAT y Clique) [Crear]
7. Explicar la clase P-espacio y su relación con la clase EXP [Explicar]

3.2.9. AL/Lenguajes Formales y Autómatas (Core Tier1: 4 hrs, Core Tier2: 10 hrs)

Temas:
Core

- Autómatas formales: 1. Estado Finito 2. Pila 3. Linealmente Acotado 4. Máquina de Turing
- Lenguajes formales, gramáticas y Jerarquía de Chomsky: 1. Regulares (Tipo-3): *a*) Expresiones Regulares 2. Libres de Contexto (Tipo-2) 3. Sensibles al Contexto (Tipo-1) 4. Recursivamente Enumerables (Tipo-0)
- Relaciones entre autómatas formales, lenguajes y gramáticas
- Autómatas deterministas y no deterministas
- Demostraciones del Lema de Bombeo: 1. Demostración de la limitación de Estado Finito/Lenguaje Regular 2. Limitación de Autómatas de Pila/Lenguaje Libre de Contexto
- Modelos equivalentes de computación algorítmica: 1. Máquinas de Turing y Variaciones (por ejemplo, multicinta, no deterministas) 2. Cálculo Lambda 3. Funciones Mu-Recursivas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Para cada autómata formal en esta unidad: *a*) Explicar su definición comparando sus características con los otros autómatas de esta unidad *b*) Usando un ejemplo, explicar paso a paso cómo el autómata opera sobre la entrada incluyendo si acepta la entrada asociada *c*) Explicar un ejemplo de entradas que pueden y no pueden ser aceptadas por el autómata. [Explicar]
2. Dado un problema, desarrollar un autómata apropiado que aborde el problema [Crear]
3. Desarrollar una expresión regular para un lenguaje regular dado expresado en lenguaje natural [Crear]
4. Explicar la diferencia entre expresiones regulares (aceptadores Tipo-3) y las expresiones regulares (aceptadores Tipo-2) usadas en lenguajes de programación [Explicar]
5. Para cada modelo formal en esta unidad: *a*) Explicar su definición comparando sus características con las otras en esta unidad *b*) Explicar ejemplos de entradas que son y no pueden ser aceptadas por el lenguaje/gramática. [Explicar]
6. Para cada autómata formal en esta unidad explicar (comparar/contrastar) sus capacidades deterministas y no deterministas [Explicar]
7. Aplicar lemas de bombeo, o medios alternativos, para demostrar las limitaciones de los autómatas de Estado Finito y Pila [Aplicar]
8. Convertir entre notaciones equivalentemente poderosas para un lenguaje, incluyendo entre DFAs, NFAs y expresiones regulares, y entre PDAs y CFGs [Aplicar]

3.2.10. AL/Computabilidad y Decidibilidad (Core Tier1: 6 hrs, Core Tier2: 14 hrs)

Temas:
Core

- Decidabilidad, (in)computabilidad y parada
- La tesis de Church-Turing
- Corrección algorítmica: 1. Invariantes (por ejemplo, en iteración, recursión, búsqueda en árbol)
- Decidabilidad: 1. Aritmetización y diagonalización
- Reducibilidad y reducciones
- Complejidad de tiempo basada en Máquina de Turing
- Complejidad de espacio (por ejemplo, Pspace, Teorema de Savitch)

Non Core

- Computación cuántica: 1. Postulados de la mecánica cuántica: *a*) Espacio de estados *b*) Evolución del estado *c*) Composición de estados *d*) Medición del estado 2. Representaciones de vector columna de qubits 3. Representaciones matriciales de operaciones cuánticas 4. Compuertas cuánticas simples (por ejemplo, XNOT, CNOT)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar una Máquina de Turing universal y su operación [Explicar]
2. Presentar a una audiencia de compañeros de trabajo y gerentes la imposibilidad de proporcionarles un programa que verifique todos los otros programas, incluyendo algunos aparentemente simples, en busca de bucles infinitos incluyendo una explicación del problema de la parada, por qué no tiene solución algorítmica y su importancia para la computación algorítmica del mundo real [Explicar]
3. Explicar ejemplos de problemas clásicos incomputables [Explicar]
4. Explicar la Tesis de Church-Turing y su importancia para la computación algorítmica [Explicar]
5. Explicar cómo los invariantes (de bucle) pueden usarse para demostrar la corrección de un algoritmo [Explicar]
6. Aplicar aritmetización y diagonalización para demostrar que el Problema de la Parada para Máquinas de Turing es Indecidible [Aplicar]
7. Dado un lenguaje indecidible conocido, aplicar una reducción por mapeo o historia computacional para demostrar que otro lenguaje es indecidible [Aplicar]
8. Explicar el teorema de Rice y su importancia [Explicar]

9. Explicar un ejemplo de demostración de un problema que es incomputable reduciendo un problema incomputable clásico conocido a él [Explicar]
10. Explicar las funciones Recursivas Primitivas y Generales (cero, sucesor, selección, recursión primitiva, composición y Mu), su importancia e implementaciones en Máquina de Turing [Explicar]
11. Explicar cómo se realiza la computación en el Cálculo Lambda (por ejemplo, conversión alfa y reducción beta) [Explicar]

NonCore:

12. Para un sistema cuántico dar ejemplos que expliquen los siguientes postulados: *a*) Espacio de Estados - estado del sistema representado como un vector unitario en espacio de Hilbert *b*) Evolución del Estado - el uso de operadores unitarios para evolucionar el estado del sistema *c*) Composición de Estados - el uso de producto tensorial para componer estados del sistema *d*) Medición del Estado - la salida probabilística de medir un estado del sistema. [Explicar]
13. Explicar la operación de una compuerta cuántica XNOT o CNOT sobre un bit cuántico representado como una matriz y vector columna, respectivamente [Explicar]

3.2.11. AL/Sociedad, ética y la Profesión

Temas:

Core

- Algoritmos sociales, éticos y seguros
- Equidad algorítmica
- Anonimato (por ejemplo, Privacidad Diferencial)
- Responsabilidad/Transparencia
- Algoritmos responsables
- Impactos económicos y otros de algoritmos ineficientes
- Sostenibilidad
- Computación consciente del contexto

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Desarrollar soluciones algorítmicas a problemas sociales del mundo real, como enrutar una ambulancia a un hospital [Crear]
2. Explicar el impacto que un algoritmo puede tener en el medio ambiente y la sociedad cuando se usa para resolver un problema del mundo real considerando su sostenibilidad y que puede afectar a diferentes grupos sociales de diferentes maneras [Explicar]
3. Preparar una presentación que justifique la selección de estructuras de datos y/o algoritmos apropiados para resolver un problema del mundo real dado [Explicar]
4. Explicar un ejemplo que articule cómo la privacidad diferencial protege el conocimiento de los datos de un individuo [Explicar]
5. Explicar los impactos ambientales de las decisiones de diseño que se relacionan con el diseño de algoritmos [Explicar]
6. Explicar los compromisos involucrados en los algoritmos de prueba de trabajo y prueba de participación [Explicar]

3.3. Arquitectura y Organización (AR)

Los profesionales de la informática pasan considerable tiempo escribiendo código eficiente para resolver un problema particular en un dominio de aplicación. A medida que ocurre el cambio del procesamiento secuencial al paralelo, se requiere una comprensión más profunda de las arquitecturas de computadoras subyacentes. La arquitectura ya no puede verse como una caja negra donde los principios de una arquitectura pueden aplicarse a otra. En cambio, los programadores deben mirar dentro de la caja negra y usar componentes específicos para mejorar el rendimiento del sistema y la eficiencia energética.

El área de conocimiento de Arquitectura y Organización (AR) tiene como objetivo desarrollar una comprensión más profunda de los entornos de hardware sobre los cuales se basa casi toda la computación, y las interfaces relevantes proporcionadas a las capas de software superiores. El hardware

objetivo comprende desde procesadores de sistemas embebidos de baja gama hasta multiprocesadores empresariales de alta gama.

Los temas en esta área de conocimiento beneficiarán a los estudiantes permitiéndoles apreciar los principios arquitectónicos fundamentales de los sistemas informáticos modernos, incluyendo el desafío de aprovechar el paralelismo para sostener las mejoras de rendimiento y energía en el futuro. Esta KA ayudará a los estudiantes de informática a alejarse del enfoque de caja negra y a ser más conscientes del sistema informático subyacente y las eficiencias que arquitecturas específicas pueden lograr.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.3.1 Lógica Digital y Sistemas Digitales (Pág. 58)		3
⭐ 3.3.2 Representación de Datos a Nivel de Máquina (Pág. 58)	1	
⭐ 3.3.3 Organización de Máquina a Nivel de Ensamblador (Pág. 59)	1	2
⭐ 3.3.4 Jerarquía de Memoria (Pág. 60)	6	
⭐ 3.3.5 Interconexión y Comunicación (Pág. 60)	1	
⭐ 3.3.6 Organización Funcional (Pág. 60)		2
⭐ 3.3.7 Rendimiento y Eficiencia Energética (Pág. 61)		3
⭐ 3.3.8 Arquitecturas Heterogéneas (Pág. 61)		2
⭐ 3.3.9 Arquitecturas de Procesador Seguras (Pág. 62)		2
⭐ 3.3.10 Arquitecturas Cuánticas (Pág. 62)		2
3.3.11 Problemas de Sostenibilidad (Pág. 63)	Electivo	

Cuadro 3.3: Lista de KUs del área de Arquitectura y Organización.

3.3.1. AR/Lógica Digital y Sistemas Digitales (Core Tier2: 3 hrs)

Temas:
Core

- Lógica combinacional vs secuencial/arreglos de puertas programables en campo (FPGAs): 1. Lógica combinacional fundamental 2. Bloque de construcción de lógica secuencial
- Herramientas de diseño asistido por computadora que procesan representaciones de hardware y arquitectura
- Síntesis de alto nivel: 1. Notación de transferencia de registros 2. Lenguaje de descripción de hardware (por ejemplo, Verilog/VHDL/Chisel)
- Flujo de diseño de sistema en chip (SoC)
- Restricciones físicas: 1. Retardos de puertas 2. Fan-in y fan-out 3. Energía/potencia 4. Velocidad de la luz

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir la progresión de los componentes de tecnología informática desde tubos de vacío hasta VLSI, desde arquitecturas de computadoras mainframe hasta la organización de computadoras a escala de almacén [Debatir]
2. Describir el paralelismo y las dependencias de datos entre y dentro de componentes en una arquitectura de computadora heterogénea moderna [Describir]
3. Explicar la relación entre paralelismo y consumo de energía [Explicar]
4. Construir el diseño de bloques básicos para una computadora: unidad aritmético-lógica (nivel de puertas), registros (nivel de puertas), unidad central de procesamiento (nivel de transferencia de registros) y memoria (nivel de transferencia de registros) [Crear]
5. Evaluar bloques básicos simples (por ejemplo, unidad aritmético-lógica, registros, movimiento entre registros) de un diseño de computadora simple [Evaluar]
6. Analizar el comportamiento de temporización de un procesador segmentado, identificando problemas de dependencia de datos [Analizar]

3.3.2. AR/Representación de Datos a Nivel de Máquina (Core Tier1: 1 hr)

Temas:
Core

- Visión general e historia de la arquitectura de computadoras
- Bits, bytes y palabras
- Representaciones sin signo, con signo y complemento a dos
- Representación de datos numéricos y bases numéricas: 1. Punto fijo 2. Punto flotante
- Representación de datos no numéricos
- Representación de registros, arreglos y tipos de datos UTF

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir por qué todo en las computadoras son datos, incluyendo instrucciones [Debatir]
2. Explicar cómo las representaciones numéricas de longitud fija pueden afectar la exactitud y precisión [Explicar]
3. Describir cómo se almacenan los enteros negativos en representaciones de magnitud con signo y complemento a dos [Describir]
4. Discutir cómo diferentes formatos pueden representar datos numéricos [Debatir]
5. Explicar la representación a nivel de bits de datos no numéricos, como caracteres, cadenas, registros y arreglos [Explicar]
6. Traducir datos numéricos de un formato a otro [Traducir]
7. Describir cómo un sumador único (sin detección de desbordamiento) puede manejar tanto entrada con signo (complemento a dos) como sin signo (binario) sin "saber" qué formato está usando una entrada dada [Describir]

3.3.3. AR/Organización de Máquina a Nivel de Ensamblador (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Arquitectura de máquina von Neumann
- Unidad de control: captación, decodificación y ejecución de instrucciones
- Introducción a SIMD vs MIMD y la taxonomía de Flynn
- Organización de multiprocesadores/multinúcleo de memoria compartida
- Arquitectura del conjunto de instrucciones (ISA) (por ejemplo, x86, ARM y RISC-V): 1. Conjuntos de instrucciones de ancho fijo vs variable 2. Formatos de instrucción 3. Manipulación de datos, control, E/S 4. Modos de direccionamiento 5. Programación en lenguaje de máquina 6. Programación en lenguaje ensamblador
- Mecanismos de llamada y retorno de subrutinas
- E/S e interrupciones
- Segmentos de montículo, estáticos, pila y código

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir cómo las unidades funcionales clásicas de von Neumann se implementan en sistemas embebidos, particularmente en memoria dentro y fuera del chip [Debatir]
2. Describir cómo se ejecutan las instrucciones en una máquina von Neumann clásica, con extensiones para hilos, sincronización de multiprocesadores y ejecución SIMD [Describir]
3. Evaluar un diagrama de ejemplo con paralelismo a nivel de instrucción y riesgos para describir cómo se manejan en las canalizaciones de procesadores típicos [Evaluar (valorar)]
4. Discutir cómo se representan las instrucciones a nivel de máquina y en el contexto de un ensamblador simbólico [Debatir]
5. Mapear un ejemplo de patrones de lenguaje de alto nivel a notaciones de lenguaje ensamblador/máquina [Mapear/Mapa conceptual]
6. Contrastar diferentes formatos de instrucción considerando aspectos como direcciones por instrucción y formatos de longitud variable vs longitud fija [Contrastar]
7. Analizar un diagrama de subrutina para comentar cómo se manejan las llamadas de subrutina a nivel de ensamblador [Analizar]
8. Describir conceptos básicos de interrupciones y operaciones de E/S [Describir]
9. Escribir un programa simple en lenguaje ensamblador para procesamiento y manipulación de cadenas/arreglos [Escribir]

3.3.4. AR/Jerarquía de Memoria (Core Tier1: 6 hrs)

Temas:
Core

- Jerarquía de memoria: la importancia de la localidad temporal y espacial
- Organización y operaciones de memoria principal
- Memoria persistente (por ejemplo, SSD, discos estándar)
- Latencia, tiempo de ciclo, ancho de banda e intercalado
- Memorias caché: 1. Mapeo de direcciones 2. Tamaño de bloque 3. Política de reemplazo y almacenamiento 4. Prebúsqueda
- Coherencia de caché multiprocesador
- Memoria virtual (soporte de hardware)
- Manejo de fallas y confiabilidad
- Confiabilidad: 1. Codificación de error 2. Compresión de datos 3. Integridad de datos
- Procesamiento en Memoria (PIM)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Usando un diagrama del sistema de memoria, identificar los principales tipos de tecnología de memoria (por ejemplo, SRAM, DRAM) y su costo y rendimiento relativos [Analizar]
2. Medir el efecto de la latencia de memoria en el tiempo de ejecución [Evaluar]
3. Enumerar las funciones de un sistema con gestión de memoria virtual [Enumerar]
4. Calcular el tiempo promedio de acceso a memoria bajo varias configuraciones de caché y memoria y mezclas de referencias de instrucciones y datos [Computar/Calcular]

3.3.5. AR/Interconexión y Comunicación (Core Tier1: 1 hr)

Temas:
Core

- Fundamentos de E/S: 1. Protocolo de enlace y almacenamiento en búfer 2. E/S programada 3. E/S basada en interrupciones
- Estructuras de interrupción: vectorizadas y priorizadas, acuse de recibo de interrupción
- Dispositivos de E/S (por ejemplo, ratón, teclado, pantalla, cámara, sensores, actuadores)
- Almacenamiento externo, organización física y unidades
- Fundamentos de buses: 1. Protocolos de bus 2. Arbitraje 3. Acceso directo a memoria (DMA)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar un diagrama de control de interrupciones para comentar cómo se usan las interrupciones para implementar control de E/S y transferencias de datos [Analizar]
2. Enumerar varios tipos de buses en un sistema informático [Enumerar]
3. Enumerar las ventajas de los discos magnéticos y contrastarlas con las de los discos de estado sólido [Listar/Enumerar]

3.3.6. AR/Organización Funcional (Core Tier2: 2 hrs)

Temas:
Core

- Implementación de rutas de datos simples, incluyendo segmentación de instrucciones, detección de riesgos y resolución (por ejemplo, paradas, reenvío)
- Unidad de control: 1. Implementación cableada 2. Realización microprogramada
- Segmentación de instrucciones
- Introducción al paralelismo a nivel de instrucción (ILP)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar implementaciones alternativas de rutas de datos en arquitecturas de computadoras modernas [Comparar]
2. Producir un conjunto de señales de control para sumar dos enteros usando implementaciones cableadas y microprogramadas [Producir]

3. Discutir el paralelismo a nivel de instrucción usando segmentación y los riesgos significativos que pueden ocurrir [Debatir]
4. Diseñar un procesador completo, incluyendo ruta de datos y control [Diseñar]
5. Calcular los ciclos promedio por instrucción para una implementación dada de procesador y sistema de memoria [Computar/Calcular]

3.3.7. AR/Rendimiento y Eficiencia Energética (Core Tier2: 3 hrs)

Temas:
Core

- Evaluación de rendimiento-energía (introducción): rendimiento, consumo de energía, costos de memoria y comunicación
- Predicción de bifurcaciones, ejecución especulativa, ejecución fuera de orden, algoritmo de Tomasulo
- Mejoras para procesadores vectoriales y GPUs
- Soporte de hardware para multihilos: 1. Condiciones de carrera 2. Implementaciones de bloqueo 3. Sincronización punto a punto 4. Implementación de barreras
- Escalabilidad
- Arquitecturas alternativas incluyendo VLIW/EPIC, aceleradores y otros procesadores de propósito especial
- Escalado dinámico de voltaje y frecuencia (DVFS)
- Silicio Oscuro

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir métricas de evaluación de rendimiento y eficiencia energética [Debatir]
2. Analizar un diagrama de ejecución especulativa y escribir sobre las decisiones que se pueden tomar [Analizar]
3. Crear un diagrama de evaluación comparativa de rendimiento-vatio de GPU [Crear]
4. Escribir un programa multihilo que sume (en paralelo) elementos de dos vectores enteros [Escribir]
5. Recomendar un conjunto de decisiones de diseño para arquitecturas de computadoras alternativas [Recomendar]
6. Enumerar conceptos clave asociados con el escalado dinámico de voltaje y frecuencia [Enumerar]
7. Medir la mejora en ahorro de energía para una cuantización de entero de 8 bits comparada con una cuantización de 32 bits [Evaluar]

3.3.8. AR/Arquitecturas Heterogéneas (Core Tier2: 2 hrs)

Temas:
Core

- Arquitecturas SIMD y MIMD (por ejemplo, GPUs de Propósito General, TPUs y NPUs)
- Sistemas de memoria heterogéneos: 1. Memoria compartida versus memoria distribuida 2. Memoria volátil vs no volátil 3. Protocolos de coherencia
- Arquitecturas Específicas de Dominio (DSAs): 1. Acelerador de Aprendizaje Automático 2. Computación en red 3. Sistemas embebidos para aplicaciones emergentes 4. Computación neuromórfica 5. Dispositivos de computación perimetral
- Soluciones de empaquetado e integración como 3DIC y chiplets
- Aprendizaje automático en diseño de arquitectura: 1. Algoritmos de IA para análisis de carga de trabajo 2. Optimización de configuraciones de arquitectura para rendimiento y eficiencia energética

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar un diagrama de sistema con arquitecturas paralelas alternativas, por ejemplo, SIMD y MIMD, e identificar las diferencias clave [Analizar]
2. Discutir qué problemas de gestión de memoria se encuentran en multiprocesadores que no están presentes en uniprocesadores y cómo estos problemas podrían resolverse [Debatir]

3. Indicar las diferencias entre placa base de memoria, interconexión de memoria del procesador y memoria remota a través de redes, sus implicaciones para la latencia de acceso y su impacto en el rendimiento del programa [Diferenciar]
4. Discutir cómo determinarías cuándo usar un acelerador específico de dominio en lugar de una CPU de propósito general [Debatir]
5. Enumerar diferencias clave en los principios de diseño arquitectónico entre una unidad de procesamiento basada en vector y basada en escalar [Enumerar]
6. Enumerar las ventajas y desventajas de una arquitectura PIM [Listar/Enumerar]

3.3.9. AR/Arquitecturas de Procesador Seguras (Core Tier2: 2 hrs)

Temas:
Core

- Principios de Hardware Seguro: 1. Análisis de Riesgos de Seguridad, Protección de Activos y Modelo de Amenazas 2. Aceleración Criptográfica con Hardware 3. Soporte para virtualización (por ejemplo, aislamiento de SO)
- Raíces de confianza en hardware, Funciones Físicamente No Clonables (PUF)
- Generadores de Números Aleatorios por Hardware
- Extensiones de protección de memoria: 1. Verificación de límites de puntero en tiempo de ejecución (por ejemplo, desbordamiento de búfer) 2. Protección a nivel microarquitectónico 3. Protección a nivel de ISA
- Entorno de Ejecución Confiable (TEE): 1. Protecciones de Base de Computadora Confiable 2. Protección de máquinas virtuales 3. Protección de contenedores 4. Módulos de software confiables (Enclaves)
- Cifrado homomórfico para procesamiento de datos que preserva la privacidad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir principios de hardware seguro, explorando un marco para análisis de riesgos y protección de activos [Debatir]
2. Resumir cómo las Funciones Físicamente No Clonables (PUF) pueden ser un identificador único de dispositivo en aplicaciones de seguridad [Resumir]
3. Distinguir un generador de números aleatorios con soporte de hardware dedicado de generadores sin hardware dedicado a generar entropía [Distinguir]
4. Enumerar las ventajas y desventajas de la protección de memoria a nivel de ISA [Listar/Enumerar]
5. Describir problemas de diseño clave de un entorno de ejecución confiable (TEE) para soportar máquinas virtuales [Describir]

3.3.10. AR/Arquitecturas Cuánticas (Core Tier2: 2 hrs)

Temas:
Core

- Principios: 1. El principio de dualidad onda-partícula 2. El principio de incertidumbre en el experimento de doble rendija 3. ¿Qué es un Qubit? Superposición, interferencia y medición. Fotones como qubits 4. Sistemas de dos qubits, Entrelazamiento, estados de Bell, El teorema de No Señalización
- Axiomas de la MC: principio de superposición, axioma de medición, evolución unitaria
- Compuertas de un solo qubit para el modelo de circuito de computación cuántica: X, Z, H
- Compuertas de dos qubits y productos tensoriales, trabajando con matrices
- El Teorema de No Clonación. El protocolo de Teleportación Cuántica
- Algoritmos: 1. Algoritmos cuánticos simples: Bernstein-Vazirani, algoritmo de Simon 2. Implementar Deutsch-Josza con Interferómetros Mach-Zehnder 3. Factorización cuántica (Algoritmo de Shor) 4. Búsqueda cuántica (Algoritmo de Grover)
- Aspectos de implementación: 1. La implementación física de qubits 2. Control clásico de una Unidad de Procesamiento Cuántico (QPU) 3. Mitigación y control de errores, NISQ y más allá 4. Enfoques de medición

- Aplicaciones Emergentes: 1. Cifrado post-cuántico 2. La Internet Cuántica 3. Computación cuántica adiabática (AQC) y recocido cuántico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir cómo un objeto cuántico producido como partícula se propaga como una onda y se detecta como partícula con una distribución de probabilidad correspondiente a la onda [Debatir]
2. Discutir la naturaleza a nivel cuántico que es inherentemente probabilística [Debatir]
3. Expresar tu punto de vista sobre el entrelazamiento que puede usarse para crear correlaciones no clásicas, pero no hay forma de usar el entrelazamiento cuántico para enviar mensajes más rápido que la velocidad de la luz [Evaluar]
4. Describir el paralelismo cuántico y el papel de la interferencia constructiva vs destructiva en algoritmos cuánticos dada la naturaleza probabilística de las mediciones [Describir]
5. Analizar un fragmento de código proporcionando el papel de la transformada cuántica de Fourier (QFT) en el algoritmo de Shor [Analizar]
6. Escribir un programa para implementar el algoritmo de Shor en un simulador, destacando los componentes clásicos y aspectos del algoritmo de Shor [Escribir]
7. Enumerar los detalles de cada modalidad de qubit (por ejemplo, ion atrapado, superconductor, espín de silicio, fotónico, punto cuántico, átomo neutro, topológico, centro de color, electrón sobre helio) [Enumerar]
8. Contrastar AQC con el modelo de compuertas de computación cuántica y los problemas para los que cada uno es más adecuado resolver [Contrastar]

3.3.11. AR/Problemas de Sostenibilidad

Temas:
Non Core

- Impactos ambientales de decisiones de implementación: 1. Objetivos de sostenibilidad, consumo de recursos y viabilidad económica 2. Huella de carbono, desechos electrónicos de hardware 3. La huella energética de los centros de datos en varias cargas de trabajo (por ejemplo, entrenamiento y uso de modelos de IA) 4. Directrices para estándares de diseño sostenible

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Evaluar los impactos ambientales del despliegue de un proyecto dado (por ejemplo, el consumo de energía de CPUs y GPUs, contribución a desechos electrónicos, y efecto de la virtualización de hardware en centros de datos) [Evaluar (valorar)]

3.4. Gestión de Datos (DM)

Desde mediados de la década de 1970, el estudio de la Gestión de Datos (DM) ha significado un estudio casi exclusivo de los sistemas de bases de datos relacionales. Dependiendo del contexto institucional, los estudiantes han estudiado, en proporciones variables, lo siguiente.

- Modelado de datos y diseño de bases de datos: por ejemplo, modelo de datos E-R, modelo relacional, teoría de normalización
- Construcción de consultas: por ejemplo, álgebra relacional, SQL
- Procesamiento de consultas: por ejemplo, índices (árbol B+, hash), algoritmos (por ejemplo, ordenamiento externo, selección, proyección, unión), optimización de consultas (transformaciones, selección de índices)
- Internos de DBMS: por ejemplo, concurrencia/bloqueos, gestión de transacciones, gestión de búfer

Se espera que los graduados de hoy posean habilidades de usuario (en lugar de implementador) de DBMS. Estas incluyen principalmente modelado de datos y construcción de consultas; capacidad para tomar una colección no organizada de datos, organizarla usando un DBMS y acceder/actualizar la colección mediante consultas.

Además, los estudiantes necesitan estudiar lo siguiente:

- El rol que juegan los datos en una organización. Esto incluye el Ciclo de Vida de los Datos: Creación-Procesamiento-Revisión/Reporte-Retención/Recuperación-Destrucción.
- Los aspectos sociales/legales de la recolección de datos: por ejemplo, escala, privacidad de datos, privacidad de base de datos (cumplimiento) por diseño, desidentificación, propiedad, confiabilidad, seguridad de base de datos y aplicaciones intencionales y no intencionales.
- Tecnologías emergentes y avanzadas que están aumentando/reemplazando los sistemas relacionales tradicionales, particularmente aquellas usadas para soportar análisis de datos (grandes), incluyendo NoSQL (por ejemplo, JSON, XML, bases de datos de almacén clave-valor), bases de datos en la nube, MapReduce y dataframes.
- Los roles existentes y emergentes para aquellos involucrados con la gestión de datos, que incluyen los siguientes.
  - Profesionales de características de producto: aquellos que usan tanto bases de datos operativas SQL como NoSQL.
  - Profesionales analíticos/Profesionales en datos: aquellos que escriben código SQL, Python y Scala analítico para construir activos de datos para grupos de negocios.
  - Analistas de negocio: aquellos que construyen/gestionan datos más frecuentemente con hojas de cálculo Excel.
  - Profesionales de infraestructura de datos: aquellos que implementan un sistema de gestión de datos en una variedad de aplicaciones de datos (por ejemplo, OLTP).
  - **Todos** los que producen o consumen datos deben comprender los problemas sociales, éticos y profesionales asociados.

Un rol que trasciende todas las categorías anteriores es el de custodio de datos. Anteriormente, los datos eran vistos como un recurso para ser gestionado (Gestión de Sistemas de Información) al igual que otros recursos empresariales. Hoy, los datos son vistos en un contexto más amplio. Los datos sobre clientes ahora pueden verse como pertenecientes a (o en algunos contextos nacionales, como propiedad de) esos clientes. Ahora existe una comprensión aceptada de que el almacenamiento y uso seguro y ético de los datos institucionales es parte de ser un custodio de datos responsable.

Además, reconocemos la tensión entre un enfoque curricular en preparación profesional versus el estudio de un área de conocimiento como un esfuerzo científico. Esto es particularmente cierto con la Gestión de Datos. Por ejemplo, demostrar (o al menos conocer) la completitud de los Axiomas de Armstrong es fundamental en la teoría de dependencias funcionales. Sin embargo, la mayoría de los graduados en informática nunca utilizarán este concepto durante sus carreras profesionales. Lo mismo puede decirse de muchos otros temas en el canon de Gestión de Datos. Por el contrario, si nuestros graduados solo pueden normalizar datos en forma normal de Boyce-Codd (usando una herramienta automatizada) y escribir consultas SQL, sin comprender el rol que juegan los índices en la ejecución eficiente de consultas, les hemos hecho un perjuicio a ellos y a la sociedad.

Con este fin, el número de horas de Núcleo CS es relativamente pequeño en relación con las horas de Núcleo KA. Este enfoque está diseñado para permitir que las instituciones con contextos diferentes personalicen sus planes de estudio apropiadamente. Una institución que se enfoque en implementación OLTP, por ejemplo, priorizaría el almacenamiento eficiente y el acceso a datos, mientras que una institución que se enfoque en características de producto priorizaría el acceso programático a bases de datos existentes.

Sin embargo, como una institución maneje esta tensión, deseamos dar voz a una de las ironías de los planes de estudio de informática. Los estudiantes típicamente pasan gran parte de su vida educativa leyendo (y escribiendo) datos de un archivo o interactivamente, mientras que fuera de la academia los datos predominantes provienen de bases de datos a las que se accede programáticamente. Quizás en un futuro no muy lejano los estudiantes aprenderán acceso programático a bases de datos desde temprano y luego continuarán esta práctica a medida que avanzan en su plan de estudios.

Finalmente, entendemos que aunque el KA de Gestión de Datos puede ser ortogonal a los KAs de SEC (Seguridad) y SEP (Sociedad, ética y la Profesión), también es punto cero para estas (y otras) áreas de conocimiento. Al diseñar almacenes de datos persistentes, la pregunta de qué debe almacenarse debe examinarse desde perspectivas legales y éticas. ¿Hay preocupaciones de privacidad? Y tan importante como eso, ¿qué tan bien protegidos están los datos?

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.4.1 El Rol de los Datos y el Ciclo de Vida de los Datos (Pág. 65)	2	
⭐ 3.4.2 Conceptos Fundamentales de Sistemas de Bases de Datos (Pág. 65)	2	1
⭐ 3.4.3 Modelado de Datos (Pág. 66)	2	3
⭐ 3.4.4 Bases de Datos Relacionales (Pág. 66)	1	3
⭐ 3.4.5 Construcción de Consultas (Pág. 67)	2	4
⭐ 3.4.6 Procesamiento de Consultas (Pág. 67)		4
⭐ 3.4.7 Internos de DBMS (Pág. 68)		4
⭐ 3.4.8 Sistemas NoSQL (Pág. 68)		2
⭐ 3.4.9 Seguridad y Privacidad de Datos (Pág. 69)	1	2
⭐ 3.4.10 Análisis de Datos (Pág. 69)		3
3.4.11 Bases de Datos Distribuidas/Computación en la Nube (Pág. 70)	Electivo	
3.4.12 Bases de Datos Semiestructuradas y No Estructuradas (Pág. 70)	Electivo	
3.4.13 Sociedad, ética y Profesión (Pág. 70)	Electivo	

Cuadro 3.4: Lista de KUs del área de Gestión de Datos.

3.4.1. DM/El Rol de los Datos y el Ciclo de Vida de los Datos (Core Tier1: 2 hrs)

Temas:
Core

- Ciclo de Vida de los Datos: Creación-Procesamiento-Revisión/Reporte-Retención/Recuperación-Destrucción

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar las cinco etapas del Ciclo de Vida de los Datos [Analizar]

3.4.2. DM/Conceptos Fundamentales de Sistemas de Bases de Datos (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Propósito y ventajas de los sistemas de bases de datos
- Componentes de los sistemas de bases de datos
- Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso)
- Arquitectura de base de datos, independencia de datos y abstracción de datos
- Gestión de transacciones
- Normalización
- Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce)
- Cómo soportar aplicaciones solo de CRUD
- Bases de datos distribuidas/sistemas basados en la nube
- Datos estructurados, semiestructurados y no estructurados
- Uso de un lenguaje de consulta declarativo
- Sistemas que soportan contenido estructurado y/o de flujo

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Analizar]
2. Enumerar los componentes de un sistema de base de datos (relacional) [Enumerar]
3. Seguir una consulta a medida que es procesada por los componentes de un sistema de base de datos (relacional) [Analizar]
4. Defender el valor de la independencia de datos [Defender]
5. Componer una consulta simple de selección-proyección-unión en SQL [Componer]

6. Enumerar las cuatro propiedades de un gestor de transacciones correcto [Enumerar]
7. Describir las ventajas de eliminar datos duplicados repetidos [Describir]
8. Esbozar cómo MapReduce usa paralelismo para procesar datos eficientemente [Esquematizar/Esbozar]
9. Evaluar las diferencias entre bases de datos estructuradas y semiestructuradas/no estructuradas [Evaluar]

3.4.3. DM/Modelado de Datos (Core Tier1: 2 hrs, Core Tier2: 3 hrs)

Temas:
Core
- Modelado de datos
- Modelo de datos relacional
- Modelos conceptuales (por ejemplo, entidad-relación, diagramas UML)
- Modelos de datos semiestructurados (expresados usando DTD, XML o JSON Schema, por ejemplo)

Non Core
- Modelos de hojas de cálculo
- Modelos orientados a objetos: 1. GraphQL
- Nuevas características en SQL
- Temas de Modelado de Datos Especializados: 1. Datos de series temporales (agregación, unión) 2. Datos de grafos (recorrido de enlaces) 3. Técnicas para evitar acceso ineficiente a datos crudos (por ejemplo, "precio promedio diario"): vistas materializadas y estructuras de datos especiales (por ejemplo, Hyperloglog, mapa de bits) 4. Datos Geoespaciales (por ejemplo, bases de datos GIS)

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Describir los componentes del modelo de datos relacional [Describir]
2. Modelar relaciones 1:1, 1:n y n:m usando el modelo de datos relacional [Usar]
3. Describir los componentes del modelo E-R (o algún otro modelo no relacional) [Describir]
4. Modelar un entorno dado usando un modelo de datos conceptual [Analizar]
5. Modelar un entorno dado usando el modelo de datos basado en documentos o basado en almacén clave-valor [Aplicar]

3.4.4. DM/Bases de Datos Relacionales (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core
- Integridad de entidad y referencial: Clave candidata, superclaves
- Diseño de base de datos relacional
- Mapeo de esquema conceptual a un esquema relacional
- Diseño físico de base de datos: estructuras de archivos y almacenamiento
- Introducción a la teoría de dependencias funcionales
- Teoría de Normalización: 1. Descomposición de un esquema; propiedades de unión sin pérdida y preservación de dependencias de una descomposición 2. Formas normales (BCNF) 3. Desnormalización (para eficiencia)

Non Core
- Teoría de dependencias funcionales: 1. Clausura de un conjunto de atributos 2. Cubierta canónica
- Teoría de normalización: 1. Dependencia multivaluada (4NF) 2. Dependencia de unión (PJNF, 5NF) 3. Teoría de representación

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Describir las características definitorias detrás del modelo de datos relacional [Describir]
2. Comentar sobre la diferencia entre una clave foránea y una superclave [Explicar]
3. Enumerar los diferentes tipos de restricciones de integridad [Enumerar]

4. Componer un esquema relacional a partir de un esquema conceptual que contenga relaciones 1:1, 1:n y n:m [Componer]
5. Mapear estructura de archivo apropiada a relaciones e índices [Mapear/Mapa conceptual]
6. Describir cómo la teoría de dependencias funcionales generaliza la noción de clave [Describir]
7. Defender una descomposición dada como sin pérdida y/o que preserva dependencias [Defender]
8. Detectar qué forma normal produce una descomposición dada [Detectar]
9. Comentar sobre las razones para desnormalizar una relación [Interpretar]

3.4.5. DM/Construcción de Consultas (Core Tier1: 2 hrs, Core Tier2: 4 hrs)

Temas:
Core

- Formación de Consultas SQL: 1. Ejecución interactiva de SQL 2. Ejecución programática de una consulta SQL
- álgebra Relacional
- SQL: 1. Definición de datos incluyendo especificaciones de integridad y otras restricciones 2. Sublenguaje de actualización

Non Core

- Cálculo Relacional
- QBE y entornos de 4ta generación
- Diferentes formas de invocar consultas no procedimentales en lenguajes convencionales
- Introducción a otros lenguajes de consulta principales (por ejemplo, XPATH, SPARQL)
- Procedimientos almacenados

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Componer consultas SQL que incorporen selección, proyección, unión, intersección, diferencia de conjuntos y división de conjuntos [Componer]
2. Determinar cuándo una consulta SQL anidada está correlacionada o no [Determinar]
3. Iterar sobre datos recuperados programáticamente de una base de datos a través de una consulta SQL [Crear]
4. Definir, en SQL, un esquema de relación, incluyendo todas las restricciones de integridad y activadores de eliminación/actualización [Definir]
5. Componer una consulta SQL para actualizar una tupla en una relación [Componer]

3.4.6. DM/Procesamiento de Consultas (Core Tier2: 4 hrs)

Temas:
Core

- Estructuras de página
- Estructuras de índice: 1. árboles B+ 2. Índices hash: estáticos y dinámicos 3. Creación de índices en SQL
- Estructuras de archivos: 1. Archivos de montón (*heap*) 2. Archivos hash
- Algoritmos para operadores de consulta: 1. Ordenamiento externo 2. Selección 3. Proyección; con y sin eliminación de duplicados 4. Uniones naturales: Bucle anidado, Mezcla-ordenada, Unión hash 5. Análisis de eficiencia de algoritmos
- Transformaciones de consultas
- Optimización de consultas: 1. Rutas de acceso 2. Construcción de plan de consulta 3. Estimación de selectividad 4. Planes solo de índice
- Procesamiento de Consultas en Paralelo (por ejemplo, exploración paralela, unión paralela, agregación paralela)
- Afinamiento/rendimiento de base de datos: 1. Selección de índices 2. Impacto de índices en el rendimiento de consultas 3. Desnormalización

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir el propósito y organización de las estructuras de índice tanto de árbol B+ como hash [Describir]

2. Componer un comando SQL para crear un índice (de cualquier tipo) [Componer]
3. Especificar los pasos para los diversos algoritmos de operadores de consulta: ordenamiento externo, proyección con eliminación de duplicados, unión por mezcla-ordenada, unión hash, unión por bucle anidado por bloques [Analizar]
4. Derivar el tiempo de ejecución (en solicitudes de E/S) para cada uno de los algoritmos anteriores [Analizar]
5. Transformar una consulta en álgebra relacional a su equivalente apropiado para una ejecución canalizada de profundidad izquierda [Rediseñar]
6. Calcular estimaciones de selectividad para una operación de selección y/o unión dada [Computar/Calcular]
7. Describir cómo modificar una estructura de índice para facilitar una operación solo de índice para una relación dada [Describir]
8. Para un escenario dado decidir qué índices soportar para la ejecución eficiente de un conjunto de consultas [Evaluar (valorar)]
9. Describir cómo los DBMS aprovechan el paralelismo para acelerar el procesamiento de consultas dividiendo el trabajo entre múltiples procesadores o nodos [Describir]

3.4.7. DM/Internos de DBMS (Core Tier2: 4 hrs)

Temas:
Core

- Gestión de Búfer de DB
- Gestión de Transacciones: 1. Niveles de Aislamiento 2. ACID 3. Serializabilidad 4. Transacciones Distribuidas
- Control de Concurrencia: 1. Bloqueo en 2 Fases 2. Estrategias de manejo de bloqueos mutuos 3. Modelos de consistencia basados en quórum
- Gestor de Recuperación: 1. Relación con el Gestor de Búfer

Non Core

- Control de Concurrencia: 1. Control de concurrencia optimista 2. Control de concurrencia por marca de tiempo
- Gestor de Recuperación: 1. Registro por escritura anticipada 2. Sistema de recuperación ARIES (Análisis, REDO, UNDO)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir cómo un DBMS gestiona su Grupo de Búferes [Describir]
2. Describir las cuatro propiedades para un gestor de transacciones correcto [Describir]
3. Esbozar el principio de serializabilidad [Esquematizar/Esbozar]

3.4.8. DM/Sistemas NoSQL (Core Tier2: 2 hrs)

Temas:
Core

- ¿Por qué NoSQL? (por ejemplo, Desajuste de impedancia entre Aplicación [CRUD] y RDBMS)
- Modelo de datos Clave-Valor y Documento

Non Core

- Sistemas de almacenamiento (por ejemplo, sistemas Clave-Valor, Lagos de Datos (*Data Lakes*))
- Modelos de Distribución (Fragmentación y Replicación)
- Bases de Datos de Grafos
- Modelos de Consistencia (Actualización y Lectura, consistencia de quórum, teorema CAP)
- Modelo de procesamiento (por ejemplo, Map-Reduce, map-reduce multi-etapa, map-reduce incremental)
- Estudios de Caso: Sistema de almacenamiento en la nube (por ejemplo, S3); Bases de datos de grafos; "Cuándo no usar NoSQL"

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Desarrollar un caso de uso para el uso de NoSQL sobre RDBMS [Crear]
2. Describir las características definitorias detrás de los modelos de datos basados en Clave-Valor y Documentos [Describir]

3.4.9. DM/Seguridad y Privacidad de Datos (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Diferencias entre seguridad de datos y privacidad de datos
- Proteger datos y sistemas de bases de datos de ataques, incluyendo ataques de inyección como inyección SQL
- Información de identificación personal (PII) y su protección
- Consideraciones éticas para asegurar la seguridad y privacidad de los datos
- Necesidad de, y diferentes enfoques para asegurar datos en reposo, en tránsito y durante el procesamiento
- Auditoría de base de datos y su rol en forensía digital
- Inferencia de datos y prevención de ataques
- Leyes y regulaciones que gobiernan la seguridad de datos y la privacidad de datos

Non Core

- Factores de riesgo típicos y medidas de prevención para asegurar la integridad de datos
- Ransomware y prevención de pérdida y destrucción de datos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las diferencias en los objetivos para seguridad de datos y privacidad de datos [Describir]
2. Identificar y mitigar riesgos asociados con diferentes enfoques para proteger datos [Analizar]
3. Describir consideraciones legales y éticas de seguridad y privacidad de datos de extremo a extremo [Describir]
4. Desarrollar un sistema de auditoría de base de datos dadas consideraciones de riesgo [Crear]
5. Aplicar varios enfoques de exploración de datos para comprender conjuntos de datos no familiares [Aplicar]

3.4.10. DM/Análisis de Datos (Core Tier2: 3 hrs)

Temas:
Core

- Técnicas exploratorias de datos (motivación, representación, estadísticas descriptivas, visualizaciones)
- Ciclo de vida de ciencia de datos: comprensión del negocio, comprensión de datos, preparación de datos, modelado, evaluación, despliegue y aceptación del usuario
- Algoritmos de minería de datos y aprendizaje automático: por ejemplo, clasificación, agrupamiento, asociación, regresión
- Adquisición y gobernanza de datos
- Consideraciones de seguridad y privacidad de datos
- Equidad y sesgo de datos
- Técnicas de visualización de datos y su uso en análisis de datos
- Resolución de Entidades

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir varios enfoques de exploración de datos, incluyendo visualización, para comprender conjuntos de datos no familiares [Describir]
2. Aplicar varios enfoques de exploración de datos para comprender conjuntos de datos no familiares [Aplicar]
3. Describir algoritmos básicos de aprendizaje automático/minería de datos y cuándo son apropiados para su uso [Describir]

4. Aplicar varios algoritmos de aprendizaje automático/minería de datos [Aplicar]
5. Describir consideraciones legales y éticas en la adquisición, uso y modificación de conjuntos de datos [Describir]
6. Describir problemas de equidad y sesgo en la recolección y uso de datos [Describir]

3.4.11. DM/Bases de Datos Distribuidas/Computación en la Nube

Temas:
Non Core

- DBMS Distribuido: 1. Almacenamiento de datos distribuido 2. Procesamiento de consultas distribuido 3. Modelo de transacción distribuida 4. Soluciones homogéneas y heterogéneas 5. Bases de datos distribuidas cliente-servidor
- DBMS Paralelo: 1. Arquitecturas de DBMS paralelo: memoria compartida, disco compartido, nada compartido 2. Aceleración y escalamiento, por ejemplo, uso del modelo de procesamiento MapReduce 3. Replicación de datos y modelos de consistencia débil

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir los componentes clave de un DBMS distribuido, incluyendo almacenamiento de datos distribuido, procesamiento de consultas y gestión de transacciones [Describir]
2. Analizar las ventajas y desventajas entre arquitecturas de DBMS paralelo: memoria compartida, disco compartido y nada compartido [Analizar]
3. Describir estrategias de replicación de datos y modelos de consistencia débil en sistemas de bases de datos distribuidas [Describir]

3.4.12. DM/Bases de Datos Semiestructuradas y No Estructuradas

Temas:
Non Core

- Datos no estructurados vectorizados (texto, video, audio, etc.) y almacenamiento vectorial: 1. Vectorizador TF-IDF con n-grama 2. Word2Vec 3. Base de datos de arreglos o manejo de tipo de datos arreglo
- Bases de datos semiestructuradas (por ejemplo, JSON): 1. Almacenamiento: *a*) Codificación y compresión de tipos de datos anidados 2. Indexación: *a*) árbol B, índice de salto, filtro Bloom *b*) Índice invertido y compresión de mapa de bits *c*) Indexación por curva de llenado de espacio para datos geo-semiestructurados 3. Procesamiento de consultas para casos de uso OLTP y OLAP: *a*) Compromisos de inserción, selección, actualización/eliminación *b*) Estudios de caso sobre Postgres/JSON, MongoDB y Snowflake/JSON

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir representaciones vectorizadas de datos no estructurados (por ejemplo, TF-IDF, Word2Vec) y sus modelos de almacenamiento [Describir]
2. Aplicar técnicas de indexación (por ejemplo, árbol B, índice invertido, filtro Bloom) a datos semiestructurados almacenados en formato JSON [Aplicar]
3. Analizar las ventajas y desventajas del procesamiento de consultas en bases de datos semiestructuradas para casos de uso OLTP y OLAP [Analizar]

3.4.13. DM/Sociedad, ética y Profesión

Temas:
Core

- Problemas relacionados con la escala
- Privacidad de datos en general: 1. Privacidad por diseño y por defecto
- Anonimato de datos
- Propiedad/custodia de datos
- Aplicaciones intencionales y no intencionales de los datos almacenados
- Fiabilidad de los datos

- Procedencia, linaje de datos y gestión de metadatos
- Seguridad de datos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar tres problemas sociales y tres legales relacionados con grandes colecciones de datos [Enumerar]
2. Describir el valor de la privacidad de datos [Describir]
3. Identificar las partes interesadas en competencia con respecto a la propiedad de los datos [Analizar]
4. Enumerar tres consecuencias negativas no intencionales de una aplicación centrada en datos (bien conocida) dada (por ejemplo, Facebook, LastPass, Ashley Madison) [Enumerar]
5. Describir el significado de procedencia y linaje de datos [Describir]
6. Identificar cómo una base de datos podría contribuir a la seguridad de los datos, así como cómo puede introducir inseguridades [Analizar]

3.5. Fundamentos de los Lenguajes de Programación (FPL)

Los fundamentos de los lenguajes de programación están enraizados en las matemáticas discretas, la lógica y los lenguajes formales, y proporcionan una base para la comprensión de los complejos lenguajes de programación modernos. Aunque los lenguajes de programación varían según el paradigma del lenguaje y el dominio del problema, y evolucionan en respuesta tanto a las necesidades sociales como al avance tecnológico, comparten un modelo abstracto subyacente de computación y desarrollo de programas. Esto sigue siendo cierto incluso a medida que el hardware del procesador y su interfaz con las herramientas de programación se entrelazan y complejizan progresivamente. Una comprensión de las abstracciones y paradigmas de programación comunes permite un aprendizaje más rápido de los lenguajes de programación.

El área de conocimiento de Fundamentos de los Lenguajes de Programación se ocupa de articular los conceptos y principios subyacentes de los lenguajes de programación, la especificación formal de un lenguaje de programación y el comportamiento de un programa, explicar cómo se implementan los lenguajes de programación, comparar las fortalezas y debilidades de varios paradigmas de programación, y describir cómo los lenguajes de programación interactúan con entidades como sistemas operativos y hardware. Los conceptos aquí cubiertos son aplicables a varios lenguajes y comprender estos principios ayuda al aprendiz a pasar fácilmente de un lenguaje a otro, así como a seleccionar un paradigma y lenguaje de programación que mejor se adapte al problema en cuestión.

Los lenguajes de programación son el medio a través del cual los programadores describen conceptos con precisión, formulan algoritmos y razonan sobre soluciones. A lo largo de su carrera, un científico de la computación aprenderá y trabajará con muchos lenguajes diferentes, por separado o juntos. Los desarrolladores de software deben entender diferentes modelos de programación, características y construcciones de programación, y conceptos subyacentes para tomar decisiones de diseño informadas entre lenguajes que admiten múltiples enfoques complementarios. Sería útil saber cómo se definen, componen e implementan las características de los lenguajes de programación para mejorar la eficiencia de ejecución y el mantenimiento a largo plazo del software desarrollado. También es útil un conocimiento básico de la traducción de lenguajes, el análisis de programas, el comportamiento en tiempo de ejecución, la gestión de memoria y la interacción de procesos concurrentes que se comunican entre sí mediante paso de mensajes, memoria compartida y sincronización. Finalmente, algunos desarrolladores e investigadores necesitarán diseñar nuevos lenguajes, un ejercicio que requiere una mayor familiaridad con los principios básicos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.5.1 Fundamentos de Programación Orientada a Objetos (Pág. 72)	2	1
⭐ 3.5.2 Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (Pág. 73)	2	1
⭐ 3.5.3 Programación Funcional (Pág. 73)	4	3
⭐ 3.5.4 Programación Lógica (Pág. 74)		2
⭐ 3.5.5 Scripting de Shell (Pág. 75)	2	
⭐ 3.5.6 Programación Dirigida por Eventos y Reactiva (Pág. 75)	2	2
⭐ 3.5.7 Computación Paralela y Distribuida (Pág. 76)	2	2
3.5.8 Programación Orientada a Aspectos (Pág. 76)	Electivo	
⭐ 3.5.9 Fundamentos de Sistemas de Tipos (Pág. 77)	2	2
⭐ 3.5.10 Sistemas de Tipos: Polimorfismo y Tipado Complementario (Pág. 77)	1	1
⭐ 3.5.11 Teoría de Tipos Avanzada (Pág. 78)	2	2
⭐ 3.5.12 Modelo de Ejecución y Memoria de Sistemas (Pág. 78)	2	
⭐ 3.5.13 Gestión de Memoria (Pág. 79)	1	
⭐ 3.5.14 Traducción y Ejecución de Lenguajes (Pág. 79)	2	3
⭐ 3.5.15 Abstracción y Representación de Programas (Pág. 80)		3
3.5.16 Análisis Sintáctico (Pág. 81)	Electivo	
3.5.17 Análisis Semántico del Compilador (Pág. 81)	Electivo	
3.5.18 Análisis de Programas y Analizadores (Pág. 82)	Electivo	
3.5.19 Generación de Código (Pág. 82)	Electivo	
3.5.20 Comportamiento en Tiempo de Ejecución y Sistemas (Pág. 83)	Electivo	
3.5.21 Construcciones de Programación Avanzadas (Pág. 83)	Electivo	
3.5.22 Pragmática del Lenguaje (Pág. 84)	Electivo	
3.5.23 Semántica Formal (Pág. 84)	Electivo	
3.5.24 Metodologías de Desarrollo Formal (Pág. 85)	Electivo	
3.5.25 Principios de Diseño de Lenguajes de Programación (Pág. 85)	Electivo	
3.5.26 Sociedad, ética y Profesión (Pág. 86)	Electivo	

Cuadro 3.5: Lista de KUs del área de Fundamentos de los Lenguajes de Programación.

3.5.1. FPL/Fundamentos de Programación Orientada a Objetos (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Programación imperativa como subconjunto de la programación orientada a objetos.
- Diseño orientado a objetos: 1. Descomposición en objetos que poseen estado y tienen comportamiento. 2. Diseño de jerarquía de clases para modelado.
- Definición de clases: campos, métodos y constructores.
- Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos.
- Despacho dinámico: definición de llamada a método.
- Manejo de excepciones.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar las diferencias entre los paradigmas de programación imperativa y orientada a objetos [Enumerar]
2. Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Componer]
3. Construir una jerarquía de clases simple utilizando subclases que permita reutilizar código para subclases distintas [Crear]
4. Predecir y validar el flujo de control en un programa usando despacho dinámico [Evaluar]
5. Comparar y contrastar cómo difieren las soluciones computacionales a un problema en los enfoques procedural, funcional y orientado a objetos [Comparar]

6. Comparar y contrastar los mecanismos para definir y proteger elementos de datos dentro de los enfoques procedural, funcional y orientado a objetos [Comparar]

3.5.2. FPL/Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Idiomas orientados a objetos para encapsulación: 1. Privacidad, ocultación de datos y visibilidad de miembros de clase. 2. Interfaces que revelan solo firmas de métodos. 3. Clases base abstractas, rasgos (*traits*) y mixins.
- Propiedades dinámicas vs estáticas.
- Composición vs herencia.
- Subtipado: 1. Polimorfismo de subtipos; conversiones ascendentes (*upcasts*) implícitas en lenguajes tipados. 2. Noción de reemplazo conductual: los subtipos actúan como el supertipo. 3. Relación entre subtipado y herencia.
- Clases de colecciones, iteradores y otros componentes comunes de biblioteca.
- Metaprogramación y reflexión.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar y contrastar los beneficios y costos/impacto de usar herencia (subclases) y composición (específicamente, cómo basar la composición en funciones de orden superior) [Comparar]
2. Explicar la relación entre la herencia orientada a objetos (compartición de código y sobrescritura) y el subtipado (la idea de que un subtipo sea usable en un contexto que espera el supertipo) [Explicar]
3. Usar mecanismos de encapsulación orientados a objetos como interfaces y miembros privados [Usar]
4. Definir y usar iteradores y otras operaciones en agregados, incluyendo operaciones que toman funciones como argumentos, en múltiples lenguajes de programación, seleccionando los idiomas más naturales para cada lenguaje [Definir]
5. Usar clases de colecciones e iteradores efectivamente para resolver un problema [Usar]
6. Comparar y contrastar cómo difieren las soluciones computacionales a un problema en los enfoques procedural, funcional y orientado a objetos [Comparar]
7. Comparar y contrastar los mecanismos para definir y proteger elementos de datos dentro de los enfoques procedural, funcional y orientado a objetos [Comparar]

3.5.3. FPL/Programación Funcional (Core Tier1: 4 hrs, Core Tier2: 3 hrs)

Temas:
Core

- Expresiones lambda y evaluación: 1. Enlace (binding) de variables y reglas de ámbito (scope). 2. Paso de parámetros. 3. Expresiones lambda anidadas y orden de reducción.
- Programación libre de efectos secundarios: 1. Las llamadas a funciones no tienen efectos secundarios, facilitando el razonamiento composicional. 2. Variables inmutables y copia de datos vs reducción. 3. Uso de recursión vs bucles vs tuberías (map/reduce).
- Procesamiento de datos estructurados (por ejemplo, árboles) mediante funciones con casos para cada variante de datos: 1. Funciones definidas sobre datos compuestos en términos de funciones aplicadas a las piezas constituyentes. 2. Estructuras de datos persistentes.
- Usar funciones de orden superior (tomar, devolver y almacenar funciones).
- Metaprogramación y reflexión.
- Clausuras (*closures*) de funciones (funciones que usan variables en el entorno léxico circundante): 1. Significado y definición básica: crear clausuras en tiempo de ejecución capturando el entorno. 2. Idiomas canónicos: devoluciones de llamada (*callbacks*), argumentos para iteradores, código reutilizable mediante argumentos de funciones. 3. Usar una clausura para encapsular datos en su entorno. 4. Evaluación diferida (*lazy*) vs ansiosa (*eager*).

Non Core

- Máquina de reducción de grafos (graph reduction) y llamada por necesidad (*call-by-need*).
- Implementar evaluación diferida.
- Integración con el paradigma de programación lógica usando conceptos como lógica ecuacional, estrechamiento (narrowing), residuación (residuation) y unificación semántica.
- Integración con otros paradigmas de programación como imperativo y orientado a objetos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Desarrollar algoritmos básicos que eviten asignar a estados mutables o considerar igualdad referencial [Crear]
2. Desarrollar funciones útiles que tomen y devuelvan otras funciones [Crear]
3. Explicar un ejemplo simple de una expresión lambda implementada usando una máquina virtual, como una máquina SECD, mostrando el almacenamiento y recuperación del entorno [Explicar]
4. Interpretar correctamente las variables y el ámbito léxico en un programa que usa clausuras de funciones [Interpretar]
5. Usar mecanismos de encapsulación funcional como clausuras e interfaces modulares [Usar]
6. Comparar y contrastar ejecución con estado (stateful) vs sin estado (stateless) [Comparar]
7. Definir y usar iteradores y otras operaciones en agregados, incluyendo operaciones que toman funciones como argumentos, en múltiples lenguajes de programación, seleccionando los idiomas más naturales para cada lenguaje [Definir]

NonCore:

8. Ilustrar la reducción de grafos usando una expresión λ con una subexpresión compartida [Aplicar]
9. Ilustrar la ejecución de una expresión λ anidada simple usando una máquina abstracta, como una máquina ABC [Aplicar]
10. Ilustrar estrechamiento (narrowing), residuación y unificación semántica usando ejemplos ilustrativos simples [Aplicar]
11. Ilustrar las construcciones de concurrencia usando ejemplos de programación simples de conceptos conocidos, como un búfer siendo leído y escrito concurrentemente o secuencialmente [Aplicar]

3.5.4. FPL/Programación Lógica (Core Tier2: 2 hrs)

Temas:
Core

- Cuantificadores universales vs existenciales.
- Lógica de predicados de primer orden vs lógica de orden superior.
- Expresar relaciones complejas usando conectivos lógicos y relaciones más simples.
- Definiciones de cláusula de Horn, hechos (facts), metas (goals) y submetas (subgoals).
- Unificación y algoritmo de unificación; unificación vs aserción vs evaluación de expresiones.
- Mezclar relaciones con funciones.
- Cortes (cuts), retroceso (backtracking) y no determinismo.
- Suposiciones de mundo cerrado (closed-world) vs mundo abierto (open-world).

Non Core

- Sobrecarga de memoria por copia de variables al manejar programas iterativos.
- Construcciones de programación para almacenar cómputo parcial y podar árboles de búsqueda.
- Mezclar programación funcional y programación lógica usando conceptos como lógica ecuacional, estrechamiento (narrowing), residuación y unificación semántica.
- Programación lógica de orden superior, con restricciones (constraint) e inductiva.
- Integración con otros paradigmas de programación como la programación orientada a objetos.
- Construcciones de programación avanzadas como listas de diferencia (difference-lists), creación de estructuras de datos definidas por el usuario, conjunto de (set of), etc.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Usar un lenguaje lógico para implementar un algoritmo convencional [Usar]
2. Usar un lenguaje lógico para implementar un algoritmo que emplee búsqueda implícita usando cláusulas, relaciones y cortes (cuts) [Usar]

3. Usar un ejemplo ilustrativo simple para mostrar la correspondencia entre la Lógica de Predicados de Primer Orden (FOPL) y los programas lógicos que usan cláusulas de Horn [Usar]
4. Usar ejemplos para ilustrar el algoritmo de unificación y su papel en el paso de parámetros en la reducción de consultas [Usar]
5. Usar programas lógicos simples que entrelazan relaciones, funciones y programación recursiva, como números factoriales y de Fibonacci, y relaciones complejas simples entre entidades, e ilustrar la ejecución y paso de parámetros usando unificación y retroceso [Usar]

NonCore:

6. Ilustrar el cómputo de programas simples como Fibonacci y mostrar la sobrecarga por recomputación, y luego mostrar cómo mejorar la sobrecarga de ejecución [Aplicar]

3.5.5. FPL/Scripting de Shell (Core Tier1: 2 hrs)

Temas:

Core

- Manejo de errores/excepciones
- Tuberías (piping)
- Comandos del sistema: 1. Interfaz con sistemas operativos
- Variables de entorno
- Abstracción de archivo y operadores
- Estructuras de datos, como arreglos y listas.
- Expresiones regulares
- Programas y procesos
- Flujo de trabajo

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Crear]
2. Resolver varios problemas de procesamiento de texto mediante scripting [Resolver]

3.5.6. FPL/Programación Dirigida por Eventos y Reactiva (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:

Core

- Programación procedural vs programación reactiva: ventajas de la programación reactiva para capturar eventos.
- Componentes de la programación reactiva: fuente de eventos, señales de eventos, escuchas (listeners) y despachadores (dispatchers), objetos de evento, adaptadores, manejadores de eventos (event-handlers).
- Modelos sin estado (stateless) y de transición de estado (state-transition) de programación basada en eventos.
- Usos canónicos como interfaces gráficas de usuario (GUIs), dispositivos móviles, robots, servidores.
- Usar un framework reactivo: 1. Definir manejadores/escuchas de eventos 2. Parametrización de emisores de eventos y argumentos de eventos 3. Eventos generados externamente y eventos generados por el programa
- Separación de modelo, vista y controlador
- Programas dirigidos por eventos y reactivos como sistemas de transición de estado

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Implementar manejadores de eventos para usar en sistemas reactivos, como GUIs [Implementar]
2. Examinar por qué un estilo de programación dirigida por eventos es natural en dominios donde los programas reaccionan a eventos externos [Examinar]
3. Definir y usar un framework reactivo [Definir]
4. Describir un sistema interactivo en términos de un modelo, una vista y un controlador [Describir]

3.5.7. FPL/Computación Paralela y Distribuida (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Seguridad (safety) y vivacidad (liveness): 1. Condiciones de carrera (race conditions) 2. Dependencias/precondiciones 3. Modelos de fallos (fault models) 4. Terminación
- Modelos de programación: 1. Modelo de actores (*actor models*) 2. Modelos procedurales y reactivos 3. Modelos de programación síncrona/asíncrona 4. Paralelismo de datos (*data parallelism*)
- Propiedades: 1. Propiedades basadas en orden: *a*) Conmutatividad *b*) Independencia 2. Propiedades basadas en consistencia: *a*) Atomicidad *b*) Consenso
- Control de ejecución: 1. Espera asíncrona *Async await* 2. Promesas (*promises*) 3. Hilos (*threads*)
- Comunicación y coordinación: 1. Mutexes 2. Paso de mensajes (*message-passing*) 3. Memoria compartida (*shared memory*) 4. Cobegin-coend 5. Monitores 6. Canales (*channels*) 7. Hilos (*threads*) 8. Guardas (*guards*)
- Futuros (futures)
- Soporte del lenguaje para paralelismo de datos como forall, desenrollado de bucles (loop unrolling), map/reduce
- Efecto de los modelos de consistencia de memoria en la semántica del lenguaje y la generación correcta de código
- Interfaces de Programación de Aplicaciones de Transferencia de Estado Representacional (REST APIs)
- Tecnologías y enfoques: computación en la nube (*cloud computing*), computación de alto rendimiento, computación cuántica, computación ubicua
- Sobrecarga del paso de mensajes
- Granularidad del programa para una explotación eficiente de la concurrencia
- Concurrencia y otros paradigmas de programación (por ejemplo, funcional)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar por qué los lenguajes de programación no garantizan la consistencia secuencial en presencia de condiciones de carrera (data races) y qué deben hacer los programadores como resultado [Explicar]
2. Implementar programas concurrentes correctos usando múltiples modelos de programación, como memoria compartida, actores, futuros, construcciones de sincronización y primitivas de paralelismo de datos [Implementar]
3. Usar un modelo de paso de mensajes para analizar un protocolo de comunicación [Usar]
4. Usar construcciones de sincronización como métodos monitorizados/sincronizados en un programa simple [Usar]
5. Modelar dependencia de datos usando construcciones de programación simples que involucran variables, lectura y escritura [Implementar]
6. Modelar dependencia de control usando construcciones simples como selección e iteración [Implementar]
7. Explicar cómo las APIs REST integran aplicaciones y automatizan procesos [Explicar]
8. Explicar beneficios, limitaciones y desafíos relacionados con la computación distribuida y paralela [Explicar]

3.5.8. FPL/Programación Orientada a Aspectos

Temas:
Non Core

- Aspectos
- Puntos de unión (join points)
- Consejos (advice): 1. Antes (*before*) 2. Después (*after*) (como *finally*, *returning* o *throwing*) 3. Alrededor (*around*)

- Cortes de punto (point cuts): 1. Designadores
- Tejido (weaving) - estático y dinámico
- Alternativas incluyendo anotaciones y IDEs

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir los conceptos fundamentales de la programación orientada a aspectos, incluyendo aspectos, puntos de unión, consejos y cortes de punto [Describir]
2. Aplicar estrategias de tejido estático y dinámico para modularizar los conceptos transversales en un programa [Aplicar]
3. Analizar alternativas a la programación orientada a aspectos, como anotaciones y enfoques basados en IDEs, para el manejo de conceptos transversales [Analizar]

3.5.9. FPL/Fundamentos de Sistemas de Tipos (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Un tipo como un conjunto de valores junto con un conjunto de operaciones: 1. Tipos primitivos (por ejemplo, números, booleanos) 2. Tipos compuestos construidos a partir de otros tipos (por ejemplo, registros/estructuras, uniones, arreglos, listas, funciones, referencias usando operaciones de conjuntos)
- Asociación de tipos a variables, argumentos, resultados y campos
- Seguridad de tipos (type safety) como un aspecto de la corrección del programa
- Seguridad de tipos y errores causados por usar valores inconsistentemente dados sus tipos pretendidos
- Objetivos y limitaciones del tipado estático y dinámico: detectar y eliminar errores lo antes posible.
- Equivalencia de tipos: equivalencia estructural vs nominal

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir, tanto para un tipo primitivo como para uno compuesto, los valores que tienen ese tipo [Describir]
2. Describir, para un lenguaje con un sistema de tipos estático, las operaciones que están prohibidas estáticamente, como pasar el tipo incorrecto de valor a una función o método [Describir]
3. Describir ejemplos de errores de programa detectados por un sistema de tipos [Describir]
4. Identificar propiedades del programa, para múltiples lenguajes de programación, que se verifican estáticamente y propiedades del programa que se verifican dinámicamente [Analizar]
5. Describir un programa de ejemplo que no verifica tipos en un lenguaje particular y que sin embargo no tendría error si se ejecutara [Describir]
6. Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]
7. Explicar cómo las reglas de tipado definen el conjunto de operaciones que son legales para un tipo [Explicar]
8. Enumerar las reglas de tipo que gobiernan el uso de un tipo compuesto particular [Listar/Enumerar]
9. Explicar por qué la indecidibilidad requiere que los sistemas de tipos aproximen conservadoramente el comportamiento del programa [Explicar]
10. Explicar múltiples beneficios y limitaciones del tipado estático al escribir, mantener y depurar software [Explicar]

3.5.10. FPL/Sistemas de Tipos: Polimorfismo y Tipado Complementario (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Tipos genéricos (polimorfismo paramétrico): 1. Definición y ventajas del polimorfismo: paramétrico, de subtipado (subtyping), de sobrecarga (overloading) y de coerción 2. Comparación de tipos monomórficos y polimórficos 3. Comparación con polimorfismo ad-hoc (sobrecarga) y

polimorfismo de subtipos 4. Parámetros genéricos y tipado 5. Uso de bibliotecas genéricas como colecciones 6. Comparación con polimorfismo ad hoc (sobrecarga) y polimorfismo de subtipos 7. Polimorfismo prescriptivo vs descriptivo 8. Modelos de implementación de tipos polimórficos 9. Subtipado

- Beneficios complementarios del tipado estático y dinámico: 1. Errores tempranos vs errores tardíos/evitados 2. Hacer cumplir invariantes durante el desarrollo y mantenimiento del código vs posponer decisiones de tipado mientras se prototipa y permitir convenientemente patrones de codificación flexibles como colecciones heterogéneas. 3. Reglas de tipado para tipos de función, producto y suma 4. Evitar mal uso del código vs permitir más reutilización de código 5. Detectar programas incompletos vs permitir que programas incompletos se ejecuten 6. Relación con el análisis estático 7. Decidibilidad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir y usar piezas de programa (como funciones, clases, métodos) que usen tipos genéricos, incluyendo para colecciones [Definir]
2. Discutir las diferencias entre genéricos, subtipado y sobrecarga [Debatir]

3.5.11. FPL/Teoría de Tipos Avanzada (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Non Core

- Constructores de tipos composicionales, como tipos producto (para agregados), tipos suma (para uniones), tipos función, tipos cuantificados y tipos recursivos
- Verificación de tipos (type checking)
- Subtipado: 1. Polimorfismo de subtipos; conversiones ascendentes implícitas en lenguajes tipados 2. Noción de reemplazo conductual: los subtipos actúan como el supertipo 3. Relación entre subtipado y herencia
- Seguridad de tipos como preservación más progreso (preservation plus progress)
- Inferencia de tipos
- Sobrecarga estática
- Proposiciones como tipos (implicación como función, conjunción como producto, disyunción como suma)
- Tipos dependientes (cuantificación universal como función dependiente, cuantificación existencial como producto dependiente).

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Definir un sistema de tipos de manera precisa y composicional [Definir]
2. Para varios constructores de tipos fundamentales, identificar los valores que describen y los invariantes que hacen cumplir [Analizar]
3. Describir precisamente los invariantes preservados por un sistema de tipos sano (sound) [Describir]
4. Demostrar seguridad de tipos para un lenguaje simple en términos de teoremas de preservación y progreso [Probar/Demostrar]
5. Implementar un algoritmo de inferencia de tipos basado en unificación para un lenguaje simple [Implementar]
6. Explicar cómo la sobrecarga estática y los algoritmos de resolución asociados influyen en el comportamiento dinámico de los programas [Explicar]

3.5.12. FPL/Modelo de Ejecución y Memoria de Sistemas (Core Tier1: 2 hrs)

Temas:
Core

- Estructuras de datos para traducción, ejecución, traducción y movilidad de código como pila, montón (heap), alias (compartición usando punteros), secuencia indexada y cadena
- Acceso directo, indirecto e indexado a ubicación de memoria

- Representación en tiempo de ejecución de abstracciones de datos como variables, arreglos, vectores, registros, elementos de datos basados en punteros como listas enlazadas y árboles, y objetos

- Máquina de bajo nivel abstracta con instrucción simple, pila y montón para explicar traducción y ejecución
- Distribución en memoria en tiempo de ejecución: registro de activación (con varios punteros), datos estáticos, pila de llamadas, montón: 1. Traducción de construcciones de selección e iterativas a diagramas de flujo de control 2. Traducción de diagramas de flujo de control a código abstracto de bajo nivel 3. Implementación de bucles, recursión y llamadas de cola (tail calls) 4. Traducción de llamadas a funciones/procedimientos y retorno de llamadas, incluyendo diferentes mecanismos de paso de parámetros usando una máquina abstracta
- Computación verde (green computing).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Explicar]

3.5.13. FPL/Gestión de Memoria (Core Tier1: 1 hr)

Temas:
Core

- Asignación y acceso de bajo nivel a estructuras de datos de alto nivel como tipos de datos básicos, arreglos n-dimensionales, vectores, registros y objetos
- Retorno de procedimiento como mecanismo de desasignación automática para elementos de datos locales en la pila
- Gestión manual de memoria: asignar, desasignar y reutilizar memoria del montón
- Gestión automática de memoria: recolección de basura (garbage collection) como técnica automatizada usando la noción de alcanzabilidad (reachability)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar cómo las implementaciones de lenguajes de programación típicamente organizan la memoria en secciones de datos globales, texto, montón y pila, y cómo características como recursión y gestión de memoria se mapean a este modelo de memoria [Explicar]
2. Explicar por qué ocurren las fugas de memoria (memory leaks) y los problemas de punteros colgantes (dangling pointers), y qué puede hacer un programador para evitarlos/corregirlos [Explicar]

3.5.14. FPL/Traducción y Ejecución de Lenguajes (Core Tier1: 2 hrs, Core Tier2: 3 hrs)

Temas:
Core

- Modelos de ejecución para JIT (Just-In-Time), compilador, intérprete
- Uso de código intermedio, por ejemplo, bytecode
- Limitaciones y beneficios de JIT, compilador e intérprete
- Compiladores/transpiladores cruzados (cross compilers/transpilers)
- Representación BNF y BNF extendida de gramática libre de contexto
- árbol de análisis (parse tree) usando una oración simple como expresión aritmética o sentencia if-then-else
- Ejecución como código nativo o dentro de una máquina virtual
- Canalización de traducción de lenguaje: análisis sintáctico, análisis (parsing), verificación de tipos opcional, generación de código/traducción y optimización, enlace (linking), carga (loading), ejecución
- Representación en tiempo de ejecución de construcciones centrales del lenguaje como objetos (tablas de métodos) y funciones que pueden pasarse como parámetros a y devolverse desde funciones (clausuras)

- Desarrollo seguro de compiladores

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar y comprender las diferencias entre implementaciones de lenguaje compilado, JIT e interpretado, incluyendo los beneficios y limitaciones de cada una [Explicar]
2. Diferenciar sintaxis y análisis (parsing) de semántica y evaluación [Diferenciar]
3. Usar BNF y BNF extendida para especificar la sintaxis de construcciones simples como if-then-else, declaración de tipo y construcciones iterativas para lenguajes conocidos como C++ o Python [Usar]
4. Ilustrar el árbol de análisis usando una oración/expresión aritmética simple [Aplicar]
5. Ilustrar traducción de diagramas de sintaxis a BNF/BNF extendida para construcciones simples como if-then-else, declaración de tipo, construcciones iterativas, etc [Aplicar]
6. Ilustrar ambigüedad en el análisis usando if-then-else anidado/expresión aritmética y mostrar resolución usando orden de precedencia [Aplicar]
7. Discutir los beneficios y limitaciones de la recolección de basura, incluyendo la noción de alcanzabilidad [Debatir]

3.5.15. FPL/Abstracción y Representación de Programas (Core Tier2: 3 hrs)

Temas:
Core

- BNF y expresiones regulares
- Programas que toman (otros) programas como entrada, como intérpretes, compiladores, verificadores de tipos, generadores de documentación
- Componentes de un lenguaje: 1. Definiciones de alfabetos, delimitadores, oraciones, sintaxis y semántica 2. Sintaxis vs semántica
- Programa como un conjunto de oraciones significativas no ambiguas
- Abstracciones básicas de programación: constantes, variables, declaraciones (incluyendo declaraciones anidadas), comando, expresión, asignación, selección, iteración definida e indefinida, iteradores, función, procedimiento, módulos, manejo de excepciones
- Variables mutables vs inmutables: ventajas y desventajas de reutilizar ubicación de memoria existente vs ventajas de copiar y mantener valores antiguos; almacenar cómputo parcial vs recomputación
- Tipos de variables: estáticas, locales, no locales (*nonlocal*), globales; necesidad y problemas con variables no locales y globales.
- Reglas de ámbito (*scope*): estático vs dinámico; visibilidad de variables; efectos secundarios (*side-effects*).
- Efectos secundarios inducidos por variables no locales, variables globales y variables con alias (*aliased*).

Non Core

- Valores-L (*L-values*) y Valores-R (*R-values*): mapear nombre de variable mutable a Valores-L; mapear nombres de variables inmutables a Valores-R
- Entorno (*environment*) vs almacén (*store*) y sus propiedades
- Abstracción de datos y de control
- Mecanismos para intercambio de información entre unidades de programa como procedimientos, funciones y módulos: variables no locales, variables globales, paso de parámetros, importación-exportación entre módulos
- Estructuras de datos para representar código para ejecución, traducción o transmisión.
- Representación de instrucciones de bajo nivel como instrucciones de máquina virtual, lenguaje ensamblador y representación binaria
- Cálculo lambda, enlace de variables (variable binding) y renombrado de variables.
- Tipos de semántica: operacional, axiomática, denotacional, conductual; definir y usar árboles de sintaxis abstracta; contrastar con sintaxis concreta.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Ilustrar el ámbito de variables y visibilidad usando programas simples [Aplicar]

2. Ilustrar diferentes tipos de paso de parámetros usando un lenguaje de programación pseudo simple [Aplicar]
3. Explicar el efecto secundario usando variables globales y no locales y cómo corregir tales programas [Explicar]
4. Explicar cómo los programas que procesan otros programas tratan a los otros programas como sus datos de entrada [Explicar]
5. Describir una gramática y un árbol de sintaxis abstracta para un lenguaje pequeño [Describir]
6. Describir los beneficios de tener representaciones de programas distintas de cadenas de código fuente [Describir]
7. Implementar un programa para procesar alguna representación de código para algún propósito, como un intérprete, un optimizador de expresiones o un generador de documentación [Implementar]

3.5.16. FPL/Análisis Sintáctico

Temas:
Non Core

- Gramáticas regulares vs gramáticas libres de contexto
- Escaneo (*scanning*) y análisis (*parsing*) basados en especificaciones de lenguaje
- Análisis léxico usando expresiones regulares
- Tokens y su uso
- Estrategias de análisis (*parsing*) incluyendo técnicas descendentes (*top-down*) (por ejemplo, descenso recursivo o LL) y ascendentes (*bottom-up*) (por ejemplo, LR o GLR): 1. Tablas de anticipación (*lookahead*) y su aplicación al análisis
- Teoría de lenguajes: 1. Jerarquía de Chomsky 2. Derivación más a la izquierda/más a la derecha y ambigüedad 3. Transformación de gramáticas
- Mecanismos de recuperación de errores del analizador (parser)
- Generar escáneres y analizadores a partir de especificaciones declarativas

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Usar gramáticas formales para especificar la sintaxis de lenguajes [Usar]
2. Ilustrar el papel de las tablas de anticipación en el análisis [Aplicar]
3. Usar herramientas declarativas para generar analizadores y escáneres [Usar]
4. Reconocer problemas clave en definiciones de sintaxis: ambigüedad, asociatividad, precedencia [Reconocer]

3.5.17. FPL/Análisis Semántico del Compilador

Temas:
Non Core

- árboles de sintaxis abstracta; contraste con sintaxis concreta
- Definir, recorrer y modificar representaciones de programa de alto nivel
- ámbito (scope) y resolución de enlaces (binding)
- Semántica estática: 1. Verificación de tipos. 2. Definir antes de usar 3. Anotaciones y entornos (*frameworks*) de verificación estática extendida (*extended static checking*).
- Valores-L/Valores-R
- Semántica de llamada
- Tipos de paso de parámetros con ilustraciones simples y comparación: llamada por valor (call by value), llamada por referencia (call by reference), llamada por valor-resultado (call by value-result), llamada por nombre (call by name), llamada por necesidad (call by need) y sus variaciones
- Especificaciones declarativas como gramáticas atribuidas (attribute grammars) y sus aplicaciones en manejo de gramáticas sensibles al contexto limitadas

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Dibujar el árbol de sintaxis abstracta para un lenguaje pequeño [Diagramar]

2. Implementar análisis estáticos sensibles al contexto, a nivel de fuente, como verificadores de tipos o resolver identificadores para identificar sus ocurrencias de enlace (binding) [Implementar]
3. Describir análisis semánticos usando una gramática atribuida [Describir]

3.5.18. FPL/Análisis de Programas y Analizadores

Temas:
Non Core

- Representaciones de programa relevantes, como bloques básicos (basic blocks), grafos de flujo de control (control-flow graphs), cadenas definición-uso (def-use chains) y asignación única estática (static single assignment)
- Indecidibilidad y consecuencias para el análisis de programas
- Análisis insensible al flujo (flow-insensitive), como verificación de tipos y análisis de punteros y alias escalables
- Análisis sensible al flujo (flow-sensitive), como análisis de flujo de datos (dataflow) hacia adelante y hacia atrás
- Análisis sensible a la ruta (path-sensitive), como verificación de modelos de software (software model checking) y verificación de software
- Herramientas y frameworks para implementar analizadores
- Papel del análisis estático en la optimización de programas y el análisis de dependencia de datos durante la explotación de concurrencia
- Papel del análisis de programas en la verificación (parcial) y la búsqueda de errores (bug-finding)

- Paralelización: 1. Análisis para auto-paralelización 2. Análisis para detectar errores de concurrencia

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Explicar la diferencia entre grafo de flujo de datos (dataflow graph) y grafo de flujo de control [Explicar]
2. Explicar por qué los análisis de programa no triviales y sólidos (sound) deben ser aproximados [Explicar]
3. Argumentar por qué un análisis es correcto (sound y terminante) [Argumentar]
4. Explicar por qué el alias potencial limita el análisis de programa sólido y cómo el análisis de alias puede ayudar [Explicar]
5. Usar los resultados de un análisis de programa para optimización de programas y/o corrección parcial del programa [Usar]

3.5.19. FPL/Generación de Código

Temas:
Non Core

- Conjuntos de instrucciones
- Flujo de control
- Gestión de memoria
- Llamadas a procedimientos y despacho de métodos (method dispatching)
- Compilación separada; enlace (linking)
- Selección de instrucciones
- Planificación de instrucciones (por ejemplo, encadenamiento - pipelining)
- Asignación de registros
- Optimización de código como una forma de análisis de programas
- Generación de programas mediante IA generativa

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Identificar todos los pasos esenciales para convertir automáticamente código fuente en lenguaje ensamblador u otros lenguajes de bajo nivel [Analizar]

2. Explicar el código de bajo nivel necesario para llamar a funciones/métodos en lenguajes modernos [Explicar]
3. Discutir por qué la compilación separada requiere convenciones de llamada uniformes [Debatir]
4. Discutir por qué la compilación separada limita la optimización debido a efectos desconocidos de las llamadas [Debatir]
5. Discutir oportunidades para la optimización introducidas por la traducción ingenua y enfoques para lograr la optimización, como selección de instrucciones, planificación de instrucciones, asignación de registros y optimización de mirilla (peephole optimization) [Debatir]

3.5.20. FPL/Comportamiento en Tiempo de Ejecución y Sistemas

Temas:
Non Core

- Modelos de proceso usando pilas y montones para asignar y desasignar registros de activación y recuperar entornos usando punteros de marco (frame pointers) y direcciones de retorno durante una llamada a procedimiento, incluyendo ejemplos de paso de parámetros
- Esquemas de búsqueda de código usando tablas hash para métodos en implementaciones de programas orientados a objetos
- Distribución de datos para objetos y registros de activación
- Asignación de objetos en el montón (heap)
- Implementar entidades virtuales y métodos virtuales; tablas de métodos virtuales (virtual method tables) y su aplicación
- Comportamiento en tiempo de ejecución de programas orientados a objetos
- Comparar y contrastar la asignación de memoria durante el intercambio de información usando paso de parámetros y variables no locales (usando cadena de enlaces estáticos).
- Enfoques y técnicas de gestión dinámica de memoria: malloc/free, recolección de basura (mark-sweep, copying, reference counting), regiones (también conocidas como arenas o zonas)
- Compilación justo a tiempo (just-in-time) y recompilación dinámica
- Interfaz con el sistema operativo (por ejemplo, para inicialización del programa)
- Interoperabilidad entre lenguajes de programación incluyendo mecanismos de paso de parámetros y representación de datos: 1. *Big endian*, *little endian* 2. Distribución de datos de tipos de datos compuestos como arreglos
- Otras características comunes de las máquinas virtuales, como carga de clases (class loading), hilos (threads) y verificación de seguridad
- Sandboxing

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Discutir beneficios y limitaciones de la gestión automática de memoria [Debatir]
2. Explicar el uso de metadatos en representaciones en tiempo de ejecución de objetos y registros de activación, como punteros de clase, longitudes de arreglo, direcciones de retorno y punteros de marco (frame pointers) [Explicar]
3. Comparar y contrastar asignación estática vs asignación basada en pila vs asignación basada en montón de elementos de datos [Comparar]
4. Explicar por qué algunos elementos de datos no pueden desasignarse automáticamente al final de una llamada a procedimiento/método (necesidad de recolección de basura) [Explicar]
5. Discutir ventajas, desventajas y dificultades de la compilación justo a tiempo y la recompilación dinámica [Debatir]
6. Discutir el uso del sandboxing en código móvil [Debatir]
7. Identificar los servicios proporcionados por los sistemas en tiempo de ejecución de lenguajes modernos [Analizar]

3.5.21. FPL/Construcciones de Programación Avanzadas

Temas:
Non Core

- Mecanismos de encapsulación

- Evaluación diferida (delayed evaluation) y flujos infinitos (infinite streams)
- Comparar y contrastar evaluación diferida vs evaluación ansiosa (eager)
- Unificación vs aserción vs evaluación de expresiones
- Abstracciones de control: manejo de excepciones, continuaciones (continuations), mónadas (monads).
- Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods)
- Metaprogramación: macros, programación generativa, desarrollo basado en modelos (model-based development)
- Manipulación de cadenas mediante coincidencia de patrones (pattern-matching) (expresiones regulares)
- Evaluación dinámica de código ("eval")
- Soporte del lenguaje para verificar aserciones, invariantes y pre/post-condiciones
- Lenguajes específicos del dominio, como lenguajes de bases de datos, lenguajes de ciencia de datos, lenguajes de computación embebida, lenguajes síncronos, lenguajes de interfaz de hardware

- Modelos y lenguajes de computación de alto rendimiento masivamente paralelos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Usar varias construcciones e idiomas de programación avanzados correctamente [Usar]
2. Discutir cómo varias construcciones de programación avanzadas tienen como objetivo mejorar la estructura del programa, la calidad del software y la productividad del programador [Debatir]
3. Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Debatir]

3.5.22. FPL/Pragmática del Lenguaje

Temas:
Non Core

- Efecto de las necesidades tecnológicas y los requisitos de software en el desarrollo y evolución de los lenguajes de programación
- Dominios de problemas y paradigma de programación
- Criterios para un buen diseño de lenguaje de programación: 1. Principios de diseño de lenguajes como la ortogonalidad 2. Definir construcciones de control e iteración 3. Modularización de software grande
- Orden de evaluación, precedencia y asociatividad
- Evaluación ansiosa vs diferida
- Definir construcciones de control e iteración
- Llamadas externas y bibliotecas del sistema

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Discutir el papel de conceptos como la ortogonalidad y valores por defecto bien elegidos en el diseño de lenguajes [Debatir]
2. Evaluar objetivamente y justificar decisiones de diseño de lenguaje [Evaluar]
3. Implementar un programa de ejemplo cuyo resultado pueda diferir bajo diferentes reglas para el orden de evaluación, precedencia o asociatividad [Implementar]
4. Ilustrar usos de la evaluación diferida, como abstracciones de control definidas por el usuario [Aplicar]
5. Discutir la necesidad de permitir llamadas a llamadas externas y bibliotecas del sistema y las consecuencias para la implementación del lenguaje [Debatir]

3.5.23. FPL/Semántica Formal

Temas:
Non Core

- Sintaxis vs semántica

- Enfoques de semántica: axiomática, operacional, denotacional, basada en tipos
- Semántica axiomática de construcciones abstractas como asignación, selección, iteración usando precondición, postcondiciones e invariante de bucle
- Análisis de semántica operacional de construcciones abstractas y secuencia de tales como asignación, evaluación de expresión, selección, iteración usando entorno y almacén: 1. Ejecución simbólica (symbolic execution) 2. Verificadores de restricciones (constraint checkers)
- Semántica denotacional: 1. Cálculo Lambda.
- Demostraciones por inducción sobre la semántica del lenguaje
- Definiciones y demostraciones formales para sistemas de tipos: 1. Proposiciones como tipos (implicación como función, conjunción como producto, disyunción como suma) 2. Tipos dependientes (cuantificación universal como función dependiente, cuantificación existencial como producto dependiente) 3. Parametricidad

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Construir semántica formal para un lenguaje pequeño [Crear]
2. Escribir un programa de cálculo lambda y mostrar su evaluación a una forma normal [Escribir]
3. Discutir los diferentes enfoques de la semántica operacional, denotacional y axiomática [Debatir]
4. Usar inducción para demostrar propiedades de todos los programas en un lenguaje [Usar]
5. Usar inducción para demostrar propiedades de todos los programas en un lenguaje que está bien tipado según un sistema de tipos definido formalmente [Usar]
6. Usar parametricidad para establecer el comportamiento del código dado solo su tipo [Usar]

3.5.24. FPL/Metodologías de Desarrollo Formal

Temas:
Non Core

- Lenguajes y metodologías de especificación formal
- Demostradores de teoremas (theorem provers), asistentes de prueba (proof assistants) y lógicas
- Verificadores de restricciones (constraint checkers)
- Tipos dependientes (cuantificación universal como función dependiente, cuantificación existencial como producto dependiente)
- Especificación y descarga de pruebas (proof discharge) para sistemas de software completamente verificados usando pre/post condiciones, tipos de refinamiento (refinement types), etc.
- Modelado formal e implementación/refinamiento manual de sistemas de software.
- Uso de pruebas simbólicas (symbolic testing) y fuzzing en el desarrollo de software.
- Verificación de modelos (model checking).
- Comprensión de situaciones donde los métodos formales pueden aplicarse efectivamente y cómo estructurar el desarrollo para maximizar su valor.

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Usar técnicas de modelado formal para desarrollar y validar arquitecturas [Usar]
2. Usar lenguajes de programación asistidos por pruebas (proof assisted) para desarrollar artefactos de software completamente especificados y verificados [Usar]
3. Usar soporte de verificador y especificación en lenguajes de programación para validar formalmente propiedades del sistema [Usar]
4. Integrar herramientas de validación simbólica en un flujo de trabajo de programación [Integrar]
5. Discutir cuándo y cómo los métodos formales pueden usarse efectivamente en el proceso de desarrollo [Debatir]

3.5.25. FPL/Principios de Diseño de Lenguajes de Programación

Temas:
Non Core

- Principios de diseño de lenguajes: 1. Simplicidad 2. Seguridad 3. Traducción rápida 4. Código objeto eficiente 5. Ortogonalidad 6. Legibilidad 7. Completitud 8. Estrategias de implementación

- Diseñar un lenguaje para ajustarse a un dominio o problema específico
- Interoperabilidad entre lenguajes de programación
- Portabilidad del lenguaje
- Descripción formal de un lenguaje de programación
- Principios de computación verde (green computing)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Entender qué constituye un buen diseño de lenguaje y aplicar ese conocimiento para evaluar un lenguaje de programación real [Evaluar]

3.5.26. FPL/Sociedad, ética y Profesión

Temas:
Non Core

- Impacto de los lenguajes de programación centrados en inglés
- Mejorar la accesibilidad e inclusión para personas con discapacidades - Compatibilidad con tecnologías de asistencia
- Factores humanos relacionados con los lenguajes de programación y la usabilidad: 1. Impacto de la sintaxis en la accesibilidad 2. Compatibilidad con diferencias culturales (por ejemplo, moneda, decimales, fechas) 3. Neurodiversidad
- Etimología de términos como ”clase” (class), ”maestro” (master) y ”esclavo” (slave) en lenguajes de programación
- Aumentar la accesibilidad mediante el soporte de múltiples idiomas dentro de las aplicaciones (UTF)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Diseñar conscientemente lenguajes de programación para que sean inclusivos y no ofensivos [Diseñar]

3.6. Gráficos y Técnicas Interactivas (GIT)

Los Gráficos y Técnicas Interactivas (GIT) engloban los principios, algoritmos y tecnologías involucrados en el renderizado, modelado, animación e interacción con contenido visual e interactivo. Esta área de conocimiento integra conceptos de gráficos por computadora, interacción humano-computadora, visualización y procesamiento de imágenes para crear experiencias digitales inmersivas y efectivas. Con la proliferación de aplicaciones interactivas en entretenimiento, educación, visualización científica e interfaces de usuario, GIT juega un papel crucial en la computación moderna. Los estudiantes que estudian GIT aprenden a diseñar, implementar y evaluar sistemas que generan contenido visual, facilitan la interacción del usuario y aprovechan tecnologías emergentes como la realidad virtual, la realidad aumentada y la computación tangible. El área enfatiza tanto los fundamentos teóricos como las habilidades prácticas, preparando a los graduados para contribuir en diversos campos, incluyendo el desarrollo de videojuegos, simulación, visualización de datos y medios interactivos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.6.1 Conceptos Fundamentales de Gráficos y Técnicas Interactivas (Pág. 87)	4	3
⭐ 3.6.2 Visualización (Pág. 87)		6
⭐ 3.6.3 Renderizado Aplicado y Técnicas (Pág. 88)		15
⭐ 3.6.4 Modelado Geométrico (Pág. 88)		6
⭐ 3.6.5 Sombreado y Renderizado Avanzado (Pág. 88)		6
⭐ 3.6.6 Animación por Computadora (Pág. 89)		6
⭐ 3.6.7 Simulación (Pág. 89)		6
⭐ 3.6.8 Inmersión (Pág. 89)		6
⭐ 3.6.9 Interacción (Pág. 89)		4
⭐ 3.6.10 Procesamiento de Imágenes (Pág. 90)		6
⭐ 3.6.11 Computación Tangible/Física (Pág. 90)		6
3.6.12 Sociedad, ética y Profesión (Pág. 90)	Electivo	

Cuadro 3.6: Lista de KUs del área de Gráficos y Técnicas Interactivas.

3.6.1. GIT/Conceptos Fundamentales de Gráficos y Técnicas Interactivas (Core Tier1: 4 hrs, Core Tier2: 3 hrs)

Temas:
Core

- Descripción general del pipeline de gráficos por computadora y sistemas interactivos
- Conceptos básicos de renderizado: rasterización, trazado de rayos (*ray tracing*), modelos de sombreado
- Teoría del color y espacios de color
- Sistemas de coordenadas y transformaciones
- Técnicas básicas de interacción: dispositivos de entrada, manejo de eventos
- Conceptos avanzados de renderizado: iluminación global (*global illumination*), renderizado basado en física
- Hardware gráfico y técnicas de aceleración
- Percepción humana y cognición visual

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las etapas básicas del pipeline de gráficos por computadora [Explicar]
2. Describir técnicas básicas de renderizado como rasterización y trazado de rayos [Describir]
3. Aplicar teoría del color para diseñar gráficos visualmente efectivos [Aplicar]
4. Implementar transformaciones de coordenadas para gráficos 2D y 3D [Implementar]
5. Diseñar aplicaciones interactivas básicas utilizando dispositivos de entrada comunes y manejo de eventos [Diseñar]
6. Comparar técnicas avanzadas de renderizado como iluminación global y renderizado basado en física [Comparar]
7. Analizar el rol del hardware gráfico en acelerar tareas de renderizado [Analizar]
8. Explicar cómo la percepción humana influye en el diseño de gráficos e interacción [Explicar]

3.6.2. GIT/Visualización (Core Tier2: 6 hrs)

Temas:
Core

- Principios de visualización de datos: percepción, cognición y codificación efectiva
- Tipos de visualizaciones: científicas, de información, geográficas
- Técnicas de visualización para datos multivariados, temporales y jerárquicos
- Visualización interactiva y exploración
- Evaluación de la efectividad de la visualización

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar principios de percepción y cognición para diseñar visualizaciones efectivas [Aplicar]
2. Seleccionar técnicas de visualización apropiadas para diferentes tipos de datos y tareas [Evaluar]
3. Implementar visualizaciones interactivas para exploración de datos [Implementar]
4. Evaluar la efectividad de una visualización usando métodos empíricos [Evaluar]

3.6.3. GIT/Renderizado Aplicado y Técnicas (Core Tier2: 15 hrs)

Temas:
Core

- Algoritmos de rasterización: conversión por barrido (*scan conversion*), antialiasing
- Trazado de rayos (*ray tracing*): estructuras de aceleración, muestreo (*sampling*), sombreado
- Mapeo de texturas (*texture mapping*) y filtrado
- Modelos de sombreado: Lambertiano, Phong, BRDF
- Técnicas avanzadas: mapeo de sombras (*shadow mapping*), mapeo ambiental (*environment mapping*), renderizado diferido (*deferred rendering*)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar algoritmos de rasterización para renderizado de primitivas [Implementar]
2. Implementar trazado de rayos básico con estructuras de aceleración [Implementar]
3. Aplicar técnicas de mapeo de texturas y filtrado para mejorar el realismo visual [Aplicar]
4. Comparar diferentes modelos de sombreado y sus aplicaciones [Comparar]
5. Implementar técnicas avanzadas de renderizado como mapeo de sombras o renderizado diferido [Implementar]

3.6.4. GIT/Modelado Geométrico (Core Tier2: 6 hrs)

Temas:
Core

- Representaciones geométricas: polígonos, curvas, superficies, formas implícitas
- Técnicas de modelado: curvas spline, superficies de subdivisión, geometría sólida constructiva
- Modelado procedural y generación
- Algoritmos geométricos: intersección, distancia, descomposición

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir diferentes representaciones geométricas y sus compensaciones (trade-offs) [Describir]
2. Implementar técnicas de modelado como curvas spline o superficies de subdivisión [Implementar]
3. Crear modelos procedurales usando algoritmos o reglas [Crear]
4. Aplicar algoritmos geométricos para resolver problemas como pruebas de intersección o procesamiento de mallas (mesh) [Aplicar]

3.6.5. GIT/Sombreado y Renderizado Avanzado (Core Tier2: 6 hrs)

Temas:
Core

- Sombreado avanzado: sombreado basado en física, dispersión subsuperficial (subsurface scattering)
- Iluminación global (global illumination): radiosidad, mapeo de fotones (photon mapping), trazado de caminos (path tracing)
- Técnicas de renderizado en tiempo real: nivel de detalle (level of detail), culling, oclusión
- Programación GPU y sombreadores (shaders)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar modelos de sombreado avanzados para materiales realistas [Implementar]
2. Comparar algoritmos de iluminación global y sus compensaciones [Comparar]
3. Optimizar el renderizado en tiempo real usando técnicas como nivel de detalle y *culling* [Crear]
4. Desarrollar sombreadores GPU para efectos de renderizado personalizados [Crear]

3.6.6. GIT/Animación por Computadora (Core Tier2: 6 hrs)

Temas:
Core

- Principios de animación: tiempo (timing), espaciado (spacing), anticipación, seguimiento (follow-through)
- Animación por fotogramas clave (keyframe) e interpolación
- Animación procedural: sistemas de partículas, comportamiento de bandadas (flocking), basada en física
- Animación de personajes: animación esquelética, skinning, cinemática inversa
- Captura de movimiento (motion capture) y retargeting

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar principios de animación para crear movimiento creíble [Aplicar]
2. Implementar animación por fotogramas clave con técnicas de interpolación [Implementar]
3. Crear animaciones procedurales como sistemas de partículas o simulaciones basadas en física [Crear]
4. Implementar animación de personajes usando sistemas esqueléticos y skinning [Implementar]
5. Procesar datos de captura de movimiento para retargeting de animación [Implementar]

3.6.7. GIT/Simulación (Core Tier2: 6 hrs)

Temas:
Core

- Simulación basada en física: cuerpos rígidos, fluidos, tela
- Métodos numéricos para simulación: integración, detección de colisiones
- Restricciones y optimizaciones de simulación en tiempo real
- Aplicaciones en videojuegos, efectos visuales y visualización científica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar simulaciones basadas en física para cuerpos rígidos o fluidos [Implementar]
2. Aplicar métodos de integración numérica para simulaciones dinámicas [Aplicar]
3. Optimizar simulaciones para rendimiento en tiempo real [Crear]
4. Desarrollar simulaciones para aplicaciones específicas como videojuegos o visualización científica [Crear]

3.6.8. GIT/Inmersión (Core Tier2: 6 hrs)

Temas:
Core

- Conceptos de realidad virtual (VR) y realidad aumentada (AR)
- Tecnologías de visualización inmersiva: pantallas montadas en la cabeza (head-mounted displays), CAVEs
- Seguimiento (tracking) y entrada para entornos inmersivos
- Presencia y comodidad en sistemas inmersivos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar los conceptos centrales y diferencias entre VR y AR [Explicar]
2. Evaluar tecnologías de visualización inmersiva y sus aplicaciones [Evaluar]
3. Implementar soluciones de seguimiento y entrada para entornos inmersivos [Implementar]
4. Diseñar experiencias inmersivas que maximicen la presencia y minimicen la incomodidad [Diseñar]

3.6.9. GIT/Interacción (Core Tier2: 4 hrs)

Temas:
Core

- Paradigmas de interacción: manipulación directa, basada en gestos, tangible
- Diseño de interfaz de usuario para aplicaciones gráficas
- Interacción multimodal: combinando retroalimentación visual, auditiva, háptica
- Evaluación de técnicas de interacción: usabilidad, experiencia de usuario

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar diferentes paradigmas de interacción y su idoneidad para varias tareas [Comparar]
2. Diseñar interfaces de usuario para aplicaciones intensivas en gráficos [Diseñar]
3. Implementar sistemas de interacción multimodal incorporando múltiples canales de retroalimentación [Implementar]
4. Realizar evaluaciones de usabilidad de técnicas de interacción [Evaluar]

3.6.10. GIT/Procesamiento de Imágenes (Core Tier2: 6 hrs)

Temas:
Core

- Representación y formatos de imagen
- Filtrado y mejora: convolución, dominio de frecuencia
- Segmentación de imágenes y detección de características
- Operaciones morfológicas
- Aplicaciones en gráficos, visión y multimedia

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Manipular imágenes usando diferentes representaciones y formatos [Procesar]
2. Aplicar técnicas de filtrado para mejora y análisis de imágenes [Aplicar]
3. Implementar algoritmos de segmentación de imágenes para extracción de características [Implementar]
4. Usar operaciones morfológicas para tareas de procesamiento de imágenes [Usar]
5. Desarrollar aplicaciones que integren procesamiento de imágenes con sistemas de gráficos o visión [Crear]

3.6.11. GIT/Computación Tangible/Física (Core Tier2: 6 hrs)

Temas:
Core

- Interfaces tangibles y computación física
- Sensores y actuadores para sistemas interactivos
- Sistemas embebidos para arte y diseño interactivo
- Instalaciones y entornos interactivos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar interfaces tangibles que unan interacciones físicas y digitales [Diseñar]
2. Integrar sensores y actuadores para crear sistemas responsivos [Integrar]
3. Desarrollar sistemas embebidos para proyectos de arte o diseño interactivo [Crear]
4. Crear instalaciones interactivas que involucren a los usuarios en espacios físicos [Crear]

3.6.12. GIT/Sociedad, ética y Profesión

Temas:
Core

- Consideraciones éticas en gráficos e interacción: representación, accesibilidad, privacidad
- Impacto social de las tecnologías inmersivas y la visualización
- Práctica profesional en industrias de gráficos e interactivas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar problemas éticos relacionados con representación, accesibilidad y privacidad en gráficos e interacción [Analizar]
2. Discutir las implicaciones sociales de las tecnologías inmersivas y la visualización de datos [Debatir]
3. Describir prácticas y estándares profesionales en industrias de gráficos e interactivas [Describir]

3.7. Interacción Humano-Computadora (HCI)

La Interacción Humano-Computadora (HCI) es un campo multidisciplinario que se centra en el diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano, y el estudio de los principales fenómenos que los rodean. HCI integra conocimientos y métodos de la informática, psicología, diseño, sociología y otros campos para crear tecnologías que sean útiles, usables y satisfactorias para las personas. El objetivo central de HCI es mejorar las interacciones entre usuarios y computadoras haciendo los sistemas más usables, accesibles y eficientes, considerando al mismo tiempo las capacidades y limitaciones humanas.

En el mundo actual, donde la tecnología informática impregna todos los aspectos de la vida diaria-desde teléfonos inteligentes y sitios web hasta dispositivos médicos y sistemas de hogares inteligentes-los principios de HCI son esenciales para crear sistemas que las personas puedan usar de manera efectiva y segura. HCI enfatiza procesos de diseño centrados en el usuario, prototipado iterativo y evaluación empírica para asegurar que los sistemas satisfagan las necesidades reales de los usuarios. El campo aborda diversos temas que incluyen usabilidad, experiencia de usuario (UX), accesibilidad, diseño de interacción, métodos de investigación de usuarios y las implicaciones sociales de la tecnología.

HCI prepara a los estudiantes para diseñar, implementar y evaluar sistemas interactivos que no solo sean funcionales sino que también consideren factores humanos, implicaciones éticas e impacto social. Los graduados con experiencia en HCI trabajan en roles como diseñadores de UX, diseñadores de interacción, ingenieros de usabilidad y gerentes de productos en diversas industrias, incluyendo desarrollo de software, atención médica, educación, entretenimiento y políticas públicas.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.7.1 Comprender al Usuario: Objetivos individuales e interacciones con otros (Pág. 91)	2	5
⭐ 3.7.2 Responsabilidad y Rendición de Cuentas en el Diseño (Pág. 92)	2	2
⭐ 3.7.3 Accesibilidad y Diseño Inclusivo (Pág. 92)	2	2
⭐ 3.7.4 Evaluación del Diseño (Pág. 92)	1	2
⭐ 3.7.5 Diseño de Sistemas (Pág. 93)	1	5
3.7.6 Sociedad, ética y Profesión (Pág. 93)	Electivo	

Cuadro 3.7: Lista de KUs del área de Interacción Humano-Computadora.

3.7.1. HCI/Comprender al Usuario: Objetivos individuales e interacciones con otros (Core Tier1: 2 hrs, Core Tier2: 5 hrs)

Temas:
Core

- Capacidades y limitaciones humanas: percepción, cognición, habilidades motoras
- Características del usuario: demografía, experiencia, contexto cultural
- Objetivos, tareas y contextos de uso del usuario
- Aspectos sociales y organizacionales de la interacción
- Modelos y teorías cognitivas relevantes para HCI
- Aspectos emocionales y afectivos de la interacción
- Métodos de investigación de usuarios: entrevistas, encuestas, observaciones

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las capacidades y limitaciones humanas que afectan el diseño de interacción [Describir]

2. Analizar cómo las características del usuario influyen en la interacción con sistemas informáticos [Analizar]
3. Identificar objetivos y tareas del usuario en contextos de uso específicos [Analizar]
4. Explicar factores sociales y organizacionales que impactan la adopción y uso de la tecnología [Explicar]
5. Aplicar modelos cognitivos para predecir y explicar el comportamiento del usuario [Aplicar]
6. Considerar factores emocionales y afectivos en el diseño de interacción [Analizar]
7. Realizar investigación de usuarios usando métodos apropiados como entrevistas u observaciones [Evaluar]

3.7.2. HCI/Responsabilidad y Rendición de Cuentas en el Diseño (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Consideraciones éticas en HCI: equidad, transparencia, rendición de cuentas
- Diseño para valores y diseño sensible a valores
- Innovación responsable y ética del diseño
- Rendición de cuentas algorítmica y sesgo en sistemas interactivos
- Responsabilidad profesional en la práctica de HCI
- Consideraciones legales y regulatorias para sistemas interactivos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar problemas éticos en el diseño e implementación de HCI [Analizar]
2. Aplicar enfoques de diseño sensible a valores al desarrollo tecnológico [Aplicar]
3. Discutir prácticas de innovación responsable en HCI [Debatir]
4. Analizar la rendición de cuentas algorítmica y el sesgo en sistemas interactivos [Analizar]
5. Describir responsabilidades profesionales en la práctica de HCI [Describir]
6. Explicar consideraciones legales y regulatorias relevantes para sistemas interactivos [Explicar]

3.7.3. HCI/Accesibilidad y Diseño Inclusivo (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Principios y pautas de accesibilidad
- Discapacidades y tecnologías de asistencia
- Diseño inclusivo y principios de diseño universal
- Métodos de evaluación para accesibilidad
- Diseñar para diversas capacidades y contextos
- Estándares de accesibilidad y cumplimiento

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar principios de accesibilidad al diseño de sistemas interactivos [Aplicar]
2. Describir cómo diferentes discapacidades afectan la interacción con la tecnología [Describir]
3. Diseñar sistemas inclusivos usando principios de diseño universal [Diseñar]
4. Realizar evaluaciones de accesibilidad de sistemas interactivos [Evaluar]
5. Crear diseños que acomoden diversas capacidades y contextos de uso [Crear]
6. Garantizar el cumplimiento con estándares y regulaciones de accesibilidad [Evaluar]

3.7.4. HCI/Evaluación del Diseño (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Principios y heurísticas de usabilidad
- Métodos de evaluación: evaluación heurística, recorrido cognitivo (cognitive walkthrough)
- Pruebas con usuarios: planificación, ejecución, análisis

- Métodos avanzados de evaluación: experimentos controlados, estudios de campo
- Métricas y medición de UX
- Métodos de evaluación analítica: modelado y simulación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar principios y heurísticas de usabilidad para evaluar diseños [Aplicar]
2. Realizar evaluaciones heurísticas y recorridos cognitivos [Evaluar]
3. Planificar, conducir y analizar sesiones de pruebas con usuarios [Planear]
4. Diseñar y conducir experimentos controlados para investigación en HCI [Diseñar]
5. Medir la experiencia de usuario usando métricas apropiadas [Evaluar]
6. Usar métodos analíticos para modelar y predecir el desempeño del usuario [Usar]

3.7.5. HCI/Diseño de Sistemas (Core Tier1: 1 hr, Core Tier2: 5 hrs)

Temas:
Core

- Proceso de diseño: diseño centrado en el usuario, diseño iterativo
- Diseño de interacción: conceptos, principios, patrones
- Técnicas de prototipado: baja fidelidad, alta fidelidad, interactivo
- Arquitectura de información y diseño de navegación
- Técnicas avanzadas de interacción: gestos, voz, multimodal
- Diseño para dominios específicos: móvil, web, computación ubicua
- Sistemas de computación colaborativa y social

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar procesos de diseño centrado en el usuario para desarrollar sistemas interactivos [Aplicar]
2. Diseñar interacciones usando principios y patrones establecidos [Diseñar]
3. Crear prototipos en diferentes niveles de fidelidad para exploración de diseño [Crear]
4. Estructurar información y navegación para una interacción efectiva del usuario [Estructurar]
5. Implementar técnicas avanzadas de interacción como interfaces de gestos o voz [Implementar]
6. Diseñar sistemas interactivos para dominios y plataformas específicos [Diseñar]
7. Diseñar sistemas colaborativos que apoyen la interacción grupal [Diseñar]

3.7.6. HCI/Sociedad, ética y Profesión

Temas:
Core

- Impacto social de las tecnologías interactivas
- Problemas éticos en la investigación y práctica de HCI
- Práctica profesional en HCI
- Consideraciones culturales e interculturales en HCI
- Sostenibilidad y HCI
- Tendencias futuras y desafíos en HCI

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar el impacto social de las tecnologías interactivas en individuos y sociedad [Analizar]
2. Abordar problemas éticos en proyectos de investigación y diseño de HCI [Evaluar]
3. Describir prácticas profesionales y trayectorias profesionales en HCI [Describir]
4. Diseñar sistemas que tomen en cuenta diferencias y contextos culturales [Diseñar]
5. Incorporar consideraciones de sostenibilidad en el diseño de HCI [Integrar]
6. Discutir tendencias emergentes y desafíos futuros en HCI [Debatir]

3.8. Fundamentos Matemáticos y Estadísticos (MSF)

Los fundamentos matemáticos y estadísticos de la informática proporcionan las bases teóricas para la disciplina. Incluyen matemáticas discretas, probabilidad, estadística, álgebra lineal y cálculo.

Estas áreas son esenciales para comprender algoritmos, estructuras de datos, aprendizaje automático, gráficos, criptografía y muchos otros temas en informática.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.8.1 Matemáticas Discretas (Pág. 94)	15	6
⭐ 3.8.2 Matemáticas Discretas Avanzadas (Pág. 94)	15	6
⭐ 3.8.3 Probabilidad (Pág. 95)	11	29
⭐ 3.8.4 Estadística (Pág. 95)	10	30
⭐ 3.8.5 álgebra Lineal (Pág. 96)	5	35
⭐ 3.8.6 Cálculo (Pág. 96)		40

Cuadro 3.8: Lista de KUs del área de Fundamentos Matemáticos y Estadísticos.

3.8.1. MSF/Matemáticas Discretas (Core Tier1: 15 hrs, Core Tier2: 6 hrs)

Temas:
Core

- Conjuntos: operaciones, cardinalidad, conjuntos potencia, productos cartesianos
- Relaciones: propiedades, relaciones de equivalencia, órdenes parciales
- Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa
- Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones
- Grafos: definiciones, caminos, ciclos, árboles, conectividad
- Conteo: permutaciones, combinaciones, teorema del binomio
- Relaciones de recurrencia: resolver recurrencias simples

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir operaciones y relaciones básicas de conjuntos [Explicar]
2. Aplicar operaciones de conjuntos para resolver problemas [Aplicar]
3. Definir propiedades de las relaciones [Explicar]
4. Determinar si una relación es una relación de equivalencia [Aplicar]
5. Definir tipos de funciones [Explicar]
6. Componer funciones y encontrar inversas [Aplicar]
7. Explicar lógica proposicional y de predicados [Explicar]
8. Construir demostraciones lógicas simples [Aplicar]
9. Definir conceptos básicos de grafos [Explicar]
10. Resolver problemas básicos de grafos [Aplicar]
11. Explicar permutaciones y combinaciones [Explicar]
12. Aplicar técnicas de conteo para resolver problemas [Aplicar]
13. Definir relaciones de recurrencia [Explicar]
14. Resolver relaciones de recurrencia simples [Aplicar]

3.8.2. MSF/Matemáticas Discretas Avanzadas (Core Tier1: 15 hrs, Core Tier2: 6 hrs)

Temas:
Core

- Teoría avanzada de conjuntos: conjuntos infinitos, argumento diagonal de Cantor
- Relaciones avanzadas: clausuras, retículos
- Funciones avanzadas: funciones recursivas, computabilidad
- Lógica avanzada: sistemas de demostración, completitud, decidibilidad
- Teoría avanzada de grafos: planaridad, coloración, redes
- Conteo avanzado: funciones generadoras, principio de inclusión-exclusión
- Recurrencia avanzada: funciones generadoras, comportamiento asintótico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar conceptos de conjuntos infinitos y cardinalidad [Explicar]

2. Aplicar operaciones de clausura a relaciones [Aplicar]
3. Analizar propiedades avanzadas de grafos [Evaluar]
4. Desarrollar funciones generadoras para problemas de conteo [Crear]

3.8.3. MSF/Probabilidad (Core Tier1: 11 hrs, Core Tier2: 29 hrs)

Temas:
Core

- Probabilidad básica: espacios muestrales, eventos, axiomas
- Probabilidad condicional, independencia, teorema de Bayes
- Variables aleatorias: discretas y continuas, distribuciones
- Esperanza, varianza, desviación estándar
- Distribuciones comunes: uniforme, binomial, normal
- Distribuciones conjuntas, covarianza, correlación
- Teoremas límite: ley de los grandes números, teorema del límite central
- Procesos estocásticos: cadenas de Markov
- Teoría de la información: entropía, información mutua

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir conceptos básicos de probabilidad [Explicar]
2. Calcular probabilidades de eventos [Aplicar]
3. Definir probabilidad condicional e independencia [Explicar]
4. Aplicar el teorema de Bayes para actualizar probabilidades [Aplicar]
5. Definir variables aleatorias y distribuciones [Explicar]
6. Calcular esperanza y varianza [Aplicar]
7. Describir distribuciones de probabilidad comunes [Explicar]
8. Aplicar distribuciones comunes para modelar fenómenos del mundo real [Aplicar]
9. Explicar distribuciones conjuntas y dependencia [Explicar]
10. Calcular covarianza y correlación [Aplicar]
11. Analizar las implicaciones de los teoremas límite [Evaluar]
12. Modelar sistemas simples usando cadenas de Markov [Crear]

3.8.4. MSF/Estadística (Core Tier1: 10 hrs, Core Tier2: 30 hrs)

Temas:
Core

- Estadística descriptiva: medidas de centro y dispersión
- Visualización de datos: histogramas, diagramas de caja, diagramas de dispersión
- Distribuciones muestrales: variabilidad muestral, teorema del límite central
- Estimación: estimaciones puntuales, intervalos de confianza
- Prueba de hipótesis: hipótesis nula y alternativa, valores p
- Regresión: regresión lineal simple y múltiple
- Análisis de varianza (ANOVA)
- Métodos no paramétricos
- Diseño experimental: aleatorización, control, enmascaramiento

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir medidas de estadística descriptiva [Explicar]
2. Calcular estadísticas descriptivas para un conjunto de datos [Aplicar]
3. Explicar técnicas comunes de visualización de datos [Explicar]
4. Crear visualizaciones apropiadas para datos dados [Aplicar]
5. Definir distribuciones muestrales y variabilidad [Explicar]
6. Aplicar el teorema del límite central para aproximar distribuciones [Aplicar]
7. Definir estimaciones puntuales e intervalos de confianza [Explicar]
8. Calcular intervalos de confianza para medias y proporciones [Aplicar]
9. Definir conceptos de prueba de hipótesis [Explicar]

10. Realizar una prueba de hipótesis simple e interpretar valores p [Aplicar]
11. Explicar regresión lineal simple y múltiple [Explicar]
12. Ajustar un modelo de regresión e interpretar coeficientes [Aplicar]
13. Analizar varianza usando ANOVA [Evaluar]
14. Diseñar un experimento simple con controles apropiados [Crear]

3.8.5. MSF/álgebra Lineal (Core Tier1: 5 hrs, Core Tier2: 35 hrs)

Temas:
Core

- Vectores: operaciones, producto punto, producto cruz
- Matrices: operaciones, inversas, determinantes
- Sistemas de ecuaciones lineales: eliminación gaussiana, forma matricial
- Espacios vectoriales: subespacios, base, dimensión
- Transformaciones lineales: representación matricial, núcleo, imagen
- Valores propios y vectores propios: diagonalización
- Ortogonalidad: bases ortonormales, Gram-Schmidt
- Descomposición en valores singulares (SVD)
- Aplicaciones: mínimos cuadrados, gráficos por computadora, aprendizaje automático

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir operaciones y propiedades de vectores [Explicar]
2. Calcular operaciones y productos vectoriales [Aplicar]
3. Definir operaciones y propiedades matriciales [Explicar]
4. Calcular operaciones matriciales, inversas y determinantes [Aplicar]
5. Explicar sistemas de ecuaciones lineales y métodos de solución [Explicar]
6. Resolver sistemas de ecuaciones lineales usando eliminación [Aplicar]
7. Definir espacios vectoriales, subespacios, base y dimensión [Explicar]
8. Determinar si un conjunto de vectores forma una base [Aplicar]
9. Definir transformaciones lineales y sus representaciones [Explicar]
10. Encontrar el núcleo y la imagen de una transformación lineal [Aplicar]
11. Explicar valores propios y vectores propios [Explicar]
12. Diagonalizar una matriz cuando sea posible [Aplicar]
13. Analizar ortogonalidad y bases ortonormales [Evaluar]
14. Aplicar descomposición en valores singulares a una matriz [Crear]

3.8.6. MSF/Cálculo (Core Tier2: 40 hrs)

Temas:
Core

- Límites y continuidad
- Derivadas: definición, reglas, aplicaciones
- Integrales: definidas e indefinidas, teorema fundamental
- Series: convergencia, series de Taylor
- Cálculo multivariable: derivadas parciales, gradientes
- Optimización: extremos locales y globales, multiplicadores de Lagrange
- Ecuaciones diferenciales: técnicas básicas de solución
- Métodos numéricos: integración, diferenciación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir límites y continuidad [Explicar]
2. Calcular límites de funciones [Aplicar]
3. Definir derivadas y reglas de diferenciación [Explicar]
4. Calcular derivadas de funciones [Aplicar]
5. Definir integrales y el teorema fundamental del cálculo [Explicar]
6. Calcular integrales definidas e indefinidas [Aplicar]

7. Definir series y convergencia [Explicar]
8. Calcular expansiones en series de Taylor [Aplicar]
9. Explicar derivadas parciales y gradientes [Explicar]
10. Calcular derivadas parciales y gradientes [Aplicar]
11. Analizar problemas de optimización usando cálculo [Evaluar]
12. Resolver ecuaciones diferenciales simples [Crear]

3.9. Redes y Comunicaciones (NC)

Las redes y comunicaciones juegan un papel central en los sistemas informáticos interconectados que están transformando la vida diaria de miles de millones de personas. Internet público proporciona conectividad para aplicaciones en red que sirven a un número cada vez mayor de individuos y organizaciones en todo el mundo. Complementando al sector público, las principales redes propietarias aprovechan su alcance global para apoyar la computación, el almacenamiento y la entrega de contenido distribuidos rentables. Los avances en las redes satelitales amplían la conectividad a las zonas rurales. La comunicación dispositivo a dispositivo subyace al emergente Internet de las Cosas.

Esta área de conocimiento trata conceptos clave en redes y comunicaciones, así como sus instanciaciones representativas en Internet y otras redes informáticas. Además de los principios básicos de conmutación y estratificación, el área en su núcleo proporciona conocimiento sobre denominación, direccionamiento, fiabilidad, control de errores, control de flujo, control de congestión, jerarquía de dominios, enrutamiento, reenvío, modulación, codificación, entramado y control de acceso. El área también cubre unidades de conocimiento en seguridad de red y movilidad, como amenazas de seguridad, contramedidas, comunicación dispositivo a dispositivo y redes inalámbricas multiescalón. Además de los principios fundamentales, el área incluye su realización específica en Internet, así como habilidades prácticas en la implementación de conceptos de redes y comunicaciones. Finalmente, el área comprende temas emergentes como la virtualización de red y las redes cuánticas.

Como resultado de aprendizaje principal, los estudiantes desarrollan una comprensión exhaustiva del papel y funcionamiento de las redes y comunicaciones en sistemas informáticos en red. Aprenden cómo la estructura de red y los protocolos de comunicación afectan el comportamiento de las aplicaciones distribuidas. El área puede usarse para educar no solo los principios clave sino también sus instanciaciones específicas en Internet y equipar al estudiante con habilidades prácticas de implementación. Si bien las tecnologías de sistemas informáticos, redes y comunicaciones avanzan a un ritmo acelerado, el conocimiento fundamental adquirido permite al estudiante aplicar fácilmente los conceptos en nuevos entornos tecnológicos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.9.1 Fundamentos de Redes y Comunicaciones (Pág. 97)	3	
⭐ 3.9.2 Aplicaciones en Red (Pág. 98)	4	
⭐ 3.9.3 Soporte de Fiabilidad (Pág. 98)		6
⭐ 3.9.4 Enrutamiento y Reenvío (Pág. 99)		4
⭐ 3.9.5 Comunicación de un solo salto (Pág. 99)		3
⭐ 3.9.6 Seguridad de Red (Pág. 99)		3
⭐ 3.9.7 Soporte de Movilidad (Pág. 100)		4
⭐ 3.9.8 Temas Emergentes (Pág. 100)		4

Cuadro 3.9: Lista de KUs del área de Redes y Comunicaciones.

3.9.1. NC/Fundamentos de Redes y Comunicaciones (Core Tier1: 3 hrs)

Temas:
Core

- Importancia de las redes en la informática contemporánea, y desafíos asociados. Ver también: Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad

- Organización de Internet (por ejemplo, usuarios, Proveedores de Servicios de Internet, sistemas autónomos, proveedores de contenido, redes de entrega de contenido)
- Técnicas de conmutación (por ejemplo, circuito y paquete)
- Capas y sus roles (aplicación, transporte, red, enlace de datos y física)
- Principios de estratificación (por ejemplo, encapsulación y modelo de reloj de arena). Ver también: Sistemas Digitales y Lógica, Procesamiento Secuencial y Paralelo, Máquinas de Estado

- Elementos de red (por ejemplo, routers, switches, hubs, puntos de acceso y hosts)
- Conceptos básicos de colas (por ejemplo, relación con latencia, congestión, niveles de servicio, etc.)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Articular la organización de Internet [Articular]
2. Listar y definir la terminología de red apropiada [Listar/Enumerar]
3. Describir la estructura en capas de una arquitectura de red típica [Describir]
4. Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Analizar]

3.9.2. NC/Aplicaciones en Red (Core Tier1: 4 hrs)

Temas:
Core

- Esquemas de denominación y direccionamiento (por ejemplo, DNS e Identificadores Uniformes de Recursos)
- Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación
- Diversidad de demandas de aplicaciones en red (por ejemplo, latencia, ancho de banda y tolerancia a pérdidas). Ver también: Comunicación, Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social, Equidad Digital y Accesibilidad, Impacto Social e IA
- Cobertura de protocolos de capa de aplicación (por ejemplo, HTTP)
- Interacciones con APIs de TCP, UDP y Socket. Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir los principios de denominación, direccionamiento y localización de recursos [Definir]
2. Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar]
3. Describir los detalles de un protocolo de capa de aplicación [Describir]
4. Implementar una aplicación simple cliente-servidor basada en sockets [Implementar]

3.9.3. NC/Soporte de Fiabilidad (Core Tier2: 6 hrs)

Temas:
Core

- Entrega no fiable (por ejemplo, UDP)
- Principios de fiabilidad (por ejemplo, entrega sin pérdidas, duplicación o fuera de orden). Ver también: Confiabilidad del Sistema
- Control de errores (por ejemplo, retransmisión, corrección de errores)
- Control de flujo (por ejemplo, parar y esperar, basado en ventana)
- Control de congestión (por ejemplo, notificación implícita y explícita de congestión)
- TCP y problemas de rendimiento (por ejemplo, Tahoe, Reno, Vegas, Cubic)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir el funcionamiento de protocolos de entrega fiable [Describir]
2. Listar los factores que afectan el rendimiento de los protocolos de entrega fiable [Listar/Enumerar]
3. Describir algunos problemas de diseño de fiabilidad de TCP [Describir]

4. Diseñar e implementar un protocolo fiable simple [Diseñar]

3.9.4. NC/Enrutamiento y Reenvío (Core Tier2: 4 hrs)

Temas:
Core

- Paradigmas y jerarquía de enrutamiento (por ejemplo, intra/inter dominio, centralizado y descentralizado, enrutamiento de origen, circuitos virtuales, QoS)
- Métodos de reenvío (por ejemplo, tablas de reenvío y algoritmos de coincidencia)
- IP y problemas de escalabilidad (por ejemplo, NAT, CIDR, BGP, diferentes versiones de IP)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir varios paradigmas y jerarquías de enrutamiento [Describir]
2. Describir cómo se reenvían los paquetes en una red IP [Describir]
3. Describir cómo Internet aborda los desafíos de escalabilidad [Describir]

3.9.5. NC/Comunicación de un solo salto (Core Tier2: 3 hrs)

Temas:
Core

- Introducción a modulación, ancho de banda y medios de comunicación
- Codificación y Entramado
- Control de Acceso al Medio (MAC) (por ejemplo, acceso aleatorio y acceso programado)
- Ethernet y WiFi
- Conmutación (por ejemplo, árboles de expansión, VLANs).
- Topologías de Red de área Local (por ejemplo, centro de datos, redes de campus).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir algunos aspectos básicos de modulación, ancho de banda y medios de comunicación [Describir]
2. Describir en detalle un protocolo MAC [Describir]
3. Demostrar comprensión de las compensaciones (trade-offs) de soluciones de codificación y entramado [Demostrar]
4. Describir detalles de la implementación de Ethernet [Describir]
5. Describir cómo funciona la conmutación [Describir]
6. Describir un tipo de topología de LAN [Describir]

3.9.6. NC/Seguridad de Red (Core Tier2: 3 hrs)

Temas:
Core

- Introducción general sobre seguridad (Amenazas, vulnerabilidades y contramedidas). Ver también: Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad, Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad
- Amenazas específicas de red y tipos de ataque (por ejemplo, denegación de servicio, suplantación, rastreo y redireccionamiento de tráfico, atacante en el medio, ataques de integridad de mensajes, ataques de enrutamiento, ransomware y análisis de tráfico). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad
- Contramedidas: 1. Criptografía (por ejemplo, SSL, TLS, simétrica/asimétrica) 2. Arquitecturas para redes seguras (por ejemplo, canales seguros, protocolos de enrutamiento seguro, DNS seguro, VPNs, DMZ, Zero Trust Network Access, hiper seguridad de red, protocolos de comunicación

anónima, aislamiento) 3. Monitoreo de red, detección de intrusiones, firewalls, protección contra suplantación y DoS, honeypots, trazabilidad, BGP Sec, RPKI

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir algunos de los modelos de amenaza de seguridad de red [Describir]
2. Describir contramedidas específicas basadas en red [Describir]
3. Analizar varios aspectos de seguridad de red a partir de un estudio de caso [Analizar]

3.9.7. NC/Soporte de Movilidad (Core Tier2: 4 hrs)

Temas:
Core

- Principios de comunicación celular (por ejemplo, 4G, 5G)
- Principios de LANs inalámbricas (principalmente 802.11)
- Comunicación dispositivo a dispositivo (por ejemplo, comunicación IoT)
- Redes inalámbricas multiescalón (por ejemplo, redes ad hoc, oportunistas, tolerantes a retrasos)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir algunos aspectos de comunicación celular como el registro [Describir]
2. Describir cómo 802.11 soporta usuarios móviles [Describir]
3. Describir usos prácticos de comunicación dispositivo a dispositivo, así como multiescalón [Describir]
4. Describir un tipo de red móvil como ad hoc [Describir]

3.9.8. NC/Temas Emergentes (Core Tier2: 4 hrs)

Temas:
Core

- Middleboxes (por ejemplo, avances en el uso de IA, redes basadas en intención, filtrado, inspección profunda de paquetes, balanceo de carga, NAT, CDN)
- Virtualización de Red (por ejemplo, SDN, Redes de Centros de Datos)
- Redes Cuánticas (por ejemplo, Introducción al dominio, teleportación, seguridad, Internet Cuántica)
- Satélite, mmWave, Luz Visible

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir el valor de los avances en middleboxes en redes [Describir]
2. Describir la importancia de las Redes Definidas por Software [Describir]
3. Describir algunos de los valores añadidos logrados al usar Redes Cuánticas [Describir]

3.10. Sistemas Operativos (OS)

El sistema operativo es una colección de servicios necesarios para interconectar de manera segura el hardware con las aplicaciones. Los temas centrales se centran en los mecanismos y políticas necesarios para virtualizar cómputo, memoria y entrada/salida (I/O). Los temas generales que se reutilizan en muchos niveles de los sistemas informáticos están bien ilustrados en los sistemas operativos (por ejemplo, sondeo vs interrupciones, caché, flexibilidad vs costos, enfoques de planificación de procesos, reemplazo de páginas, etc.). El área de conocimiento de Sistemas Operativos contiene los conceptos subyacentes clave para otras áreas de conocimiento: límites de confianza, concurrencia, persistencia y extensibilidad segura.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.10.1 Rol y Propósito de los Sistemas Operativos (Pág. 101)	2	
⭐ 3.10.2 Principios del Sistema Operativo (Pág. 102)	2	
⭐ 3.10.3 Concurrencia (Pág. 102)	2	1
⭐ 3.10.4 Protección y Seguridad (Pág. 103)	2	1
⭐ 3.10.5 Planificación (Pág. 104)		2
⭐ 3.10.6 Modelo de Proceso (Pág. 104)		2
⭐ 3.10.7 Gestión de Memoria (Pág. 105)		1
⭐ 3.10.8 Gestión de Dispositivos (Pág. 105)		1
⭐ 3.10.9 API de Sistemas de Archivos e Implementación (Pág. 106)		2
⭐ 3.10.10 Sistemas de Archivos Avanzados (Pág. 106)		1
⭐ 3.10.11 Virtualización (Pág. 107)		1
⭐ 3.10.12 Sistemas en Tiempo Real y Embebidos (Pág. 108)		1
⭐ 3.10.13 Tolerancia a fallos (Pág. 108)		1
3.10.14 Sociedad, ética y Profesión (Pág. 109)	Electivo	

Cuadro 3.10: Lista de KUs del área de Sistemas Operativos.

3.10.1. OS/Rol y Propósito de los Sistemas Operativos (Core Tier1: 2 hrs)

Temas:
Core

- Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación.
- Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos)
- Funciones extendidas y/o especializadas del sistema operativo (por ejemplo, sistemas embebidos, tipos de servidor como archivos, web, multimedia, gestores de arranque y seguridad de arranque)
- Problemas de diseño (por ejemplo, eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad, energía, seguridad, compensaciones entre verificación de errores y rendimiento, flexibilidad y rendimiento, y seguridad y rendimiento). Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad
- Influencias de seguridad, redes, multimedia, computación paralela y distribuida
- Preocupación general de seguridad/protección: Omitir considerar la seguridad en cada capa crea una oportunidad para acceder de manera inapropiada a los recursos. 1. Se puede lograr acceso no autorizado a archivos en una unidad no cifrada moviendo el medio a otra computadora. 2. La seguridad impuesta por los sistemas operativos puede ser derrotada infiltrando la capa de arranque antes de que se cargue el sistema operativo. 3. El aislamiento de procesos puede ser subvertido por una verificación de autorización inadecuada en los límites de la API. 4. Las vulnerabilidades en el firmware del sistema pueden proporcionar vectores de ataque que eviten por completo el sistema operativo. 5. El aislamiento inadecuado de la memoria, cómputo y hardware de la máquina virtual puede exponer el sistema anfitrión a ataques desde sistemas invitados. 6. El sistema operativo puede necesitar mitigar la explotación de vulnerabilidades de hardware y firmware, lo que lleva a posibles reducciones de rendimiento (por ejemplo, mitigaciones de Spectre y Meltdown).
- Exposición de funciones del sistema operativo en shells y programación de sistemas. Ver también: Scripting de Shell

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar]
2. Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar]
3. Comprender las funciones de un sistema operativo contemporáneo con respecto a conveniencia,

eficiencia y capacidad de evolucionar [Explicar]
4. Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]

3.10.2. OS/Principios del Sistema Operativo (Core Tier1: 2 hrs)

Temas:
Core

- Diseño y enfoques de software del sistema operativo (por ejemplo, monolítico, en capas, modular, micro-kernel, unikernel)
- Abstracciones, procesos y recursos
- Concepto de llamadas al sistema y enlaces a interfaces de programas de aplicación (por ejemplo, Win32, Java, Posix). Ver también: Organización de Máquina a Nivel de Ensamblador
- La evolución del vínculo entre la arquitectura del hardware y las funciones del sistema operativo
- Protección de recursos significa proteger algunas instrucciones/funciones de la máquina. Ver también: Organización de Máquina a Nivel de Ensamblador 1. Las aplicaciones no pueden acceder arbitrariamente a ubicaciones de memoria o direcciones de dispositivos de almacenamiento de archivos. 2. Protección de coprocesadores y dispositivos de red
- Aprovechamiento de interrupciones desde el nivel de hardware: rutinas de servicio e implementaciones. Ver también: Organización de Máquina a Nivel de Ensamblador 1. Interrupciones de temporizador para implementar segmentos de tiempo 2. Interrupciones de E/S para poner hilos bloqueados en espera sin sondeo
- Concepto de estado usuario/sistema y protección, transición al modo kernel usando llamadas al sistema. Ver también: Organización de Máquina a Nivel de Ensamblador
- Mecanismo para invocar llamadas al sistema, el cambio de modo y contexto correspondiente y retorno desde interrupción. Ver también: Organización de Máquina a Nivel de Ensamblador
- Costos de rendimiento de cambios de contexto y vaciados de caché asociados al realizar cambios de proceso en entornos mitigados para Spectre.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comprender cómo la aplicación de enfoques de diseño de software al diseño/implementación de sistemas operativos (por ejemplo, en capas, modular, etc.) afecta la robustez y mantenibilidad de un sistema operativo [Explicar]
2. Categorizar llamadas al sistema por propósito [Categorizar]
3. Comprender la dinámica de invocar una llamada al sistema (por ejemplo, pasar parámetros, cambio de modo) [Explicar]
4. Evaluar si una función puede implementarse en la capa de aplicación o solo puede realizarse mediante llamadas al sistema [Evaluar]
5. Aplicar técnicas de SO para aislamiento, protección y rendimiento a través de las funciones del SO (por ejemplo, similitudes de inanición en planificación de procesos, planificación de solicitudes de disco, semáforos, etc.) y más allá [Aplicar]
6. Comprender cómo la separación en modo kernel y usuario afecta la seguridad y el rendimiento [Explicar]
7. Comprender las ventajas y desventajas de usar procesamiento de interrupciones para habilitar la multiprogramación [Explicar]
8. Analizar vectores potenciales de ataque a través de los sistemas operativos y las características de seguridad diseñadas para proteger contra ellos [Analizar]

3.10.3. OS/Concurrencia (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Abstracción de hilos relativa a la concurrencia
- Condiciones de carrera, regiones críticas (rol de interrupciones, si es necesario). Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Progra-

mas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades
- Interbloqueos (deadlocks) e inanición. Ver también: Coordinación
- Problemas de multiprocesador (bloqueos de giro (spin-locks), reentrada).
- Concurrencia multiproceso vs multihilo
- Creación de hilos, estados, estructuras. Ver también: Sistemas Digitales y Lógica, Procesamiento Secuencial y Paralelo, Máquinas de Estado
- APIs de hilos
- Interbloqueos e inanición (condiciones necesarias/mitigaciones). Ver también: Coordinación
- Implementar código seguro para hilos (semáforos, bloqueos mutex, variables de condición). Ver también: Rendimiento y Eficiencia Energética, Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento, Evaluación
- Condiciones de carrera en memoria compartida. Ver también: Coordinación

Non Core
- Gestionar acceso atómico a objetos del SO (por ejemplo, gran bloqueo del kernel vs muchos bloqueos pequeños vs estructuras de datos sin bloqueo como listas)

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Comprender las ventajas y desventajas de la concurrencia como funciones inseparables dentro del marco del sistema operativo [Explicar]
2. Comprender cómo la implementación a nivel de arquitectura resulta en problemas de concurrencia incluyendo condiciones de carrera [Explicar]
3. Comprender problemas de concurrencia en sistemas multiprocesador [Explicar]
4. Comprender el rango de mecanismos que pueden emplearse a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Explicar]
5. Comprender técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir cómo puede implementarse un semáforo usando primitivas del SO) incluyendo control de intraconcurrencia y uso de atómicos de hardware [Explicar]
6. Analizar con precisión código para identificar condiciones de carrera y soluciones apropiadas para abordarlas [Analizar]

3.10.4. OS/Protección y Seguridad (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core
- Descripción general de mecanismos de seguridad del sistema operativo. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad
- Ataques y antagonismo (planificación, etc.). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad
- Revisión de vulnerabilidades principales en sistemas operativos reales. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad
- Estrategias de mitigación del sistema operativo como copias de seguridad. Ver también: Confiabilidad del Sistema
- Separación política/mecanismo. Ver también: Gestión de Riesgos y Datos, Gobierno, Gestión y Política de Seguridad
- Métodos y dispositivos de seguridad. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad 1. Anillos de protección (historia desde Multics hasta x86 virtualizado) 2. Anillos -1 y -2 de x86_64 (hipervisor y ME/PSP)
- Protección, control de acceso y autenticación. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Protocolos, Aplicaciones y Criptografía Avanzada, Algoritmos Criptográficos, Fundamentos de Criptografía

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Comprender el requisito de mecanismos de protección y seguridad en sistemas operativos [Explicar]
2. Listar y describir los vectores de ataque que aprovechan vulnerabilidades del SO [Listar/Enumerar]

3. Comprender los mecanismos disponibles en un SO para controlar el acceso a recursos [Explicar]
4. Resumir las características y limitaciones de un sistema operativo que impactan la protección y seguridad [Resumir]

3.10.5. OS/Planificación (Core Tier2: 2 hrs)

Temas:
Core

- Planificación con expropiación y sin expropiación
- Planificadores y políticas (por ejemplo, primero en llegar primero en ser servido, trabajo más corto primero, prioridad, round robin, multinivel). Ver también: Gestión de Recursos
- Conceptos de planificación de multiprocesador simétrico (SMP) y coherencia de caché Ver también: Jerarquía de Memoria
- Temporizadores (por ejemplo, construir muchos temporizadores a partir de temporizadores de hardware finitos). Ver también: Organización de Máquina a Nivel de Ensamblador
- Justicia e inanición

Non Core

- Subtemas de sistemas operativos como planificación consciente de energía y planificación en tiempo real. Ver también: Rendimiento y Eficiencia Energética, Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores, Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo
- Planificación cooperativa, como futexes de Linux y planificación en espacio de usuario.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar y contrastar los algoritmos comunes utilizados para planificación con y sin expropiación de tareas en sistemas operativos, como prioridad, comparación de rendimiento y esquemas de participación justa [Comparar]
2. Explicar las relaciones entre algoritmos de planificación y dominios de aplicación [Explicar]
3. Explicar las distinciones entre tipos de planificadores de procesador como corto plazo, mediano plazo, largo plazo y E/S [Explicar]
4. Evaluar un problema o solución para determinar la idoneidad para multiprocesamiento asimétrico y/o simétrico [Evaluar]
5. Evaluar un problema o solución para determinar la idoneidad como proceso vs hilos [Evaluar]
6. Listar algunos contextos que se benefician de la planificación con expropiación y con plazo límite [Listar/Enumerar]

NonCore:

7. Explicar las formas en que la lógica incorporada en los algoritmos de planificación es aplicable a otros mecanismos de sistemas operativos, como primero en llegar primero en ser servido o prioridad para E/S de disco, planificación de red, planificación de proyectos y problemas más allá de la computación [Explicar]

3.10.6. OS/Modelo de Proceso (Core Tier2: 2 hrs)

Temas:
Core

- Procesos e hilos relativos a virtualización de memoria protegida, estado del proceso, aislamiento de memoria, etc.
- Huella de memoria/segmentación (por ejemplo, pila, montón, etc.). Ver también: Organización de Máquina a Nivel de Ensamblador
- Creación y carga de ejecutables, bibliotecas compartidas y enlace dinámico. Ver también: Traducción y Ejecución de Lenguajes
- Despacho y cambio de contexto. Ver también: Organización de Máquina a Nivel de Ensamblador
- Comunicación entre procesos (por ejemplo, memoria compartida, paso de mensajes, señales, variables de entorno). Ver también: Comunicación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comprender cómo los procesos e hilos usan características de concurrencia para virtualizar el control [Explicar]
2. Comprender las razones para usar interrupciones, despacho y cambio de contexto para apoyar concurrencia y virtualización en un sistema operativo [Explicar]
3. Comprender los diferentes estados por los que puede pasar una tarea, y las estructuras de datos necesarias para apoyar la gestión de muchas tareas [Explicar]
4. Comprender las diferentes formas de asignar memoria a tareas, citando los méritos relativos de cada una [Explicar]
5. Aplicar el mecanismo de comunicación entre procesos apropiado para un propósito específico en un artefacto de software programado [Aplicar]

3.10.7. OS/Gestión de Memoria (Core Tier2: 1 hr)

Temas:
Core

- Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas
- Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas
- Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas y marcos
- Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad 1. Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes)
- Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento)
- Mecanismos y conceptos de seguridad en gestión de memoria incluyendo sandboxing, protección, aislamiento y vectores relevantes de ataque. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad

Non Core

- Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar]
2. Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir]
3. Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar]
4. Describir la razón y el uso de la memoria caché (rendimiento y proximidad, cómo las cachés complican el aislamiento y la abstracción de máquina virtual) [Describir]
5. Codificar/Desarrollar programas eficientes que consideren los efectos del reemplazo de páginas y asignación de marcos en el rendimiento de un proceso y el sistema en el que se ejecuta [Crear]

NonCore:

6. Explicar cómo se utiliza el hardware para una virtualización eficiente [Explicar]

3.10.8. OS/Gestión de Dispositivos (Core Tier2: 1 hr)

Temas:
Core

- Estrategias de almacenamiento en búfer. Ver también: Interconexión y Comunicación

- Acceso Directo a Memoria (DMA) y E/S sondeada, E/S mapeada en memoria. Ver también: Interconexión y Comunicación 1. Protocolos de comunicación DMA (por ejemplo, búfers en anillo, etc.)
- Histórico y contextual - Gestión de dispositivos de almacenamiento persistente. Por ejemplo: Magnético, Dispositivo de Estado Sólido (SSD). Ver también: Fundamentos e Historia Temprana de la Informática, Historia Moderna: PC, Internet e IA

Non Core

- Abstracciones de interfaz de dispositivo, capa de abstracción de hardware
- Propósito del controlador de dispositivo, abstracción, implementación y desafíos de prueba
- Tolerancia a fallos de alto nivel en comunicación de dispositivos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar la implementación de control de dispositivos a nivel de arquitectura y vincular mecanismos y políticas relevantes del sistema operativo (por ejemplo, estrategias de almacenamiento en búfer, acceso directo a memoria) [Explicar]
2. Explicar las capas de gestión de dispositivos del SO y la arquitectura (por ejemplo, controlador de dispositivo, controlador de dispositivo, abstracción de dispositivo) [Explicar]
3. Explicar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Explicar]
4. Explicar el almacenamiento en búfer de datos de E/S y describir estrategias para implementarlo [Explicar]
5. Describir las ventajas y desventajas del acceso directo a memoria y discutir las circunstancias en las que su uso está justificado [Describir]

NonCore:

6. Describir la complejidad y mejores prácticas para la creación de controladores de dispositivos [Describir]

3.10.9. OS/API de Sistemas de Archivos e Implementación (Core Tier2: 2 hrs)

Temas:

Core

- Concepto de archivo incluyendo datos, metadatos, operaciones y modo de acceso
- Montaje de sistema de archivos
- Control de acceso a archivos
- Compartición de archivos
- Métodos básicos de asignación de archivos, incluida la tabla de asignación enlazada
- Estructuras del sistema de archivos que comprenden asignación de archivos incluyendo varias estructuras de directorio y métodos para identificar archivos de manera única (por ejemplo, nombre, identificador o ubicación de almacenamiento de metadatos)
- Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) impacto en rendimiento y flexibilidad (es decir, fragmentación interna y externa y compactación)
- Gestión de espacio libre como usar tablas de bits vs enlazado
- Implementación de directorios para segmentar y rastrear ubicación de archivos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar las elecciones que deben tomarse en el diseño de sistemas de archivos [Explicar]
2. Evaluar diferentes enfoques para organización de archivos, reconociendo las fortalezas y debilidades de cada uno [Evaluar]
3. Aplicar constructores de software apropiadamente dado el conocimiento de la implementación del sistema de archivos [Aplicar]

3.10.10. OS/Sistemas de Archivos Avanzados (Core Tier2: 1 hr)

Temas:

Core

- Sistemas de archivos: particionamiento, montar/desmontar, sistemas de archivos virtuales
- Técnicas de implementación en profundidad
- Archivos mapeados en memoria. Ver también: Interconexión y Comunicación
- Sistemas de archivos de propósito especial
- Denominación, búsqueda, acceso, copias de seguridad
- Sistemas de archivos con registro (journaling) y estructurados en registro (log-structured). Ver también: Confiabilidad del Sistema

Non Core

- Sistemas de archivos distribuidos
- Sistemas de archivos cifrados
- Tolerancia a fallos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar cómo los desarrollos de hardware han llevado a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Explicar]
2. Mapear abstracciones de archivo a una lista de dispositivos e interfaces relevantes [Mapear/Mapa conceptual]
3. Identificar y categorizar diferentes tipos de montaje [Analizar]
4. Explicar requisitos específicos de sistemas de archivos y las características de sistemas de archivos especializados que los satisfacen [Explicar]
5. Explicar el uso de registro y cómo los sistemas de archivos estructurados en registro mejoran la tolerancia a fallos [Explicar]

NonCore:

6. Explicar propósito y complejidad de los sistemas de archivos distribuidos [Explicar]
7. Listar ejemplos de protocolos de sistemas de archivos distribuidos [Listar/Enumerar]
8. Explicar mecanismos en sistemas de archivos para mejorar la tolerancia a fallos [Explicar]

3.10.11. OS/Virtualización (Core Tier2: 1 hr)

Temas:

Core

- Usar virtualización y aislamiento para lograr protección y rendimiento predecible. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Paginación avanzada y memoria virtual. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Sistemas de archivos virtuales y dispositivos virtuales.
- Contenedores y su comparación con máquinas virtuales.
- Trashing (por ejemplo, requisitos de Popek y Goldberg para sistemas recursivamente virtualizables).

Non Core

- Tipos de virtualización (incluyendo hardware/software, SO, servidor, servicio, red). Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Virtualización portátil; emulación vs aislamiento. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Costo de virtualización. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento
- Máquinas virtuales y escapes de contenedores, peligros desde una perspectiva de seguridad. Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad
- Hipervisores - extensiones de máquina virtual de hardware, anfitriones con soporte de kernel, QEMU KVM

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar cómo la arquitectura de hardware proporciona soporte y eficiencias para la virtualización [Explicar]
2. Explicar la diferencia entre emulación y aislamiento [Explicar]

3. Evaluar las compensaciones de la virtualización [Evaluar]

NonCore:

4. Explicar hipervisores y la necesidad de ellos junto con diferentes tipos de hipervisores [Explicar]

3.10.12. OS/Sistemas en Tiempo Real y Embebidos (Core Tier2: 1 hr)

Temas:

Core

- Planificación de procesos y tareas.
- Plazos límite y problemas de tiempo real. Ver también: Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores
- Baja latencia vs "tiempo real suave" vs "tiempo real duro". Ver también: Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores, Programación Dirigida por Eventos y Reactiva

Non Core

- Requisitos de gestión de memoria/disco en un entorno de tiempo real.
- Fallas, riesgos y recuperación.
- Preocupaciones especiales en sistemas de tiempo real (seguridad).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar qué hace que un sistema sea un sistema de tiempo real [Explicar]
2. Explicar la latencia y sus fuentes en sistemas de software y sus características [Explicar]
3. Explicar las preocupaciones especiales que presentan los sistemas de tiempo real, incluido el riesgo, y cómo se abordan estas preocupaciones [Explicar]

NonCore:

4. Explicar características y mecanismos específicos de sistemas operativos de tiempo real [Explicar]

3.10.13. OS/Tolerancia a fallos (Core Tier2: 1 hr)

Temas:

Core

- Sistemas confiables y disponibles. Ver también: Confiabilidad del Sistema
- Enfoques de software y hardware para abordar la tolerancia (RAID). Ver también: Confiabilidad del Sistema

Non Core

- Redundancia espacial y temporal. Ver también: Confiabilidad del Sistema
- Métodos utilizados para implementar tolerancia a fallos. Ver también: Confiabilidad del Sistema
- Mecanismos de identificación y corrección de errores, sumas de comprobación (checksums) de memoria volátil en RAM. Ver también: Jerarquía de Memoria
- Verificación de consistencia y recuperación del sistema de archivos.
- Sistemas de archivos con registro y estructurados en registro. Ver también: Confiabilidad del Sistema
- Casos de uso para tolerancia a fallos (bases de datos, críticos para la seguridad). Ver también: Confiabilidad del Sistema
- Ejemplos de mecanismos del SO para detección, recuperación, reinicio para implementar tolerancia a fallos, uso de estas técnicas para los servicios propios del SO. Ver también: Confiabilidad del Sistema

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar cómo los sistemas operativos pueden facilitar la tolerancia a fallos, confiabilidad y disponibilidad [Explicar]
2. Explicar el rango de métodos para implementar tolerancia a fallos en un sistema operativo [Explicar]

3. Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Explicar]
4. Explicar las compensaciones de rendimiento y flexibilidad que impactan el uso de tolerancia a fallos [Explicar]

NonCore:

5. Describir en detalle los problemas y mecanismos de tolerancia a fallos del sistema operativo [Describir]

3.10.14. OS/Sociedad, ética y Profesión

Temas:

Core

- Código abierto en sistemas operativos. Ver también: Open Source, Piratería y DRM, Derechos de PI Digital y Derechos de Autor, Fundamentos de la Propiedad Intelectual, Licencias, Plagio y Autoría 1. Identificación de vulnerabilidades en kernels de código abierto, 2. Sistemas operativos invitados de código abierto, 3. Sistemas operativos anfitriones de código abierto, y 4. Cambios en monetización (actualizaciones pagas vs gratuitas).
- Problemas de fin de vida con la retirada de sistemas operativos. Concepto ejemplo: Implicaciones de privacidad del uso de sistemas operativos/entornos operativos propietarios, incluyendo telemetría, escaneo automatizado de datos personales, publicidad incorporada e integración automática en la nube.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar ventajas y desventajas de encontrar y abordar errores en kernels de código abierto [Explicar]
2. Contextualizar la historia e impacto positivo y negativo de Linux como producto de código abierto [Contextualizar]
3. Listar complicaciones con la dependencia de sistemas operativos pasados de fin de vida [Listar/Enumerar]
4. Comprender diferencias en encontrar y abordar errores para varios modelos de pago de sistemas operativos [Explicar]

3.11. Computación Paralela y Distribuida (PDC)

La programación paralela y distribuida organiza, coordina y controla múltiples cálculos que ocurren al mismo tiempo en diferentes lugares. La ubicuidad del paralelismo y la distribución son consecuencias inevitables del creciente número de puertas en los procesadores, procesadores en las computadoras y computadoras en todas partes que pueden usarse para mejorar el rendimiento en comparación con los programas secuenciales, al mismo tiempo que se hace frente a la interconectividad intrínseca del mundo y la posibilidad de que algunos componentes o conexiones fallen o se comporten de manera maliciosa. La programación paralela y distribuida elimina las restricciones de la programación secuencial que requieren que los pasos computacionales ocurran en un orden serial en un solo lugar, revelando distinciones, técnicas y análisis adicionales que se aplican en cada capa de los sistemas informáticos.

En la mayoría de los usos convencionales, la programación "paralela" se centra en establecer y coordinar múltiples actividades que pueden ocurrir al mismo tiempo, la programación "distribuida" se centra en establecer y coordinar actividades que pueden ocurrir en diferentes lugares, y la programación "concurrente" se centra en las interacciones de las actividades en curso entre sí y con el entorno. Sin embargo, los tres términos pueden aplicarse en la mayoría de los contextos. El paralelismo generalmente implica alguna forma de distribución porque múltiples actividades que ocurren sin restricciones de orden secuencial suceden en múltiples lugares físicos (a menos que dependan de cambios de contexto o efectos cuánticos). Por el contrario, las acciones en diferentes lugares no necesitan tener un orden secuencial específico entre sí en ausencia de restricciones de comunicación.

Las técnicas de programación paralela, distribuida y concurrente forman el núcleo de la Computación de Alto Rendimiento (HPC), los sistemas distribuidos y, cada vez más, casi todas las aplicaciones informáticas. El área de conocimiento PDC ha evolucionado desde un conjunto diverso de temas

avanzados hasta un cuerpo central de conocimiento y práctica, que impregna casi todos los demás aspectos de la informática. El crecimiento del campo ha ocurrido de manera irregular en los diferentes subcampos de la informática, a veces con diferentes objetivos, terminología y prácticas, ocultando la superposición considerable de ideas y habilidades básicas que son el foco principal de esta área de conocimiento. Casi todos los problemas con una solución secuencial también admiten soluciones paralelas y/o distribuidas; surgen problemas y soluciones adicionales solo en el contexto de la concurrencia. Casi todos los dominios de aplicación de la computación paralela y distribuida son un área de estudio y/o ingeniería bien desarrollada, demasiado grande para enumerarla.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.11.1 Programas: Paralelismo Declarativo (Pág. 110)	1	1
⭐ 3.11.2 Programas: Inicio de Actividades (Pág. 111)	1	1
⭐ 3.11.3 Programas: Propiedades de Ejecución (Pág. 111)	1	1
⭐ 3.11.4 Programas: Distribución (Pág. 111)	1	1
⭐ 3.11.5 Programas: Mapeos de Implementación (Pág. 112)	1	1
3.11.6 Programación GPU (Pág. 112)	Electivo	
⭐ 3.11.7 Comunicación (Pág. 112)	2	4
⭐ 3.11.8 Comunicación: Propiedades y Extensiones (Pág. 113)	1	1
⭐ 3.11.9 Memoria y Consistencia (Pág. 114)	1	1
⭐ 3.11.10 Coordinación (Pág. 114)	1	3
⭐ 3.11.11 Coordinación: Sincronización y Atomicidad (Pág. 115)	1	2
⭐ 3.11.12 Coordinación: Propiedades Avanzadas (Pág. 115)	1	2
⭐ 3.11.13 Evaluación (Pág. 115)	1	3
⭐ 3.11.14 Evaluación: Métodos Formales (Pág. 116)	1	1
⭐ 3.11.15 Evaluación: Pruebas y Medición (Pág. 116)	1	1
⭐ 3.11.16 Algoritmos (Pág. 117)	1	3
⭐ 3.11.17 Algoritmos: Dominios de Aplicación (Pág. 117)	1	3
⭐ 3.11.18 Algoritmos: Dominios Algorítmicos (Pág. 118)	1	3

Cuadro 3.11: Lista de KUs del área de Computación Paralela y Distribuida.

3.11.1. PDC/Programas: Paralelismo Declarativo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Paralelismo 1. Paralelismo declarativo: Determinar qué acciones pueden, o no deben, realizarse en paralelo, a nivel de instrucciones, funciones, clausuras, acciones compuestas, sesiones, tareas y servicios es la idea principal que subyace a los algoritmos PDC; no hacerlo es la principal fuente de errores. Ver también: Algoritmos 2. Definir orden: por ejemplo, usando relaciones de "sucede antes" o grafos acíclicos dirigidos serie/paralelo que representan programas. 3. Independencia: determinar cuándo el orden no importa, en términos de conmutatividad, dependencias, precondiciones. 4. Garantizar el orden entre acciones que de otro modo serían paralelas cuando sea necesario, incluyendo bloqueos, publicación segura; e imponer comunicación - enviar un mensaje sucede antes de recibirlo; y relajarlo cuando no sea necesario.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Mostrar gráficamente (como un Grafo Acíclico Dirigido - *Directed Acyclic Graph* (DAG)) cómo paralelizar una expresión numérica compuesta; por ejemplo, $a = (b + c) * (d + e)$ [Diseñar]
2. Explicar por qué los conceptos de consistencia y tolerancia a fallos no surgen en programas puramente secuenciales [Explicar]

3.11.2. PDC/Programas: Inicio de Actividades (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Iniciar actividades 1. Las opciones que permiten que las acciones se realicen (eventualmente) en lugares van desde el cableado directo hasta scripts de configuración; también establecer comunicación y gestión de recursos; estos se expresan de manera diferente en los lenguajes y contextos, generalmente confiando en el aprovisionamiento y gestión automatizados por las plataformas. Ver también: Gestión de Recursos 2. Procedimental: Permitir que múltiples acciones comiencen en un punto de programa dado; por ejemplo, iniciar nuevos hilos, posiblemente delimitando su alcance u organizándolos en grupos jerárquicos. 3. Reactiva: Habilitar al ocurrir un evento instalando un manejador de eventos, con menos control de cuándo comienzan o terminan las acciones, y puede aplicarse incluso en monoprocesadores. 4. Dependiente: Habilitar al completarse otras; por ejemplo, secuenciando conjuntos de acciones paralelas. Ver también: Coordinación. 5. Granularidad: El costo de ejecución de los cuerpos de las acciones debe superar la sobrecarga de organizarlos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]

3.11.3. PDC/Programas: Propiedades de Ejecución (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Propiedades de Ejecución 1. Ejecución no determinista de acciones sin orden. 2. Consistencia: Garantizar el acuerdo entre partes sobre valores y predicados cuando sea necesario para evitar carreras, mantener seguridad y atomicidad, o llegar a consenso. 3. Tolerancia a fallos: Manejar fallos en las partes o la comunicación, incluyendo (Bizantino) mal comportamiento debido a partes y protocolos no confiables, cuando sea necesario para mantener el progreso o la disponibilidad. Ver también: Confiabilidad del Sistema. 4. Los compromisos son un foco de evaluación. Ver también: Evaluación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir una función que cuente eventos, como recepciones de paquetes de red, de manera eficiente [Escribir]

3.11.4. PDC/Programas: Distribución (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Distribución 1. Definir lugares, como dispositivos que ejecutan acciones, incluyendo componentes de hardware, hosts remotos; también pueden incluir dispositivos, hosts y usuarios externos no controlados. Ver también: Interconexión y Comunicación 2. Un dispositivo puede dividir el tiempo o emular múltiples acciones paralelas mediante menos procesadores mediante planificación y virtualización. Ver también: Planificación 3. Nombrar o identificar lugares (por ejemplo, IDs de dispositivo) y acciones como partes (por ejemplo, IDs de hilo). 4. Las actividades a través de lugares pueden comunicarse a través de medios. Ver también: Comunicación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir]

3.11.5. PDC/Programas: Mapeos de Implementación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Una o más de las siguientes asignaciones y mecanismos a través de sistemas en capas: 1. Paralelismo a nivel de instrucción y datos en la CPU. Ver también: Organización Funcional. 2. SIMD y paralelismo de datos heterogéneo. Ver también: Arquitecturas Heterogéneas. 3. Concurrencia planificada en multinúcleo, tareas, actores. Ver también: Planificación. 4. Clústeres, nubes; aprovisionamiento elástico. Ver también: Aspectos Comunes: Plataformas, APIs y Restricciones, Aspectos Comunes: Lenguajes de Plataforma y Patrones Web. 5. Sistemas distribuidos en red. Ver también: Aplicaciones en Red. 6. Tecnologías emergentes como la computación cuántica y la computación molecular.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las compensaciones entre diferentes estrategias de mapeo en términos de rendimiento, costo y complejidad de implementación [Explicar]

3.11.6. PDC/Programación GPU

Temas:
Non Core

- Computación de Propósito General en GPU (GPGPU) 1. Arquitectura de GPU: multiprocesadores de flujo (SMs), núcleos CUDA y modelo de ejecución por warp. 2. Modelo de ejecución SIMT (Single Instruction, Multiple Threads) y sus diferencias respecto al SIMD de CPU. 3. Interacción host-dispositivo: transferencia de datos entre memoria de CPU (host) y GPU (dispositivo) vía PCIe. 4. Casos de uso: computación científica, aprendizaje automático, procesamiento de imágenes y simulación a gran escala.
- Modelo de Programación CUDA 1. Jerarquía de ejecución CUDA: grids, bloques de hilos e hilos individuales; mapeo al hardware GPU. 2. Definición y lanzamiento de kernels; parámetros de configuración (dimensiones de grid y bloque). 3. Indexación de hilos: threadIdx, blockIdx, blockDim, gridDim; cálculo de índices globales en 1D, 2D y 3D. 4. Divergencia de warp: implicaciones de las bifurcaciones dentro de un warp en el rendimiento. 5. Primitiva de sincronización intra-bloque: __syncthreads().
- Jerarquía de Memoria CUDA 1. Memoria global: grande, alta latencia, accesible por todos los hilos; patrones de acceso coalescente para el rendimiento. 2. Memoria compartida: memoria en chip de baja latencia compartida dentro de un bloque de hilos; conflictos de banco. 3. Registros y memoria local: almacenamiento privado por hilo. 4. Memoria constante y de textura: cachés de solo lectura optimizadas para patrones de acceso específicos. 5. Memoria unificada: modelo de programación simplificado con migración automática de datos entre host y dispositivo.

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Escribir un kernel CUDA que realice una operación data-paralela (p. ej., suma de vectores, multiplicación de matrices) y configurar correctamente las dimensiones de grid y bloque [Escribir]
2. Analizar el impacto de los patrones de acceso a memoria global (acceso coalescente vs. acceso entrelazado) y el uso de memoria compartida en el rendimiento de un kernel GPU [Analizar]
3. Diseñar un algoritmo paralelo usando el modelo de programación CUDA, incluyendo la jerarquía de hilos, estrategia de asignación de memoria y transferencias de datos entre host y dispositivo [Diseñar]

3.11.7. PDC/Comunicación (Core Tier1: 2 hrs, Core Tier2: 4 hrs)

Temas:
Core

- Medios 1. Variedades: canales (paso de mensajes o E/S), memoria compartida, heterogéneos, almacenes de datos. 2. Dependencia de la disponibilidad y naturaleza del hardware subyacen-

te, conectividad y protocolos; soporte del lenguaje, emulación. Ver también: Interconexión y Comunicación.
- Canales 1. Medios de comunicación de parte a parte explícitos (generalmente nombrados). 2. APIs: Sockets, constructos arquitectónicos, basados en lenguaje y de kits de herramientas, como Message Passing Interface (MPI), y constructos en capas como Llamada a Procedimiento Remoto (RPC). Ver también: Fundamentos de Redes y Comunicaciones. 3. APIs de canales de E/S.
- Memoria 1. Arquitecturas de memoria compartida en las que las partes se comunican directamente solo con la memoria en direcciones dadas, con extensiones a memoria heterogénea que admite múltiples almacenes de memoria con transferencia de datos explícita entre ellos; por ejemplo, memoria local y compartida de GPU, Acceso Directo a Memoria (DMA). 2. Jerarquías de memoria: Múltiples capas de dominios, alcances y cachés compartidos; localidad: latencia, falso uso compartido. 3. Propiedades de consistencia: Límites de atomicidad a nivel de bits, coherencia, orden local.
- Almacenes de Datos 1. Estructuras de datos mantenidas cooperativamente que implementan mapas y TADs relacionados. 2. Variedades: Propiedad exclusiva, compartida, fragmentada, replicada, inmutable, versionada.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las similitudes y diferencias entre: (1) La parte A envía un mensaje en el canal X con contenido 1 recibido por la parte B (2) A establece la variable compartida X a 1, leída por B (3) A establece "X=1" en un mapa compartido distribuido al que accede B [Explicar]
2. Escribir un programa que distribuya diferentes segmentos de un conjunto de datos a múltiples trabajadores y recoja los resultados (por ejemplo, sumar segmentos de un arreglo) [Escribir]
3. Escribir un programa paralelo que solicite datos de múltiples sitios y los resuma usando alguna forma de reducción [Escribir]
4. Comparar el rendimiento de versiones con y sin búfer de un programa productor-consumidor [Comparar]

3.11.8. PDC/Comunicación: Propiedades y Extensiones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Una o más de las siguientes propiedades y extensiones: 1. Topologías: Unicast, Multicast, Buzones, Conmutadores; Enrutamiento a través de redes de interconexión de hardware y software. 2. Propiedades de concurrencia de medios: Orden, consistencia, idempotencia, superposición de comunicación con computación. 3. Rendimiento del medio: Latencia, ancho de banda (rendimiento), contención (congestión), capacidad de respuesta (vivacidad), confiabilidad (tasas de error y pérdida), progreso basado en protocolo (acks, tiempos de espera, mediación). 4. Propiedades de seguridad del medio: integridad, privacidad, autenticación, autorización. Ver también: Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada. 5. Formatos de datos: Serialización, validación, cifrado, compresión. 6. Políticas de canal: Puntos finales, sesiones, almacenamiento en búfer, respuesta a saturación (esperar vs. descartar), control de velocidad. 7. Multiplexación y demultiplexación de muchos dispositivos o partes de E/S relativamente lentos; técnicas basadas en finalización y basadas en planificador; APIs async-await, select y polling. 8. Formalización y análisis de comunicación por canales; por ejemplo, CSP. 9. Aplicaciones de la teoría de colas para modelar y predecir el rendimiento.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Determinar si un esquema de comunicación dado proporciona propiedades de seguridad suficientes para un uso dado [Determinar]
2. Dar un ejemplo de un escenario en el que los envíos de mensajes bloqueantes pueden causar un punto muerto (deadlock) [Crear]

3. Describir al menos una técnica de diseño para evitar fallos de vivacidad en programas que usan múltiples bloqueos [Describir]

3.11.9. PDC/Memoria y Consistencia (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Modelos de memoria, consistencia de datos y tolerancia a fallos: 1. Modelos de memoria: consistencia secuencial y de liberación/adquisición. 2. Gestión de memoria; incluyendo la reclamación de datos compartidos; conteo de referencias y alternativas. 3. Colocación y transferencia masiva de datos; reducción del tráfico de mensajes y mejora de la localidad; superposición de transferencia de datos y computación; impacto del diseño de datos como array de estructuras vs. estructura de arrays. 4. Emular memoria compartida: memoria compartida distribuida, Acceso Directo a Memoria Remota (RDMA). 5. Consistencia de almacenes de datos: Atomicidad, linealizabilidad, transaccionalidad, coherencia, orden causal, resolución de conflictos, consistencia eventual, cadenas de bloques. 6. Fallos, particionamiento y fallos parciales; votación; protocolos como Paxos y Raft. 7. Compromisos de diseño entre consistencia, disponibilidad, tolerancia a partición (fallos); imposibilidad de cumplir con todos a la vez. 8. Seguridad y confianza: Fallos bizantinos, prueba de trabajo y alternativas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Dar un ejemplo de un orden de accesos entre actividades concurrentes (por ejemplo, un programa con una carrera de datos) que no sea secuencialmente consistente [Crear]
2. Escribir un programa que ilustre el reordenamiento de acceso a memoria o de mensajes [Escribir]
3. Describir los méritos relativos del control de concurrencia optimista versus conservador bajo diferentes tasas de contención entre actualizaciones [Describir]
4. Dar un ejemplo de un escenario en el que un intento de actualización optimista podría nunca completarse [Crear]
5. Modificar un sistema concurrente para usar un almacén de datos más escalable, confiable o disponible [Crear]
6. Usando una plataforma existente que admita almacenes de datos replicados, escribir un programa que mantenga un mapeo clave-valor incluso cuando uno o más hosts fallen [Escribir]

3.11.10. PDC/Coordinación (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Dependencias 1. La iniciación o progreso de una actividad puede depender de otras actividades, para evitar condiciones de carrera, garantizar la terminación o cumplir otros requisitos. 2. Garantizar el progreso evitando ciclos de dependencia, usando condiciones monótonas, eliminando dependencias no esenciales.
- Constructos de control y patrones de diseño: 1. Basados en finalización: Barreras, reuniones (joins), incluido el control de terminación. 2. Habilitados por datos: Colas, diseños productor-consumidor. 3. Basados en condición: Polling, reintentos, retrocesos, ayuda, suspensión, señalización, tiempos de espera. 4. Reactivos: Habilitar y activar continuaciones.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Mostrar cómo garantizar que un programa termine correctamente cuando todas las tareas concurrentes de un conjunto hayan finalizado [Diseñar]
2. Escribir una función que cuente eventos, como entradas de sensores o recepciones de paquetes de red, de manera eficiente [Escribir]
3. Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir]
4. Escribir un programa en el que la terminación de un conjunto de acciones paralelas sea seguida por otra [Escribir]
5. Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]

3.11.11. PDC/Coordinación: Sincronización y Atomicidad (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Atomicidad 1. Instrucciones atómicas, ordenamientos de acceso local impuestos. 2. Bloqueos y exclusión mutua; granularidad de bloqueo. 3. Uso de bloqueos en un lenguaje específico; mantener la vivacidad sin introducir carreras. 4. Evitar punto muerto (*deadlock*): Ordenamiento, mayor granularidad, reintentos aleatorios; retrocesos, encapsulación mediante administradores de bloqueos. 5. Errores comunes: No bloquear o desbloquear cuando es necesario, mantener bloqueos mientras se invocan operaciones desconocidas. 6. Evitar bloqueos: replicación, solo lectura, propiedad, y construcciones no bloqueantes.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Mostrar cómo evitar o reparar un error de carrera en un programa dado [Analizar]

3.11.12. PDC/Coordinación: Propiedades Avanzadas (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Una o más de las siguientes propiedades y extensiones: 1. Propiedades de progreso incluyendo libre de bloqueo (*lock-free*), libre de espera (*wait-free*), equidad (*fairness*), planificación por prioridad, interacciones con consistencia, confiabilidad. 2. Rendimiento con respecto a contención, granularidad, convoy, escalabilidad. 3. Estructuras de datos y algoritmos no bloqueantes. 4. Propiedad y control de recursos. 5. Variantes y alternativas de bloqueos: bloqueos de secuencia, bloqueos de lectura-escritura; Actualizar-Copiando-Lectura (RCU), reentrada; tickets; control del ciclo activo (*spinning*) versus bloqueo. 6. Control basado en transacciones: Optimista y conservador. 7. Bloqueo distribuido: confiabilidad. 8. Alternativas a barreras: Relojes; contadores, relojes virtuales; flujo de datos y continuaciones; futuros y RPC; basadas en consenso, recolección de resultados con reductores y colectores. 9. Especulación, selección, cancelación; consecuencias en observabilidad y seguridad. 10. Control de recursos usando semáforos y variables de condición. 11. Flujo de control: Planificación de computaciones, bucles serie-paralelo con líderes (posiblemente elegidos), tuberías y flujos, paralelismo anidado. 12. Excepciones y fallos. Manejadores, detección, tiempos de espera, tolerancia a fallos, votación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir un programa que busque especulativamente una solución mediante múltiples actividades, terminando las demás cuando se encuentre una [Escribir]
2. Escribir un programa en el que una excepción numérica (como división por cero) en una actividad cause la terminación de las demás [Escribir]
3. Escribir un programa para que múltiples partes acuerden la hora actual del día; discutir sus limitaciones en comparación con protocolos como el protocolo de transferencia de red (NTP) [Escribir]

3.11.13. PDC/Evaluación (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Requisitos de seguridad (safety) y vivacidad (*liveness*) en términos de constructos de lógica temporal para expresar ”siempre” y ”eventualmente” Ver también: Computación Paralela y Distribuida.
- Identificar, probar y reparar violaciones, incluyendo formas comunes de errores como no garantizar el orden necesario (errores de carrera), atomicidad (incluyendo errores de ”verificar luego actuar”) y terminación (bloqueo activo).

- Métricas de requisitos de rendimiento para rendimiento (*throughput*), capacidad de respuesta, latencia, disponibilidad, consumo de energía, escalabilidad, uso de recursos, costos de comunicación, espera y control de velocidad, equidad; acuerdos de nivel de servicio. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento.
- Impacto en el rendimiento de las opciones de diseño e implementación, incluyendo granularidad, sobrecarga, costos de consenso y consumo de energía. Ver también: Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social.
- Estimar limitaciones de escalabilidad, por ejemplo usando la Ley de Amdahl o la Ley de Escalabilidad Universal. Ver también: Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Revisar una especificación para habilitar el paralelismo y la distribución sin violar otras propiedades o características esenciales [Rediseñar]
2. Explicar cómo las nociones concurrentes de seguridad (safety) y vivacidad (liveness) extienden sus contrapartes secuenciales [Explicar]
3. Especificar un conjunto de invariantes que deben mantenerse en cada paso de cómputo de paralelismo masivo [Analizar]
4. Escribir un programa de prueba que pueda revelar un error de carrera de datos; por ejemplo, perder una actualización cuando dos actividades intentan incrementar una variable [Escribir]
5. En un contexto dado, explicar hasta qué punto se esperaría que introducir paralelismo en un programa por lo demás secuencial mejoraría el rendimiento (*throughput*) y/o reduciría la latencia, y cómo podría afectar la eficiencia energética [Explicar]
6. Mostrar cómo cambian la escalabilidad y la eficiencia para problemas de muestra con y sin el supuesto de que el tamaño del problema cambia con el número de procesadores; además, explicar si y cómo cambiaría la escalabilidad bajo relajaciones de dependencias secuenciales [Diseñar]

3.11.14. PDC/Evaluación: Métodos Formales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Métodos formales de verificación y análisis: 1. Extensiones a requisitos formales secuenciales como la linealizabilidad. 2. Especificaciones de protocolo, sesión y transacciones. 3. Uso de herramientas como Lenguaje Unificado de Modelado (UML), Lógica Temporal de Acciones (TLA), lógicas de programa. 4. Análisis de seguridad: seguridad y vivacidad en presencia de comportamientos hostiles o con errores de otras partes; propiedades requeridas de mecanismos de comunicación (por ejemplo, ausencia de fugas entre capas), filtrado de entrada, limitación de velocidad. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad. 5. Análisis estático aplicado a corrección, rendimiento (*throughput*), latencia, recursos, energía. Ver también: Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social. 6. Análisis del modelo de Grafo Acíclico Dirigido (DAG) de eficiencia algorítmica (trabajo, span, caminos críticos).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Especificar y medir el comportamiento cuando un servicio es solicitado por un número inesperadamente grande de clientes [Analizar]
2. Identificar y reparar un problema de rendimiento debido a cuellos de botella secuenciales [Analizar]
3. Comparar empíricamente el rendimiento (*throughput*) de dos implementaciones de un diseño común (quizás usando un marco de pruebas existente) [Comparar]

3.11.15. PDC/Evaluación: Pruebas y Medición (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Herramientas de prueba y técnicas de medición: 1. Pruebas y depuración; herramientas como detectores de carreras, fuzzers, verificadores de dependencia de bloqueos, pruebas de unidad/esfuerzo/tortura, visualizaciones, integración continua (CI), despliegue continuo (CD), y generadores de pruebas. 2. Medir y comparar rendimiento (*throughput*), sobrecarga, espera, contención, comunicación, movimiento de datos, localidad, uso de recursos, comportamiento en presencia de números excesivos de eventos, clientes o hilos. Ver también: Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento. 3. Análisis específicos del dominio de aplicación y técnicas de evaluación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar y reparar un problema de rendimiento debido a latencia de comunicación o datos [Analizar]
2. Identificar y reparar un problema de rendimiento debido a sobrecarga en la gestión de recursos [Analizar]
3. Identificar y reparar un problema de confiabilidad o disponibilidad [Analizar]

3.11.16. PDC/Algoritmos (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Expresar e implementar algoritmos en lenguajes y marcos de trabajo dados, para iniciar actividades (por ejemplo hilos), usar constructos de memoria compartida, y APIs de canales, sockets y/o llamada a procedimiento remoto (RPC). Ver también: Computación Paralela y Distribuida. 1. Ejemplos de datos paralelos incluyendo mapeo/reducción. 2. Uso de APIs de canal, socket y/o RPC en un lenguaje dado, con control del programa para enviar (generalmente procedimental) vs recibir (generalmente reactivo o basado en RPC). 3. Uso de bloqueos, barreras y/o sincronizadores para mantener la vivacidad sin introducir carreras.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar un componente paralelo/distribuido basado en un algoritmo conocido [Implementar]
2. Escribir un programa de datos paralelos que, por ejemplo, calcule el promedio de un arreglo de números [Escribir]
3. Escribir un programa productor-consumidor en el que un componente genere números y otro calcule su promedio. Medir las aceleraciones cuando los números son escalares pequeños versus valores de multiprecisión grandes [Escribir]

3.11.17. PDC/Algoritmos: Dominios de Aplicación (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Panorama de dominios de aplicación comunes en multinúcleo, reactivo, datos paralelos, clúster, nube, sistemas distribuidos abiertos y marcos de trabajo (con referencia a la siguiente tabla). 1. Multinúcleo: Agentes de ejecución típicos: Hilos. Mecanismos de comunicación típicos: Memoria compartida, Atómicos, bloqueos. Dominios algorítmicos típicos: Gestión de recursos, procesamiento de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*), latencia, energía. 2. Reactivo: Agentes de ejecución típicos: Manejadores, hilos. Mecanismos de comunicación típicos: Canales de E/S. Dominios algorítmicos típicos: Servicios, tiempo real. Objetivos de ingeniería típicos: Latencia. 3. Datos paralelos: Agentes de ejecución típicos: GPU, SIMD, aceleradores, híbridos. Mecanismos de comunicación típicos: Memoria heterogénea. Dominios algorítmicos típicos: álgebra lineal, gráficos, análisis de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*), energía. 4. Clúster: Agentes de ejecución típicos: Hosts gestionados. Mecanismos de comunicación típicos: Sockets, canales. Dominios algorítmicos típicos: Simulación, análisis de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*). 5. Nube: Agentes de ejecución típicos: Hosts aprovisionados. Mecanismos de comunicación típicos: APIs

de servicio. Dominios algorítmicos típicos: Aplicaciones web. Objetivos de ingeniería típicos: Escalabilidad. 6. Distribuido abierto: Agentes de ejecución típicos: Hosts autónomos. Mecanismos de comunicación típicos: Sockets, Almacenes de datos. Dominios algorítmicos típicos: Almacenes de datos y servicios tolerantes a fallos. Objetivos de ingeniería típicos: Confiabilidad.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Extender un programa secuencial dirigido por eventos estableciendo una nueva actividad en un manejador de eventos (por ejemplo, un nuevo hilo en un manejador de acciones de GUI) [Diseñar]
2. Mejorar el rendimiento de un componente secuencial introduciendo paralelismo y/o distribución [Crear]
3. Elegir entre diferentes diseños paralelos/distribuidos para componentes de un sistema dado [Evaluar (valorar)]

3.11.18. PDC/Algoritmos: Dominios Algorítmicos (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:

Core

- Uno o más de los siguientes dominios algorítmicos. Ver también: Estrategias Algorítmicas: 1. álgebra lineal: Operaciones con vectores y matrices, precisión/estabilidad numérica, aplicaciones en análisis de datos y aprendizaje automático. 2. Procesamiento de datos: ordenación, búsqueda y recuperación, estructuras de datos concurrentes. 3. Grafos, búsqueda y combinatoria: Marcado, paralelización de aristas, acotamiento, especulación, análisis basado en redes. 4. Modelado y simulación: ecuaciones diferenciales; aleatorización, problemas de N-cuerpos, algoritmos genéticos. 5. Lógica computacional: satisfactibilidad (SAT), programación lógica concurrente. 6. Gráficos y geometría computacional: Transformaciones, renderizado, trazado de rayos. 7. Gestión de recursos: Asignar, colocar, reciclar y planificar procesadores, memoria, canales y hosts; recursos exclusivos vs compartidos; algoritmos estáticos, dinámicos y elásticos; Restricciones de tiempo real; Lotes, priorización, partición; descentralización mediante robo de trabajo y técnicas relacionadas. 8. Servicios: Implementar APIs web, moneda electrónica, sistemas de transacción, juegos multijugador.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Diseñar, implementar, analizar y evaluar un componente o aplicación para X que opere en un contexto dado, donde X esté en uno de los dominios listados, por ejemplo, un algoritmo genético para el diseño de una planta de fábrica [Diseñar]
2. Criticar el diseño e implementación de un componente o aplicación existente, o uno desarrollado por compañeros de clase [Críticar (análisis crítico)]
3. Comparar el rendimiento y la eficiencia energética de múltiples implementaciones de un diseño similar, por ejemplo, multinúcleo versus clúster versus GPU [Comparar]

3.12. Fundamentos de Desarrollo de Software (SDF)

La fluidez en el proceso de desarrollo de software es fundamental para el estudio de las ciencias de la computación. Para utilizar las computadoras de manera más efectiva para resolver problemas, los estudiantes deben ser competentes en la lectura y escritura de programas. Sin embargo, más allá de las habilidades de programación, deben poder seleccionar y utilizar estructuras de datos y algoritmos apropiados, así como herramientas modernas de desarrollo y prueba.

El área de conocimiento SDF reúne conceptos y habilidades fundamentales relacionados con el desarrollo de software, centrándose en aquellos que deben enseñarse al inicio de un programa de ciencias de la computación, típicamente en el primer año. Esto incluye conceptos de programación fundamentales y su uso efectivo para escribir programas, el uso de estructuras de datos fundamentales que pueden ser proporcionadas por el lenguaje de programación, conceptos básicos de buenas prácticas de programación para escribir programas de calidad, la lectura y comprensión de programas, y cierta

comprensión del impacto de los algoritmos en el rendimiento de los programas. Las 43 horas de material en esta área de conocimiento pueden complementarse con material central de otras áreas a medida que los estudiantes avanzan a cursos de nivel intermedio y superior.

Esta área de conocimiento asume un lenguaje de programación contemporáneo con soporte incorporado para tipos de datos comunes, incluidos tipos de datos asociativos como diccionarios/mapas, como vehículo para introducir a los estudiantes a la programación (por ejemplo, Python, Java). Sin embargo, esto no pretende desalentar el uso de lenguajes más antiguos o de bajo nivel para SDF: las unidades de conocimiento a continuación pueden adaptarse adecuadamente al lenguaje real utilizado.

El surgimiento de la IA generativa y los Modelos de Lenguaje Grande (LLMs), que pueden generar programas para muchas tareas de programación, sin duda afectará a la profesión de programación y, en consecuencia, a la enseñanza de muchos temas de CS. Sin embargo, para poder utilizar efectivamente la IA generativa en tareas de programación, un programador debe tener una buena comprensión de los programas y, por lo tanto, aún debe aprender los fundamentos de la programación y desarrollar habilidades básicas de programación, que es el objetivo de SDF. En consecuencia, consideramos que los resultados deseados para SDF deberían permanecer iguales, aunque diferentes instructores ahora podrían dar más énfasis a la comprensión, documentación, especificaciones, análisis y pruebas de programas. (Esto es similar a enseñar a los estudiantes multiplicación, suma, etc., aunque se puedan usar calculadoras para hacerlo).

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.12.1 Conceptos y Construcciones Básicas de Programación (Pág. 119)	10	
⭐ 3.12.2 Prácticas de Programación: Pruebas, Depuración y Documentación (Pág. 120)	10	
⭐ 3.12.3 Estructuras de Datos Fundamentales (Pág. 120)	12	
⭐ 3.12.4 Algoritmos (Pág. 121)	6	
⭐ 3.12.5 Prácticas de Desarrollo de Software (Pág. 121)	5	
3.12.6 Sociedad, ética y la Profesión (Pág. 121)	Electivo	

Cuadro 3.12: Lista de KUs del área de Fundamentos de Desarrollo de Software.

3.12.1. SDF/Conceptos y Construcciones Básicas de Programación (Core Tier1: 10 hrs)

Temas:

Core

- Conceptos básicos como variables, tipos de datos primitivos, expresiones y su evaluación
- Cómo funcionan los programas imperativos: estado y transiciones de estado en la ejecución de sentencias, flujo de control
- Construcciones básicas como sentencias de asignación, sentencias condicionales e iterativas, E/S básica
- Construcciones clave de modularidad como funciones (y métodos y clases, si son compatibles en el lenguaje) y conceptos relacionados como paso de parámetros, alcance, abstracción, encapsulamiento de datos Fundamentos de Programación Orientada a Objetos, Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión
- Entrada y salida usando archivos y APIs
- Tipos de datos estructurados disponibles en el lenguaje de programación elegido como secuencias (por ejemplo, arreglos, listas), contenedores asociativos (por ejemplo, diccionarios, mapas), otros (por ejemplo, conjuntos, tuplas) y cuándo y cómo usarlos Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Desarrollar programas que utilicen las construcciones fundamentales de programación: asignación y expresiones, E/S básica, sentencias condicionales e iterativas [Crear]
2. Desarrollar programas utilizando funciones con paso de parámetros [Crear]
3. Desarrollar programas que utilicen efectivamente los diferentes tipos de datos estructurados proporcionados en el lenguaje, como arreglos/listas, diccionarios y conjuntos [Crear]

4. Desarrollar programas que utilicen E/S de archivos para proporcionar persistencia de datos a través de múltiples ejecuciones [Crear]
5. Desarrollar programas que utilicen APIs para acceder o actualizar datos (por ejemplo, desde la web) [Crear]
6. Desarrollar programas que creen clases simples e instancien objetos de esas clases (si es compatible con el lenguaje) [Crear]
7. Rastrear el flujo de control durante la ejecución de un programa [Trazar/Rastrear]
8. Usar la terminología apropiada para identificar elementos de un programa (por ejemplo, identificador, operador, operando) [Usar]

3.12.2. SDF/Prácticas de Programación: Pruebas, Depuración y Documentación (Core Tier1: 10 hrs)

Temas:
Core

- Bibliotecas y frameworks proporcionados por el lenguaje (cuando/donde sea aplicable)
- Recursión
- Manejo de errores de tiempo de ejecución en programas (por ejemplo, manejo de excepciones).
- Conceptos básicos de errores de programación, pruebas y depuración Prácticas de Codificación, Construcción a Escala y Proceso, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada
- Documentar/comentar código a nivel de programa y de módulo. Prácticas de Codificación, Construcción a Escala y Proceso
- Desarrollar una mentalidad de seguridad. Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Desarrollar programas que utilicen bibliotecas y frameworks proporcionados por el lenguaje (donde sea aplicable) [Crear]
2. Explicar el concepto de recursión e identificar cuándo y cómo usarlo efectivamente [Explicar]
3. Desarrollar funciones recursivas [Crear]
4. Desarrollar programas que puedan manejar errores de tiempo de ejecución [Crear]
5. Leer un programa dado y explicar lo que hace [Explicar]
6. Escribir comentarios para un programa o módulo especificando lo que hace [Escribir]

3.12.3. SDF/Estructuras de Datos Fundamentales (Core Tier1: 12 hrs)

Temas:
Core

- Tipos abstractos de datos estándar como listas, pilas, colas, conjuntos y mapas/diccionarios, incluyendo operaciones sobre ellos.
- Seleccionar y usar estructuras de datos apropiadas.
- Implicaciones de rendimiento de la elección de estructura(s) de datos.
- Cadenas y procesamiento de cadenas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir programas que utilicen cada uno de los tipos abstractos de datos clave proporcionados en el lenguaje (por ejemplo, arreglos, tuplas/registros/estructuras, listas, pilas, colas y tipos de datos asociativos como conjuntos, diccionarios/mapas) [Escribir]
2. Seleccionar la estructura de datos apropiada para un problema dado [Evaluar]
3. Explicar cómo puede cambiar el rendimiento de un programa al usar diferentes estructuras de datos u operaciones [Explicar]
4. Escribir programas que trabajen con texto utilizando las capacidades de procesamiento de cadenas proporcionadas por el lenguaje [Escribir]

3.12.4. SDF/Algoritmos (Core Tier1: 6 hrs)

Temas:
Core

- Concepto de algoritmo y noción de eficiencia algorítmica
- Algunos algoritmos comunes (por ejemplo, ordenación, búsqueda, recorrido de árboles, recorrido de grafos)
- Impacto de los algoritmos en la eficiencia tiempo-espacio de los programas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar el rol de los algoritmos para escribir programas [Explicar]
2. Demostrar cómo un problema puede ser resuelto por diferentes algoritmos, cada uno con diferentes propiedades [Demostrar]
3. Explicar algunos algoritmos comunes (por ejemplo, ordenación, búsqueda, recorrido de árboles, recorrido de grafos) [Explicar]
4. Explicar el impacto en el rendimiento espacio/tiempo de algunos algoritmos [Explicar]

3.12.5. SDF/Prácticas de Desarrollo de Software (Core Tier1: 5 hrs)

Temas:
Core

- Pruebas básicas, incluido el diseño de casos de prueba
- Uso de un IDE de propósito general, incluido su depurador
- Estilo de programación que mejora la legibilidad
- Especificar la funcionalidad de un módulo en lenguaje natural.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Desarrollar pruebas para módulos y aplicar una variedad de estrategias para diseñar casos de prueba [Crear]
2. Explicar algunas limitaciones de las pruebas de programas [Explicar]
3. Compilar, ejecutar y depurar programas utilizando un IDE moderno y herramientas asociadas como depuradores visuales [Analizar]
4. Aplicar pautas básicas de estilo de programación para ayudar a la legibilidad de los programas, como comentarios, sangría, nombres adecuados de variables, etc [Aplicar]
5. Escribir especificaciones de un módulo como comentario de módulo que describa su funcionalidad [Escribir]

3.12.6. SDF/Sociedad, ética y la Profesión

Temas:
Core

- Derechos de propiedad intelectual de los programadores para los programas que desarrollan.
- Plagio e integridad académica.
- Responsabilidad y obligación de los programadores con respecto al código que desarrollan para soluciones. Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad
- ética profesional básica del trabajo de los programadores.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar/comprender algunos de los problemas de propiedad intelectual relacionados con programas [Explicar]
2. Explicar/comprender cuándo se puede utilizar código desarrollado por otros y las formas adecuadas de revelar su uso [Explicar]
3. Explicar/comprender la responsabilidad de los programadores al desarrollar código para una solución general (que puede ser desarrollada por un equipo) [Explicar]
4. Explicar/comprender uno o más códigos de conducta aplicables a los programadores [Explicar]

3.13. Ingeniería de Software (SE)

Ya a principios de la década de 1970, el informático británico Brian Randell supuestamente dijo: "La ingeniería de software es la construcción multi-persona de programas multi-versión". Esta es una visión esencial: mientras que la programación es la habilidad que gobierna nuestra capacidad para escribir un programa, la ingeniería de software es distinta en dos dimensiones: tiempo y personas.

Primero, un proyecto de ingeniería de software es un esfuerzo de equipo; ser un experto en programación solitario es insuficiente. Los ingenieros de software calificados deben demostrar experiencia en comunicación y colaboración. La programación puede ser una actividad individual, pero la ingeniería de software es colaborativa, profundamente ligada a cuestiones de profesionalismo, trabajo en equipo y comunicación.

Segundo, un proyecto de ingeniería de software suele ser "multi-versión". Tiene una vida útil esperada; necesita funcionar correctamente durante meses, años o décadas. Las funcionalidades pueden agregarse o eliminarse para cumplir con los requisitos del producto. El propio equipo de ingeniería probablemente cambiará. El contexto tecnológico cambiará, a medida que evolucionen nuestras plataformas informáticas, cambien los lenguajes de programación, se actualicen las dependencias, etc. Esta exposición a cuestiones de tiempo y cambio es novedosa en comparación con un proyecto de programación: no basta con construir algo que funcione; debe funcionar y seguir funcionando. Muchos de los temas más desafiantes en tecnología comparten "el tiempo conducirá al cambio" como causa raíz: compatibilidad con versiones anteriores, desfase de versiones, gestión de dependencias, cambios de esquema, evolución de protocolos.

La ingeniería de software presenta un desafío particularmente difícil para el aprendizaje en un entorno académico. Dado que las principales diferencias entre la programación y la ingeniería de software son el tiempo y el trabajo en equipo, es difícil generar lecciones que requieran un trabajo en equipo exitoso y que presenten fielmente los desafíos del tiempo. Además, algunos temas de ingeniería de software serán más auténticos y relevantes si nuestros estudiantes experimentan proyectos colaborativos y a largo plazo in vivo en lugar de en el aula. Independientemente de que esto suceda como una pasantía, la participación en un proyecto de código abierto o un rol de ingeniería a tiempo completo, un mes de experiencia práctica a tiempo completo tiene más horas disponibles que el curso promedio de ingeniería de software.

Por lo tanto, un plan de estudios de ingeniería de software debe centrarse en los conceptos necesarios para la mayoría de los nuevos contratados graduados, y que son novedosos para aquellos que se capacitan principalmente como programadores, o que son conceptos abstractos que pueden no declararse/compartirse explícitamente en el trabajo. Tales temas incluyen, entre otros:

- Pruebas
- Trabajo en equipo, colaboración
- Comunicación
- Diseño
- Mantenimiento y evolución
- Herramientas de ingeniería de software

Todas las pruebas sugieren que el papel del software en nuestra sociedad seguirá creciendo en el futuro previsible. Además, la era de "dos programadores en un garaje" parece haber llegado a su fin. La mayoría del software importante hoy en día es un esfuerzo de equipo, que se basa en código existente y aprovecha la funcionalidad existente. El estudio de las habilidades de ingeniería de software es un contrapunto profundamente importante a la experiencia cotidiana de los estudiantes de informática: debemos impresionarles con la realidad de que pocos proyectos de software se gestionan escribiendo desde cero como un esfuerzo en solitario. La comunicación, el trabajo en equipo, la planificación, las pruebas y las herramientas son mucho más importantes a medida que nuestros estudiantes pasan del aula y dejan su huella en el mundo.

Aunque la mayoría de los graduados en CS ocuparán un puesto en la industria que requiere este material, los temas del CS Core presentados aquí son valiosos independientemente de si los graduados van a la industria o a la academia.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.13.1 Colaboración y Dinámica de Equipo (Pág. 123)	2	2
⭐ 3.13.2 Diversidad, Stakeholders y Equipos Distribuidos (Pág. 124)	1	1
⭐ 3.13.3 Control de Versiones y CI/CD (Pág. 124)	1	2
⭐ 3.13.4 Entorno de Desarrollo e IDE (Pág. 125)	1	2
⭐ 3.13.5 Ingeniería de Requisitos (Pág. 125)		2
⭐ 3.13.6 Evolución y Estimación de Requisitos (Pág. 126)		1
⭐ 3.13.7 Principios y Arquitectura de Software (Pág. 126)	1	2
⭐ 3.13.8 Diseño de API y Modelado (Pág. 127)	1	1
⭐ 3.13.9 Calidad y Evaluación del Diseño (Pág. 127)	1	2
⭐ 3.13.10 Prácticas de Codificación (Pág. 128)	1	2
⭐ 3.13.11 Construcción a Escala y Proceso (Pág. 128)	1	2
⭐ 3.13.12 Conceptos de verificación y validación (Pág. 129)	1	1
⭐ 3.13.13 Planificación y Tipos de Pruebas (Pág. 129)	1	1
⭐ 3.13.14 Prácticas Avanzadas de Pruebas (Pág. 130)	1	1
⭐ 3.13.15 Herramientas y Análisis de Pruebas (Pág. 130)	1	1
⭐ 3.13.16 Pruebas de Rendimiento y Evaluación Comparativa (Pág. 131)	1	1
⭐ 3.13.17 Compatibilidad y Versionado de API (Pág. 131)		2
⭐ 3.13.18 Técnicas de Refactorización (Pág. 132)		1
⭐ 3.13.19 Conceptos de Confiabilidad (Pág. 132)		2
⭐ 3.13.20 Ingeniería de Confiabilidad (Pág. 133)		1
3.13.21 Métodos Formales (Pág. 133)	Electivo	

Cuadro 3.13: Lista de KUs del área de Ingeniería de Software.

3.13.1. SE/Colaboración y Dinámica de Equipo (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Temas:

Core

- Comunicación efectiva, incluida la oral y escrita, así como formal (correo electrónico, documentos, comentarios, presentaciones) e informal (chat de equipo, reuniones). Comunicación Técnica y Profesional, Comunicación en Equipo y Cultura
- Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos.
- Programación cooperativa: 1. Programación por parejas (pair programming) o Enjambre (swarming) 2. Revisión de código 3. Colaboración mediante control de versiones
- Roles y responsabilidades en un equipo de software: Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo 1. Ventajas del trabajo en equipo 2. Riesgos y complejidad de dicha colaboración
- Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Seguir prácticas efectivas de comunicación en equipo [Analizar]
2. Articular las fuentes, peligros y beneficios potenciales del conflicto en el equipo, especialmente enfocándose en el valor de estar en desacuerdo sobre ideas o propuestas sin insultar a las personas [Articular]
3. Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar]
4. Colaborar efectivamente en desarrollo/programación cooperativa [Analizar]
5. Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer]
6. Redactar y seguir una agenda para una reunión de equipo [Componer]
7. Facilitar, a través de la participación en un proyecto de equipo, los elementos centrales de la

formación de equipos, el establecimiento de una cultura de equipo saludable y la gestión de equipos, incluida la creación y ejecución de un plan de trabajo del equipo [Crear]

3.13.2. SE/Diversidad, Stakeholders y Equipos Distribuidos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Importancia de la diversidad e inclusión en el equipo. Comunicación Técnica y Profesional, Comunicación en Equipo y Cultura
- Interfaz con las partes interesadas, como equipo: 1. Gestión y otros equipos no técnicos 2. Clientes 3. Usuarios
- Riesgos asociados con equipos físicos, distribuidos, híbridos y virtuales, incluyendo comunicación, percepción, estructura, puntos de falla, mitigación y recuperación, etc.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Promover la importancia y los beneficios que la diversidad y la inclusión aportan a un equipo de desarrollo de software [Defender]
2. Referenciar, como equipo, la importancia de, y las estrategias para interactuar con partes interesadas fuera del equipo en niveles tanto técnicos como no técnicos [Referenciar]
3. Enumerar los riesgos asociados con equipos físicos, distribuidos, híbridos y virtuales y los posibles puntos de falla y cómo mitigarlos y recuperarse/aprender de los fallos [Enumerar]

3.13.3. SE/Control de Versiones y CI/CD (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software 1. Configuración en control de versiones, builds/configuración reproducibles. 2. Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). 3. Estrategias de fusión/rebase, cuando sean relevantes.
- Gestión de lanzamientos (*release management*).
- Herramientas de prueba, incluyendo herramientas de análisis estático y dinámico. Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada
- Automatización de procesos de software: 1. Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. 2. Integración Continua (CI): el uso de automatización y pruebas automatizadas para hacer una validación preliminar de que la revisión actual del tronco se construye y pasa las pruebas (básicas). 3. Despliegue Continuo (CD): el uso de automatización para liberar de forma automática cada cambio que supera las pruebas hacia el entorno de producción, garantizando entregas frecuentes y confiables. 4. Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer.
- Flujo de revisión colaborativa como puerta de calidad 1. Revisión obligatoria de *pull requests* antes de fusionar 2. Propiedad del código mediante reglas de tipo CODEOWNERS 3. Convenciones de mensajes de *commit* (*conventional commits*) 4. Reglas de protección de ramas que exigen verificaciones antes de fusionar
- Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir]
2. Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir]

3. Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar]
4. Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir]
5. Comprender el uso de sistemas de CI/CD como una fuente de verdad para el estado del código compartido del equipo (éxito en la construcción y pruebas) [Explicar]
6. Aplicar un flujo de revisión de *pull requests* con reglas de propiedad de código y protección de ramas para imponer puertas de calidad en un proyecto de equipo [Aplicar]
7. Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]

3.13.4. SE/Entorno de Desarrollo e IDE (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Herramientas de diseño y comunicación (documentos, diagramas, formas comunes de diagramas de diseño como UML).
- Conceptos y mecanismos de integración de herramientas. Prácticas de Desarrollo de Software
- Uso de las facilidades de un IDE moderno: depuración, refactorización, búsqueda/indexación, asistentes de código basados en ML, etc. Prácticas de Desarrollo de Software

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir los problemas importantes al seleccionar un conjunto de herramientas para el desarrollo de un sistema de software específico, incluyendo herramientas para seguimiento de requisitos, modelado de diseño, implementación, automatización de construcción y pruebas [Describir]
2. Demostrar la capacidad de usar herramientas de software para apoyar el desarrollo de un producto de software de tamaño mediano [Demostrar]

3.13.5. SE/Ingeniería de Requisitos (Core Tier2: 2 hrs)

Temas:
Core

- Describir requisitos funcionales usando, por ejemplo, casos de uso o historias de usuario. 1. Usar al menos un método para documentar y estructurar requisitos funcionales. 2. Comprender cómo el método apoya el diseño y la implementación. 3. Fortalezas y debilidades de usar un enfoque específico.
- Propiedades de los requisitos, incluyendo consistencia, validez, completitud y viabilidad.
- Obtención (*elicitation*) de requisitos. 1. Fuentes de requisitos, por ejemplo, usuarios, administradores o personal de soporte. 2. Métodos de recopilación de requisitos, por ejemplo, encuestas, entrevistas o análisis de comportamiento.
- Requisitos no funcionales, por ejemplo, seguridad, usabilidad o rendimiento, también llamados Atributos de Calidad. Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social
- Identificación y gestión de riesgos, incluyendo consideraciones éticas en torno al producto propuesto. Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo
- Comunicar y/o formalizar especificaciones de requisitos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar diferentes métodos de obtención de requisitos a lo largo de múltiples ejes [Comparar]
2. Identificar diferencias entre dos métodos de describir requisitos funcionales (por ejemplo, entrevistas con clientes, estudios de usuarios) y las situaciones donde se preferiría cada uno [Analizar]
3. Identificar qué comportamientos son requeridos, permitidos o prohibidos a partir de un conjunto dado de requisitos y una lista de comportamientos candidatos [Analizar]
4. Recopilar un conjunto de requisitos para un sistema de software simple [Analizar]

5. Identificar áreas de un sistema de software que deben cambiarse, dada una descripción del sistema y un conjunto de nuevos requisitos a implementar [Analizar]
6. Identificar los requisitos funcionales y no funcionales en un conjunto de requisitos [Analizar]

3.13.6. SE/Evolución y Estimación de Requisitos (Core Tier2: 1 hr)

Temas:
Non Core

- Prototipado de una herramienta tanto para obtener como para validar/confirmar requisitos.
- Evolución del producto: cuando cambian los requisitos, cómo entender qué efecto tiene eso y qué cambios deben realizarse.
- Estimación de esfuerzo: 1. Aprender técnicas para estimar mejor el esfuerzo requerido para completar una tarea; 2. Practicar la estimación y compararla con cuánto tiempo toman las tareas; 3. La estimación de esfuerzo es bastante difícil, por lo que es probable que los estudiantes se equivoquen en muchos casos, pero ver el proceso desarrollarse con su propio trabajo es valioso.

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Crear un prototipo de un sistema de software para validar un conjunto de requisitos: construir una maqueta, MVP, etc [Crear]
2. Estimar el tiempo para completar un conjunto de tareas, luego comparar las estimaciones con el tiempo real tomado [Estimar]
3. Determinar una secuencia de implementación para un conjunto de tareas, respetando las dependencias entre ellas, con el objetivo de retirar el riesgo lo antes posible [Determinar]
4. Escribir una especificación de requisitos para un sistema de software simple [Escribir]

3.13.7. SE/Principios y Arquitectura de Software (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Principios de diseño de sistemas. Confiabilidad del Sistema 1. Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) 2. Separación de preocupaciones (separation of concerns) 3. Ocultación de información (information hiding) 4. Acoplamiento y cohesión (coupling and cohesion)
- Arquitectura de software. Confiabilidad del Sistema 1. Paradigmas de diseño *a*) Descomposición funcional de arriba hacia abajo/diseño por capas *b*) Arquitectura orientada a datos *c*) Análisis y diseño orientado a objetos *d*) Diseño dirigido por eventos 2. Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) 3. Identificar límites y dependencias de componentes
- Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema
- Malos olores del código (code smells) y otras indicaciones de la calidad del código, distintas de la corrección. Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar]
2. Seleccionar y usar un paradigma de diseño apropiado para diseñar un sistema de software simple y explicar cómo se han aplicado los principios de diseño de sistemas en este diseño [Evaluar]
3. Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear]
4. Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]

3.13.8. SE/Diseño de API y Modelado (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Principios de diseño de API 1. Consistencia *a*) Las API consistentes son más fáciles de aprender y menos propensas a errores *b*) La consistencia es tanto interna (entre diferentes partes de la API) como externa (siguiendo patrones comunes de API) 2. Capacidad de composición (*composability*) 3. Documentación de contratos *a*) Las operaciones de la API deben describir su efecto en el sistema, pero generalmente no su implementación *b*) Precondiciones, postcondiciones e invariantes 4. Capacidad de expansión (*expandability*) 5. Reporte de errores *a*) Los errores deben ser claros, predecibles y accionables *b*) La entrada que no coincida con el contrato debe producir un error *c*) Los errores que pueden gestionarse de manera confiable sin reportarse deben gestionarse
- Identificar y codificar invariantes de datos e invariantes temporales
- Modelos estructurales y de comportamiento de diseños de software

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar una API para un solo componente de un sistema de software grande, incluyendo la identificación y documentación de las invariantes, el contrato y las condiciones de error de cada operación [Diseñar]
2. Evaluar una descripción de API en términos de consistencia, capacidad de composición y capacidad de expansión [Evaluar]
3. Expandir un diseño existente para incluir una nueva funcionalidad [Crear]

3.13.9. SE/Calidad y Evaluación del Diseño (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Diseño de datos Modelado de Datos 1. Estructuras de datos 2. Sistemas de almacenamiento
- Trazabilidad de requisitos 1. Comprender qué requisitos son satisfechos por un diseño

Non Core

- Modelado de diseño, por ejemplo con diagramas de clases, diagramas de entidad-relación o diagramas de secuencia
- Medición y análisis de la calidad del diseño
- Principios de diseño y codificación seguros Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad 1. Principio de privilegio mínimo 2. Principio de valores predeterminados seguros ante fallos (fail-safe defaults) 3. Principio de aceptabilidad psicológica
- Evaluar compensaciones de diseño (tradeoffs) (por ejemplo, eficiencia vs confiabilidad, seguridad vs usabilidad)
- Modelo de características de calidad ISO/IEC 25010 y costo de la mala calidad 1. Características de calidad: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad, portabilidad 2. Costo de la mala calidad y la deuda técnica como preocupación de negocio/económica, no solo técnica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar un conjunto de estructuras de datos para implementar una superficie de API proporcionada [Diseñar]
2. Identificar qué requisitos son satisfechos por un diseño de software proporcionado [Analizar]

NonCore:

3. Traducir un diseño de software en lenguaje natural a diagramas de clases [Traducir]
4. Adaptar un diseño de sistema defectuoso para que siga mejor los principios de privilegio mínimo y valores predeterminados seguros ante fallos [Crear]
5. Contrastar dos diseños de software a través de diferentes cualidades, como eficiencia o usabilidad [Contrastar]
6. Evaluar un sistema de software frente a las características de calidad ISO/IEC 25010 y estimar el costo de negocio de la mala calidad/deuda técnica [Evaluar]

3.13.10. SE/Prácticas de Codificación (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Pruebas prácticas a pequeña escala Prácticas de Desarrollo de Software 1. Pruebas unitarias 2. Desarrollo guiado por pruebas (Test-driven development): esto es particularmente valioso para los estudiantes psicológicamente, ya que es mucho más fácil involucrarse constructivamente con el desafío de identificar entradas desafiantes para una API dada (casos límite, casos extremos) a priori. Si implementan primero, el instinto es a menudo evitar intentar hacer fallar su nueva creación, mientras que un enfoque de pruebas primero les da la satisfacción intelectual de detectar los casos problemáticos y luego ver cómo pasan más pruebas durante el proceso de desarrollo.
- Documentación Prácticas de Desarrollo de Software 1. Documentación de la interfaz: describir los requisitos de la interfaz, potencialmente incluyendo contratos (formales o informales), pre y post condiciones, invariantes. 2. La documentación de la implementación debe centrarse en partes de código complicadas y no obvias, ya sea porque el código usa características avanzadas del lenguaje o porque el comportamiento del código es complejo. (No agregar comentarios que reafirmen operaciones comunes/obvias y características simples del lenguaje.) *a*) Aclarar el flujo de datos, cómputo, etc., centrándose en lo que el código es. *b*) Identificar partes sutiles/complicadas del código y refactorizar para que sean autoexplicativas si es posible o proporcionar comentarios apropiados para aclarar.
- Estilo de codificación Prácticas de Desarrollo de Software 1. Guías de estilo 2. Comentarios 3. Nomenclatura (*naming*)
- "Mejores Prácticas" para la codificación: técnicas, patrones/idiomas, mecanismos para construir programas de calidad Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada 1. Prácticas de codificación defensiva 2. Prácticas y principios de codificación segura 3. Usar mecanismos de manejo de excepciones para hacer los programas más robustos, tolerantes a fallos
- Depuración (debugging) Prácticas de Desarrollo de Software
- Registro (logging)
- Uso de bibliotecas y frameworks desarrollados por otros Prácticas de Desarrollo de Software

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Escribir pruebas unitarias apropiadas para un componente pequeño (varias funciones, un solo tipo, etc.) [Escribir]
2. Escribir comentarios apropiados de interfaz y (si es necesario) de implementación para un componente pequeño [Escribir]
3. Describir técnicas, patrones de codificación y mecanismos para implementar diseños para lograr propiedades deseadas como confiabilidad, eficiencia y robustez [Describir]
4. Escribir código robusto utilizando mecanismos de manejo de excepciones [Escribir]
5. Describir prácticas de codificación segura y defensiva [Describir]
6. Seleccionar y usar un estándar de codificación definido en un proyecto de software pequeño [Usar]
7. Comparar y contrastar estrategias de integración, incluyendo integración de arriba hacia abajo (top-down), de abajo hacia arriba (bottom-up) y sándwich [Comparar]
8. Describir el proceso de analizar e implementar cambios en la base de código desarrollada para un proyecto específico [Describir]
9. Describir el proceso de analizar e implementar cambios en una base de código existente grande [Describir]

3.13.11. SE/Construcción a Escala y Proceso (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Non Core

- Pruebas a mayor escala 1. Dobles de prueba (test doubles) (*stubs*, *mocks*, *fakes*) 2. Inyección de dependencias (*dependency injection*)

- Secuenciación del trabajo, incluyendo identificación de dependencias, hitos y retiro de riesgo 1. Identificación de dependencias: identificar las dependencias entre diferentes tareas 2. Hitos: una colección de tareas que sirve como marcador de progreso cuando se completa. Idealmente, el hito abarca una unidad útil de funcionalidad. 3. Retiro de riesgo: identificar qué elementos de un proyecto son riesgosos y priorizar la finalización de tareas que aborden esos riesgos.
- Problemas potenciales de seguridad en programas Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada 1. Desbordamientos de búfer y otros tipos de desbordamientos 2. Condiciones de carrera (race conditions) 3. Inicialización incorrecta, incluida la elección de privilegios 4. Validación de entrada (input validation)
- Documentación (autogenerada)
- Contexto de desarrollo: "campo verde" (*green field*) vs base de código existente 1. Análisis de impacto del cambio (*change impact analysis*) 2. Realización del cambio (*change actualization*)
- Gestión de lanzamientos (release management)
- Prácticas de DevOps 1. Definition of Done como umbral de calidad compartido para el trabajo terminado 2. Postmortems sin culpa (*blameless postmortems*) para aprender de incidentes/fallas 3. Cultura de calidad en equipo: propiedad compartida, retroalimentación y aprendizaje continuo

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Reescribir un programa simple para eliminar vulnerabilidades comunes, como desbordamientos de búfer, desbordamientos de enteros y condiciones de carrera [Aplicar]
2. Escribir un componente de software que realice alguna tarea no trivial y sea resistente a errores de entrada y de tiempo de ejecución [Escribir]

3.13.12. SE/Conceptos de verificación y validación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Conceptos de verificación y validación 1. Verificación: ¿Estamos construyendo la cosa correctamente? 2. Validación: ¿Construimos la cosa correcta?
- Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona?
- Objetivos de las pruebas 1. Usabilidad 2. Confiabilidad 3. Conformidad con la especificación 4. Rendimiento 5. Seguridad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar por qué las pruebas son importantes [Explicar]
2. Distinguir entre validación y verificación de programas [Distinguir]
3. Describir diferentes objetivos de las pruebas [Describir]

3.13.13. SE/Planificación y Tipos de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Tipos de pruebas 1. Unitarias 2. Integración 3. Validación 4. Sistema
- Diferencias estilísticas entre el código de prueba y el código de producción: DAMP vs DRY - se justifica más duplicación en el código de prueba.
- Planificación y generación de pruebas 1. Generación de casos de prueba, a partir de modelos formales, especificaciones, etc. 2. Cobertura de pruebas *a*) Matrices de prueba *b*) Cobertura de código - ¿cuánto del código está probado? *c*) Cobertura de entorno - ¿cuántas arquitecturas de hardware, sistemas operativos, navegadores, etc. están probados? 3. Datos y entradas de prueba

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar y contrastar los diferentes tipos y niveles de pruebas (regresión, unitarias, integración, sistemas y aceptación) [Comparar]
2. Describir técnicas para crear un plan de pruebas y generar casos de prueba [Describir]
3. Crear un plan de pruebas para un segmento de código de tamaño mediano que incluya una matriz de pruebas y generación de datos y entradas de prueba [Crear]
4. Implementar un plan de pruebas para un segmento de código de tamaño mediano [Implementar]

3.13.14. SE/Prácticas Avanzadas de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Desarrollo de pruebas 1. Desarrollo guiado por pruebas 2. Pruebas orientadas a objetos, uso de mocks e inyección de dependencias 3. Técnicas de prueba de caja opaca (anteriormente, caja negra) y caja transparente (anteriormente, caja blanca) 4. Herramientas de prueba, incluyendo cobertura de código, análisis estático y fuzzing
- Verificación y validación en el ciclo de desarrollo 1. Revisiones de código 2. Automatización de pruebas, incluyendo automatización de herramientas 3. Pruebas pre-commit y post-commit 4. Compensaciones entre cobertura de pruebas y rendimiento/latencia de las pruebas 5. Seguimiento y priorización de defectos: reproducibilidad de los defectos reportados
- Desafíos de verificación y validación específicos del dominio 1. Pruebas de rendimiento y evaluación comparativa (benchmarking) 2. Asincronía, paralelismo y concurrencia 3. Crítico para la seguridad (*safety-critical*) 4. Numérico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar los principios fundamentales de los métodos de desarrollo guiado por pruebas y explicar el papel de las pruebas automatizadas en estos métodos [Analizar]
2. Discutir problemas relacionados con las pruebas de software orientado a objetos [Debatir]
3. Describir el uso de mocks y la inyección de dependencias y su aplicación [Describir]
4. Realizar, como parte de una actividad en equipo, una revisión de código de un segmento de código de tamaño mediano [Evaluar]
5. Describir el papel que las herramientas pueden desempeñar en la validación del software [Describir]
6. Automatizar las pruebas en un proyecto de software pequeño [Diseñar]
7. Explicar los roles, pros y contras de las pruebas pre-commit y post-commit [Explicar]
8. Discutir las compensaciones entre la cobertura de pruebas y el rendimiento/latencia de las pruebas y cómo esto puede afectar la verificación [Debatir]
9. Usar una herramienta de seguimiento de defectos para gestionar defectos de software en un proyecto de software pequeño [Usar]
10. Discutir las limitaciones de las pruebas en ciertos dominios [Debatir]

3.13.15. SE/Herramientas y Análisis de Pruebas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Non Core

- Herramientas y automatización de verificación y validación 1. Análisis estático 2. Cobertura de código 3. Fuzzing 4. Análisis dinámico y contención de fallos (sanitizadores, etc.) 5. Registro y seguimiento de fallos
- Planificación y generación de pruebas 1. Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos 2. Uso de números aleatorios y pseudoaleatorios en las pruebas
- Pruebas de sistemas asíncronos, paralelos y concurrentes
- Verificación y validación de artefactos que no son código (documentación, materiales de capacitación)
- Desarrollo guiado por comportamiento (BDD) 1. Expresar criterios de aceptación como escenarios Gherkin Given/When/Then 2. Ejecutar escenarios BDD como pruebas automatizadas de

extremo a extremo
- Pruebas de mutación y análisis estático de calidad de código 1. Pruebas de mutación para evaluar la efectividad de una suite de pruebas 2. Detección estática de *code smells* y complejidad ciclomática excesiva

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir y comparar diferentes herramientas para verificación y validación [Describir]
2. Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar]
3. Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar]
4. Describir enfoques para la estimación de fallos [Describir]
5. Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar]
6. Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir]
7. Crear un plan de pruebas para un segmento de código de tamaño mediano que contenga código asíncrono, concurrente y/o paralelo, incluyendo una matriz de pruebas y generación de datos y entradas de prueba [Crear]
8. Describir técnicas para la verificación y validación de artefactos que no son código [Describir]
9. Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir]
10. Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]

3.13.16. SE/Pruebas de Rendimiento y Evaluación Comparativa (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Non Core

- Pruebas de rendimiento y evaluación comparativa 1. Rendimiento (*throughput*) y latencia 2. Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) 3. Aceleración (*speedup*) y escalado (*scaling*) *a*) Ley de Amdahl *b*) Ley de Gustafson *c*) Escalado suave (*soft*) y débil (*weak*) 4. Identificar y medir figuras de mérito (*figures of merit*) 5. Cuellos de botella de rendimiento comunes *a*) Limitado por cómputo (*compute-bound*) *b*) Limitado por ancho de banda de memoria *c*) Limitado por latencia 6. Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) *a*) Estimación de la incertidumbre *b*) Intervalos de confianza 7. Análisis y presentación (gráficos, etc.) 8. Técnicas de medición de tiempo (*timing*)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir]
2. Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar]
3. Describir cuellos de botella de rendimiento comunes [Describir]
4. Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir]
5. Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar]
6. Identificar las figuras de mérito, construir y ejecutar una evaluación comparativa (benchmark) y analizar estadísticamente y visualizar los resultados para un proyecto de software pequeño [Analizar]

3.13.17. SE/Compatibilidad y Versionado de API (Core Tier2: 2 hrs)

Temas:
Core

- Ley de Hyrum / Ley de las Interfaces Implícitas
- Compatibilidad con versiones anteriores 1. La compatibilidad no es una propiedad de una sola entidad, es una propiedad de una relación. 2. La compatibilidad con versiones anteriores debe evaluarse en términos de proveedor + consumidor(es) o con un modelo bien especificado de qué formas de compatibilidad aspira/promete un proveedor.
- Control de versiones (versioning) 1. Versionado Semántico (SemVer) 2. Desarrollo basado en tronco (*trunk-based development*)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar tanto el comportamiento explícito como el implícito de una interfaz e identificar riesgos potenciales de la Ley de Hyrum [Analizar]
2. Identificar cambios que pueden considerarse ampliamente "compatibles con versiones anteriores", potencialmente con declaraciones explícitas sobre qué uso está o no soportado [Analizar]
3. Evaluar si un cambio propuesto es lo suficientemente seguro dada la metodología de control de versiones en uso para un proyecto dado [Evaluar]

3.13.18. SE/Técnicas de Refactorización (Core Tier2: 1 hr)

Temas:
Core

- Refactorización 1. Patrones de refactorización estándar (renombrar, integrar, extraer, etc.) 2. Uso de herramientas de refactorización en el IDE 3. Aplicación de herramientas de análisis estático (para identificar código que necesita refactorización, generar cambios, etc.) 4. Valor de la refactorización como remedio para la deuda técnica

Non Core

- Refactorización "a Gran Escala": técnicas cuando un cambio de refactorización es demasiado grande para comprometerlo de forma segura (proyectos grandes), o cuando es imposible sincronizar el cambio entre proveedor + todos los consumidores (múltiples repositorios, consumidores con código privado). 1. Expresar tanto las API antiguas como las nuevas para que puedan coexistir. 2. Minimizar el tamaño de los cambios de comportamiento. 3. ¿Por qué se requieren estas técnicas?. Por ejemplo, "consumidores de API que puedo ver" vs "consumidores que no puedo ver").

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Considerar las entradas de herramientas de análisis estático y/o los principios de Diseño de Software para identificar código que necesita refactorización [Analizar]
2. Refactorizar la implementación de una interfaz para mejorar el diseño, claridad, etc., con impacto mínimo/cero en los usuarios existentes [Crear]

NonCore:

3. Planificar una refactorización compleja de múltiples pasos para cambiar el comportamiento predeterminado de una API de manera segura [Planear]

3.13.19. SE/Conceptos de Confiabilidad (Core Tier2: 2 hrs)

Temas:
Core

- Concepto de confiabilidad como probabilidad de fallo o tiempo medio entre fallos, y fallas (faults) como causa de fallos (failures)
- Identificar requisitos de confiabilidad para diferentes tipos de software
- Fallos de software causados por defectos/errores (bugs), por lo que para alta confiabilidad el objetivo es tener un mínimo de defectos - inyectando menos defectos (mejor capacitación, educación, planificación) y eliminando la mayoría de los defectos inyectados (pruebas, revisión de código, etc.)
- Confiabilidad del software, confiabilidad del sistema y comportamiento de fallos
- Ciclo de inyección y eliminación de defectos, y diferentes enfoques para la eliminación de defectos

- Comparar el enfoque de "presupuesto de error" (error budget) para la confiabilidad con el enfoque "libre de errores" (error-free) e identificar dominios donde cada uno es relevante.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir cómo determinar el nivel de confiabilidad requerido por un sistema de software [Describir]
2. Explicar los problemas que existen para lograr niveles muy altos de confiabilidad [Explicar]
3. Comprender enfoques para minimizar fallas que pueden aplicarse en cada etapa del ciclo de vida del software [Explicar]

3.13.20. SE/Ingeniería de Confiabilidad (Core Tier2: 1 hr)

Temas:
Non Core

- Modelos de confiabilidad de software
- Técnicas y modelos de tolerancia a fallos de software 1. Diferencias contextuales en la tolerancia a fallos (por ejemplo, hacer fallar un sistema crítico para la aviación se evita firmemente, hacer fallar un sistema de procesamiento de datos antes de que se escriban datos corruptos en el almacenamiento es muy valioso)
- Prácticas de ingeniería de confiabilidad de software - incluyendo revisiones, pruebas, verificación de modelos práctica (practical model checking)
- Identificación de dominios de fallos dependientes e independientes, y su impacto en la confiabilidad del sistema
- Análisis basado en mediciones de la confiabilidad del software - telemetría, monitorización y alertas, paneles de control, métricas de calificación de lanzamientos, etc.

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Demostrar la capacidad de aplicar múltiples métodos para desarrollar estimaciones de confiabilidad para un sistema de software [Demostrar]
2. Identificar métodos que conducirán a la realización de una arquitectura de software que logre un nivel especificado de confiabilidad [Analizar]
3. Identificar formas de aplicar redundancia para lograr tolerancia a fallos [Analizar]
4. Identificar dependencias de punto único de fallo (single-point-of-failure, SPF) en un diseño de sistema [Analizar]

3.13.21. SE/Métodos Formales

Temas:
Non Core

- Especificación formal de interfaces 1. Especificación de pre y post condiciones 2. Lenguajes formales para escribir y analizar pre y post condiciones.
- áreas de problemas bien atendidas por los métodos formales 1. Programación libre de bloqueos (*lock-free*), condiciones de carrera de datos 2. Sistemas asíncronos y distribuidos, bloqueos (*deadlock*), bloqueos vivos (*livelock*), etc.
- Comparación con otras herramientas y técnicas para la detección de defectos 1. Pruebas 2. *Fuzzing*
- Enfoques formales para el modelado y análisis de software 1. Verificadores de modelos (model checkers) 2. Buscadores de modelos (model finders)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir el papel que las técnicas de especificación y análisis formal pueden desempeñar en el desarrollo de software complejo y comparar su uso como técnicas de validación y verificación con las pruebas [Describir]
2. Aplicar técnicas de especificación y análisis formal a diseños y programas de software de baja complejidad [Aplicar]

3. Explicar los beneficios y desventajas potenciales del uso de lenguajes de especificación formal [Explicar]

3.14. Seguridad (SEC)

La informática sustenta casi cada faceta de la infraestructura crítica moderna: transporte, comunicación, salud, educación, generación y distribución de energía, por nombrar algunas. Con los ataques desenfrenados y las violaciones de esta infraestructura, los graduados en ciencias de la computación tienen un papel importante en el diseño, implementación y operación de sistemas de software que sean robustos, seguros y que protejan la seguridad (security).

El área de conocimiento de Seguridad (SEC) se centra en desarrollar una mentalidad de seguridad en la ética general de los graduados en ciencias de la computación, de modo que la seguridad esté incorporada en todos sus productos de trabajo. Los estudiantes de informática necesitan aprender sobre las vulnerabilidades del sistema y comprender las amenazas contra los sistemas informáticos. La elección del título Seguridad fue intencional para servir como un término general de una sola palabra para esta área de conocimiento, que también incluye conceptos para apoyar la privacidad, la criptografía, los sistemas seguros, los datos seguros y el código seguro.

El área de conocimiento SEC se basa en conceptos compartidos omnipresentes en todas las demás áreas de CS2023. Identifica siete conceptos transversales de la ciberseguridad: confidencialidad, integridad, disponibilidad, evaluación de riesgos, pensamiento sistémico, pensamiento adversarial y pensamiento centrado en el humano. El séptimo concepto, pensamiento centrado en el humano, se añade a los seis conceptos transversales definidos originalmente en los Planes de Estudios de Ciberseguridad 2017 (CSEC2017). Esta adición refuerza a los estudiantes que los humanos también son un eslabón en la cadena general de seguridad. Los principios para proteger sistemas incluyen seguridad por diseño (security-by-design), privacidad por diseño (privacy-by-design), defensa en profundidad (defense-in-depth) y confianza cero (zero-trust).

Otro concepto es la noción de garantía (assurance), que es una atestación de que los mecanismos de seguridad deben cumplir con las políticas de seguridad que se han definido para los datos, procesos y sistemas. La garantía está ligada a los conceptos de verificación y validación. Las consideraciones de privacidad y seguridad de los datos se comparten con las áreas de conocimiento DM (aspectos técnicos) y SEP.

El área de conocimiento SEC se sitúa así sobre varias de las otras áreas de conocimiento de CS2023, al tiempo que incluye conceptos adicionales que no están presentes en esas áreas de conocimiento. CS2023 trata la seguridad como un componente crucial del conjunto de habilidades de cualquier graduado en CS, y las horas necesarias para la preparación en seguridad provienen de todas las otras 16 áreas de conocimiento de CS2023.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.14.1 Principios y Mentalidad de Seguridad (Pág. 135)	1	3
⭐ 3.14.2 Control de Acceso y Aplicaciones (Pág. 136)	1	3
⭐ 3.14.3 Seguridad en IA y Hardware (Pág. 136)	1	2
⭐ 3.14.4 Ética, Ley y Sociedad en Seguridad (Pág. 137)	1	2
⭐ 3.14.5 Privacidad y Seguridad por Diseño (Pág. 137)	1	1
⭐ 3.14.6 Inyección y Validación de Entrada (Pág. 138)	1	2
⭐ 3.14.7 Seguridad de Memoria y Tipos (Pág. 138)	1	1
⭐ 3.14.8 Flujo de Información y No Interferencia (Pág. 139)	1	1
⭐ 3.14.9 Análisis de Malware y Seguridad Avanzada (Pág. 139)	1	3
⭐ 3.14.10 Fundamentos de Criptografía (Pág. 140)	1	2
⭐ 3.14.11 Algoritmos Criptográficos (Pág. 140)	1	2
⭐ 3.14.12 Protocolos, Aplicaciones y Criptografía Avanzada (Pág. 140)	1	2
⭐ 3.14.13 Análisis de Amenazas e Ingeniería de Seguridad (Pág. 141)	1	4
⭐ 3.14.14 Diseño e Ingeniería de Controles de Seguridad (Pág. 142)	1	3
⭐ 3.14.15 Computación Confiable e Ingeniería de Privacidad (Pág. 142)	1	3
⭐ 3.14.16 Técnicas de Informática Forense (Pág. 142)		4
⭐ 3.14.17 Aspectos Legales y Manejo de Incidentes (Pág. 143)		3
⭐ 3.14.18 Gobierno, Gestión y Política de Seguridad (Pág. 143)		2
⭐ 3.14.19 Gestión de Riesgos y Datos (Pág. 144)		1

Cuadro 3.14: Lista de KUs del área de Seguridad.

3.14.1. SEC/Principios y Mentalidad de Seguridad (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Esta unidad de conocimiento introduce conceptos fundamentales de seguridad, principios transversales y el desarrollo de una mentalidad de seguridad. Cubre confidencialidad, integridad, disponibilidad, evaluación de riesgos, pensamiento sistémico, pensamiento adversarial y pensamiento centrado en el humano. También incluye conceptos básicos de autenticación, autorización, vulnerabilidades, amenazas, superficies de ataque, denegación de servicio, principios de protección y el impacto de la IA en la seguridad y la privacidad.

Temas:

Core

- Desarrollar una mentalidad de seguridad que incorpore conceptos transversales: confidencialidad, integridad, disponibilidad, evaluación de riesgos, pensamiento sistémico, pensamiento adversarial, pensamiento centrado en el humano.
- Vulnerabilidades, amenazas, superficies de ataque y vectores de ataque Protección y Seguridad.
- Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS) Protección y Seguridad.
- Principios y prácticas de protección, por ejemplo, privilegio mínimo, diseño abierto, valores predeterminados seguros ante fallos, defensa en profundidad y confianza cero; y cómo pueden implementarse Principios del Sistema Operativo, Protección y Seguridad, Prácticas de Codificación, Construcción a Escala y Proceso, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Consideraciones de optimización entre seguridad, privacidad, rendimiento y otros objetivos de diseño Diseño de Sistemas, Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Conceptos de confianza (trust) y confiabilidad (trustworthiness).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Evaluar un sistema para posibles ataques que pueda lanzar un adversario [Evaluar]
2. Diseñar y desarrollar enfoques para proteger un sistema de un conjunto de amenazas identificadas [Diseñar]

3. Describir cómo se puede evitar el daño a la privacidad del usuario [Describir]
4. Desarrollar un sistema que incorpore varios principios de seguridad y privacidad [Crear]

3.14.2. SEC/Control de Acceso y Aplicaciones (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Conceptos básicos de autenticación y autorización/control de acceso.
- Modelos de control de acceso (por ejemplo, discrecional, obligatorio, basado en roles y basado en atributos).
- Controles de seguridad.
- Aplicaciones de una mentalidad de seguridad: web, nube y dispositivos móviles Diseño del Sistema, Aspectos Comunes: Plataformas, APIs y Restricciones, Aspectos Comunes: Lenguajes de Plataforma y Patrones Web.
- Protección de sistemas embebidos y ciberfísicos Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores.
- Principios de seguridad usable y computación centrada en el humano Diseño de Sistemas, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar los diferentes modelos de control de acceso en términos de funcionalidad y rendimiento [Comparar]

3.14.3. SEC/Seguridad en IA y Hardware (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:
Core

- Impacto de la IA en la seguridad y la privacidad: usar la IA para reforzar las defensas y abordar las capacidades adversariales aumentadas debido a la IA Aplicaciones e Impacto Social de la IA, Aplicaciones Adicionales e Impacto Social de la IA, Sociedad, ética y Profesión, Diseño de Sistemas.
- Seguridad y confianza en sistemas de IA/aprendizaje automático, por ejemplo, adecuados para el propósito, límites operativos éticos, fuentes de conocimiento autorizadas, datos de entrenamiento verificados, pruebas de evaluación del sistema repetibles, atestación del sistema, validación/certificación independiente; consecuencias no deseadas de: efecto adverso Aplicaciones e Impacto Social de la IA, Aplicaciones Adicionales e Impacto Social de la IA, Formulación del Aprendizaje Automático y Evaluación, Aprendizaje Profundo, RL y Ética, Introducción a la IA: Conceptos y Comportamiento Inteligente, Características del Problema y Naturaleza de los Agentes, Fundamentos de Aprendizaje Automático, Aprendizaje Automático: Datos, Representaciones y Evaluación, Redes Neuronales y Ética, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Riesgos de seguridad en la construcción y operación de sistemas de IA/aprendizaje automático (por ejemplo, sesgo algorítmico, sesgo del corpus de conocimiento, sesgo del corpus de entrenamiento, violación de derechos de autor) Aplicaciones e Impacto Social de la IA, Aplicaciones Adicionales e Impacto Social de la IA, Formulación del Aprendizaje Automático y Evaluación, Aprendizaje Profundo, RL y Ética, Introducción a la IA: Conceptos y Comportamiento Inteligente, Características del Problema y Naturaleza de los Agentes, Fundamentos de Aprendizaje Automático, Aprendizaje Automático: Datos, Representaciones y Evaluación, Redes Neuronales y Ética.
- Consideraciones de hardware en seguridad, por ejemplo, principios de hardware seguro, arquitecturas de procesador seguro, aceleración criptográfica, compartimentación, interacción software-hardware Organización de Máquina a Nivel de Ensamblador, Representación de Datos a Nivel de Máquina, Rol y Propósito de los Sistemas Operativos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Mostrar cómo un adversario podría usar algoritmos de aprendizaje automático para reducir la seguridad de un sistema [Analizar]
2. Mostrar cómo un desarrollador podría mejorar la seguridad de un sistema usando algoritmos de aprendizaje automático [Analizar]
3. Describir vulnerabilidades de hardware (especialmente de la CPU) que pueden impactar al software [Describir]

3.14.4. SEC/Ética, Ley y Sociedad en Seguridad (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Esta unidad cubre los aspectos sociales, éticos, legales y profesionales de la seguridad y la privacidad, incluyendo seguridad por diseño, privacidad por diseño, hacking ético y las implicaciones de las tecnologías emergentes.

Temas:

Core

- Impactos sociales de las fallas en seguridad y privacidad Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Aplicabilidad de leyes y regulaciones sobre seguridad y privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Consideraciones éticas profesionales al diseñar sistemas seguros y mantener la privacidad; hacking ético Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Implicaciones de seguridad y privacidad de actores maliciosos de IA/aprendizaje automático, por ejemplo, identificación de deep fakes Formulación del Aprendizaje Automático y Evaluación, Aprendizaje Profundo, RL y Ética, Introducción a la IA: Conceptos y Comportamiento Inteligente, Características del Problema y Naturaleza de los Agentes, Fundamentos de Aprendizaje Automático, Aprendizaje Automático: Datos, Representaciones y Evaluación, Redes Neuronales y Ética, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Impactos sociales de los dispositivos del Internet de las Cosas (IoT) y otras tecnologías emergentes en la seguridad y la privacidad Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el impacto de una falla de seguridad en un sistema dado [Calcular]
2. Construir un sistema que se ajuste a las leyes de seguridad [Crear]
3. Aplicar un conjunto de regulaciones de privacidad para diseñar un sistema que proteja la privacidad [Aplicar]
4. Evaluar las ramificaciones legales de un sistema que no se ajusta a las leyes y regulaciones aplicables [Evaluar]

3.14.5. SEC/Privacidad y Seguridad por Diseño (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Principios y prácticas de privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Seguridad por diseño (security by design) Diseño del Sistema, Seguridad del Sistema.

- Privacidad por diseño (privacy by design) e ingeniería de la privacidad Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Construir un sistema diseñado para evitar daños a la privacidad del usuario [Crear]

3.14.6. SEC/Inyección y Validación de Entrada (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Esta unidad cubre las vulnerabilidades comunes de software, las prácticas de codificación segura, la programación defensiva y las herramientas para el análisis estático y dinámico para prevenir y mitigar fallos de seguridad.
Temas:
Core

- Vulnerabilidades y debilidades comunes.
- Inyección SQL y otros ataques de inyección.
- Técnicas de cross-site scripting y mitigaciones.
- Validación de entrada y sanitización de datos Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar problemas subyacentes en ejemplos dados de una enumeración de debilidades comunes y explicar cómo pueden evitarse [Analizar]
2. Aplicar técnicas de validación de entrada y sanitización de datos para mejorar la seguridad de un programa [Aplicar]
3. Evaluar y prevenir ataques de inyección SQL en una aplicación de base de datos [Evaluar]
4. Evaluar y prevenir ataques de cross-site scripting contra un sitio web [Evaluar]
5. Construir un programa que pruebe todos los errores de manejo de entrada [Crear]

3.14.7. SEC/Seguridad de Memoria y Tipos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Seguridad de tipos (type safety) y lenguajes seguros en tipos Teoría de Tipos Avanzada, Modelo de Ejecución y Memoria de Sistemas, Fundamentos de Sistemas de Tipos, Sistemas de Tipos: Polimorfismo y Tipado Complementario, Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Desbordamientos de búfer (buffer overflows), aplastamiento de pila (*stack smashing*) y desbordamientos de enteros Organización de Máquina a Nivel de Ensamblador, Modelo de Ejecución y Memoria de Sistemas, Protección y Seguridad.
- Problemas de seguridad debido a condiciones de carrera (race conditions) Computación Paralela y Distribuida, Evaluación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir cómo la selección de un lenguaje de programación puede impactar la seguridad del sistema que se está construyendo [Describir]
2. Reescribir un programa en un lenguaje seguro en tipos (por ejemplo, Java o Rust) originalmente escrito en un lenguaje de programación no seguro (por ejemplo, C/C++) [Rediseñar]

3. Evaluar un programa en busca de posibles ataques de desbordamiento de búfer y reescribirlo para prevenir tales ataques [Evaluar]
4. Evaluar un conjunto de programas relacionados en busca de posibles condiciones de carrera y evitar que un adversario las explote [Evaluar]

3.14.8. SEC/Flujo de Información y No Interferencia (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Principios de no interferencia (noninterference) y no deducibilidad (nondeducibility).
- Prevención de ataques de flujo de información.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Construir un sistema que esté protegido de flujos de información no autorizados [Crear]

3.14.9. SEC/Análisis de Malware y Seguridad Avanzada (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Técnicas de seguridad ofensiva como defensa.
- Técnicas de detección de malware asistidas por IA.
- Ransomware: creación, prevención y mitigación.
- Uso seguro de componentes de terceros Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Malware: variedades, creación, ingeniería inversa y defensa contra ellos Traducción y Ejecución de Lenguajes, Modelo de Ejecución y Memoria de Sistemas.
- Garantía (assurance): pruebas (incluyendo fuzzing y pruebas de penetración), verificación y validación Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Análisis estático y dinámico Análisis de Programas y Analizadores, Protección y Seguridad, Evaluación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Compiladores seguros y generación de código seguro Traducción y Ejecución de Lenguajes, Comportamiento en Tiempo de Ejecución y Sistemas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir diferentes tipos de software malicioso [Describir]
2. Explicar los riesgos del mal uso de interfaces con código de terceros y cómo usar correctamente el código de terceros [Explicar]
3. Discutir la necesidad de actualizar el software para corregir vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Debatir]
4. Aplicar herramientas estáticas y dinámicas para identificar fallos de programación [Aplicar]
5. Evaluar un sistema para detectar la existencia de malware y eliminarlo [Evaluar]
6. Implementar técnicas preventivas para reducir la ocurrencia de ransomware [Implementar]

3.14.10. SEC/Fundamentos de Criptografía (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Esta unidad introduce principios, algoritmos, protocolos criptográficos y sus aplicaciones en la protección de datos y comunicaciones, incluyendo encriptación simétrica/asimétrica, funciones hash, firmas digitales y temas emergentes como la criptografía cuántica y poscuántica.

Temas:

Core

- Diferencias entre las vistas algorítmica, aplicada y matemática de la criptografía.
- Preliminares matemáticos: aritmética modular, algoritmo euclidiano, independencia probabilística, conceptos básicos de álgebra lineal, teoría de números, campos finitos, complejidad, análisis asintótico Matemáticas Discretas, álgebra Lineal.
- Criptografía básica: criptografía de clave simétrica y de clave pública Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, Matemáticas Discretas.
- Criptosistemas clásicos, como cifrados de desplazamiento, sustitución, transposición, libros de códigos y máquinas Matemáticas Discretas.
- Principio de Kerckhoff y uso de bibliotecas evaluadas (vetted) Prácticas de Codificación, Construcción a Escala y Proceso.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar el papel de la criptografía en el soporte de la seguridad y la privacidad [Explicar]
2. Discutir los riesgos de inventar los propios métodos criptográficos [Debatir]
3. Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Debatir]
4. Implementar y criptoanalizar cifrados clásicos [Implementar]

3.14.11. SEC/Algoritmos Criptográficos (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:

Core

- Bloques de construcción criptográficos básicos, incluyendo encriptación simétrica, encriptación asimétrica, funciones hash y autenticación de mensajes Matemáticas Discretas.
- Matemáticas adicionales: primalidad, factorización y criptografía de curva elíptica Matemáticas Discretas.
- Criptosistemas de clave privada: redes de sustitución-permutación, criptoanálisis lineal, criptoanálisis diferencial, DES y AES Matemáticas Discretas, Seguridad de Red.
- Criptosistemas de clave pública: Diffie-Hellman y RSA Matemáticas Discretas.
- Integridad y autenticación de datos: funciones hash y firmas digitales Seguridad y Privacidad de Datos, Matemáticas Discretas.
- Capacidades del atacante: ataque de mensaje elegido (para firmas), ataques de cumpleaños (birthday attacks), ataques de canal lateral y ataques de inyección de fallos Seguridad de Red.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir cómo se pueden gestionar las claves criptográficas de forma segura [Describir]
2. Comparar el rendimiento en espacio y tiempo de un conjunto dado de métodos criptográficos [Comparar]
3. Discutir cómo funcionan los criptosistemas modernos de clave privada y formas de criptoanalizarlos [Debatir]
4. Discutir cómo funcionan los criptosistemas modernos de clave pública y formas de criptoanalizarlos [Debatir]
5. Comparar diferentes algoritmos criptográficos en términos de seguridad [Comparar]
6. Explicar protocolos de intercambio de claves y mostrar enfoques para reducir su fallo [Explicar]

3.14.12. SEC/Protocolos, Aplicaciones y Criptografía Avanzada (Core Tier1: 1 hr, Core Tier2: 2 hrs)

Temas:

Core

- Uso de la criptografía en aplicaciones del mundo real, por ejemplo, dinero electrónico, canales seguros entre clientes y servidores, correo electrónico seguro, autenticación de entidades, emparejamiento de dispositivos, esteganografía y sistemas de votación Procesamiento de Imágenes, Seguridad de Red.
- Protocolos criptográficos: autenticación de desafío-respuesta, protocolos de conocimiento cero, compromisos (commitment), transferencia inconsciente, cómputo seguro de dos o múltiples partes, funciones hash, compartición de secretos y aplicaciones Matemáticas Discretas.
- Criptografía cuántica; criptografía poscuántica/resistente a la cuántica Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, Matemáticas Discretas.
- Blockchain y criptomonedas Matemáticas Discretas, Comunicación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir aplicaciones reales de primitivas y protocolos criptográficos [Describir]
2. Discutir cómo funciona la criptografía cuántica y el impacto de la computación cuántica en los algoritmos criptográficos [Debatir]

3.14.13. SEC/Análisis de Amenazas e Ingeniería de Seguridad (Core Tier1: 1 hr, Core Tier2: 4 hrs)

Esta unidad cubre enfoques sistemáticos para la ingeniería de seguridad, incluyendo modelado de amenazas, análisis de riesgos, diseño de seguridad, dominios de ataque y computación confiable (trustworthy computing).
Temas:
Core

- Objetivos de la ingeniería de seguridad: construir sistemas que permanezcan confiables a pesar de errores, accidentes o adversarios maliciosos Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Análisis de problemas y análisis situacional para abordar la seguridad del sistema Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Análisis de seguridad, que cubre análisis de requisitos de seguridad; análisis de controles de seguridad; análisis de amenazas; y análisis de vulnerabilidades Análisis de Programas y Analizadores, Evaluación.
- Dominios de ataque de seguridad y superficies de ataque, por ejemplo, comunicaciones y redes, hardware, físico, ingeniería social, software y cadena de suministro Seguridad de Red.
- Modos, técnicas y tácticas de ataque de seguridad, por ejemplo, abuso de autenticación; fuerza bruta; manipulación de búfer; inyección de código; inserción de contenido; denegación de servicio; escucha; omisión de función; suplantación; ataque de integridad; intercepción; phishing; análisis de protocolo; abuso de privilegio; suplantación (spoofing); e inyección de tráfico Seguridad de Red, Protección y Seguridad, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Crear un modelo de amenazas para un sistema o diseño de sistema [Crear]
2. Aplicar análisis situacional para desarrollar soluciones seguras bajo un escenario especificado [Aplicar]
3. Identificar y mitigar vulnerabilidades y debilidades de seguridad en un sistema [Analizar]

3.14.14. SEC/Diseño e Ingeniería de Controles de Seguridad (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Análisis de compensaciones (tradeoff) de ingeniería basado en tiempo, costo, tolerancia al riesgo, aceptación de riesgo, retorno de la inversión, etc. Evaluación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Diseño e ingeniería de seguridad, incluyendo requisitos funcionales, subsistemas de seguridad, protección de información, pruebas de seguridad, evaluación de seguridad y evaluación (*assessment*) Evaluación, Pruebas de Rendimiento y Evaluación Comparativa, Ingeniería de Requisitos, Evolución y Estimación de Requisitos, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Diseño y desarrollo de sistemas ciberfísicos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar un escenario dado para análisis de compensaciones (tradeoff) para el rendimiento del sistema, evaluación de riesgos y costos [Evaluar]
2. Diseñar un conjunto de controles de seguridad técnicos, contramedidas y protecciones de información para cumplir con los requisitos de seguridad y los objetivos de seguridad de un sistema [Diseñar]
3. Evaluar la efectividad de las funciones de seguridad, controles técnicos y componentes para un sistema [Evaluar]

3.14.15. SEC/Computación Confiable e Ingeniería de Privacidad (Core Tier1: 1 hr, Core Tier2: 3 hrs)

Temas:
Core

- Objetivos de la ingeniería de la privacidad: construir sistemas que diseñen, implementen y despliegen características y controles de privacidad Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad.
- Atestación (attestation) de productos de software con respecto a su especificación y adaptabilidad Pruebas de Rendimiento y Evaluación Comparativa, Ingeniería de Requisitos, Evolución y Estimación de Requisitos, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Consideraciones para la computación confiable (trustworthy computing), por ejemplo, empaquetado resistente a la manipulación, arranque confiable (trusted boot), núcleo confiable (*trusted kernel*), raíz de confianza de hardware (hardware root of trust), firma y verificación de software, criptografía basada en hardware, virtualización y contenedores Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar y predecir comportamientos emergentes en áreas como Ciencia de Datos, IA y Aprendizaje Automático [Evaluar]

3.14.16. SEC/Técnicas de Informática Forense (Core Tier2: 4 hrs)

Esta unidad cubre principios, metodologías y herramientas para la informática forense (digital forensics), incluyendo recolección de evidencia, análisis, respuesta a incidentes y consideraciones legales.
Temas:
Core

- Principios y metodologías básicas para la informática forense.

- Diseño de sistemas para informática forense.
- Informática forense en diferentes situaciones: sistemas operativos, sistemas de archivos, informática forense de aplicaciones, informática forense web, informática forense de red, informática forense de dispositivos móviles, uso de auditoría de bases de datos Seguridad de Red.
- Ataques a la informática forense y prevención de tales ataques.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar qué es una investigación digital y cómo se puede implementar Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Explicar]
2. Diseñar e implementar software para apoyar la informática forense [Diseñar]
3. Extraer datos de un disco duro para cumplir con la ley Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Extraer]
4. Recuperar datos basados en un término de búsqueda dado desde un sistema del que se ha hecho una imagen (imaged) [Analizar]
5. Reconstruir datos y eventos a partir de un historial de aplicación, o un artefacto web, o una base de datos en la nube, o un dispositivo móvil. Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo, Arquitectura Cloud y Gestión de Datos, Frontend, Frameworks y Estándares Web [Aplicar]
6. Capturar y analizar tráfico de red. Seguridad de Red [Analizar]
7. Desarrollar enfoques para abordar los desafíos asociados con la informática forense de dispositivos móviles [Crear]
8. Aplicar herramientas de informática forense para investigar violaciones de seguridad [Aplicar]
9. Identificar y mitigar métodos anti-forenses [Analizar]

3.14.17. SEC/Aspectos Legales y Manejo de Incidentes (Core Tier2: 3 hrs)

Temas:
Core

- Procesos de manejo de incidentes.
- Reglas de la evidencia - conceptos generales y diferencias entre jurisdicciones Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.
- Problemas legales: protección y gestión de evidencia digital, cadena de custodia, reporte, actuar como testigo experto Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir los requisitos legales para el uso de datos incautados y su uso. Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Describir]
2. Describir e implementar una cadena de custodia de extremo a extremo desde la incautación inicial de evidencia digital hasta la disposición de la evidencia. Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Describir]
3. Discutir las responsabilidades y obligaciones profesionales de una persona cuando testifica como experto en informática forense Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo [Debatir]

3.14.18. SEC/Gobierno, Gestión y Política de Seguridad (Core Tier2: 2 hrs)

Esta unidad cubre el gobierno organizacional de seguridad, la gestión de riesgos, el desarrollo de políticas, la gestión del ciclo de vida de los datos y la protección de activos críticos.
Temas:
Core

- Proteger activos críticos de amenazas.

- Gobierno de seguridad (security governance): objetivos organizacionales y evaluación general de riesgos.
- Gestión de seguridad (security management): lograr y mantener niveles apropiados de confidencialidad, integridad, disponibilidad, responsabilidad (accountability), autenticidad y confiabilidad Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.
- Política de seguridad (security policy): políticas organizacionales, políticas específicas de temas, políticas específicas de sistemas.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir activos críticos y cómo pueden protegerse [Describir]
2. Diferenciar entre gobierno de seguridad, gestión y controles, dando ejemplos de cada uno [Diferenciar]
3. Describir un control técnico e implementarlo para mitigar amenazas específicas [Describir]

3.14.19. SEC/Gestión de Riesgos y Datos (Core Tier2: 1 hr)

Temas:
Core

- Enfoques para identificar y mitigar riesgos para la infraestructura informática.
- Políticas de gestión del ciclo de vida de los datos: recolección de datos, copias de seguridad y retención; almacenamiento y servicios en la nube; divulgación de violaciones (breach disclosure) Seguridad y Privacidad de Datos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar y evaluar el riesgo de que programas y aplicaciones de base de datos causen violaciones [Analizar]
2. Diseñar e implementar estrategias de copia de seguridad apropiadas que se ajusten a una política dada [Diseñar]
3. Discutir una política de divulgación de violaciones basada en requisitos legales e implementar la política [Debatir]
4. Identificar los riesgos y beneficios de la externalización a la nube [Analizar]

3.15. Sociedad, ética y la Profesión (SEP)

El Código de ética y Conducta Profesional de la ACM establece: ”Las acciones de los profesionales de la informática cambian el mundo. Para actuar de manera responsable, deben reflexionar sobre los impactos más amplios de su trabajo, apoyando consistentemente el bien público”. El Código de ética del IEEE comienza reconociendo ”la importancia de nuestras tecnologías en la afectación de la calidad de vida en todo el mundo”. El Código de ética Profesional y Conducta de la AAAI comienza con ”Las acciones de los profesionales de la informática, y en particular, los profesionales de la IA, cambian el mundo. Para actuar de manera responsable, deben reflexionar sobre los impactos más amplios de su trabajo, apoyando consistentemente el bien público”.

Si bien los problemas técnicos dominan el plan de estudios de informática, no constituyen un programa educativo completo en un contexto más amplio. Es más evidente hoy que nunca que los estudiantes también deben estar expuestos al contexto social más amplio de la informática para desarrollar una comprensión de los problemas y responsabilidades sociales, éticos, legales y profesionales críticos y relevantes. Esta necesidad de incorporar el estudio de estos problemas no técnicos en el plan de estudios de la ACM se reconoció formalmente en 1991.

Los educadores en ciencias de la computación pueden optar por impartir el material en esta área de conocimiento dentro de los contextos de cursos técnicos y teóricos tradicionales, en cursos dedicados y como parte de cursos de proyecto integrador, proyecto y práctica profesional. El material en esta área de conocimiento se cubre mejor mediante una combinación de todo lo anterior.

El Código de ética y Conducta Profesional de la ACM, el Código de ética del IEEE y el Código de ética Profesional y Conducta de la AAAI proporcionan orientación que sirve como base para la

conducta de todos los profesionales de la informática en su trabajo. El Código de la ACM enfatiza que el razonamiento ético no es un algoritmo a seguir, y se espera que los profesionales de la informática consideren cómo su trabajo impacta el bien público como consideración principal.

Un estudiante de ciencias de la computación no debe graduarse sin comprender cómo la sociedad y la ética influyen en la profesión de la informática. Tampoco debería ser posible completar un título en ciencias de la computación sin aprender cómo los profesionales de la informática influyen en la sociedad, las consideraciones éticas involucradas en dar forma a ese impacto, y el papel del estudiante convertido en graduado en estas relaciones, tanto como profesionales de la informática como miembros de la sociedad.

Área de Conocimiento (KA)	**Core Tier1**	**Core Tier2**
⭐ 3.15.1 Impacto Social e IA (Pág. 145)	2	1
⭐ 3.15.2 Equidad Digital y Accesibilidad (Pág. 146)	2	1
⭐ 3.15.3 Fundamentos y Marcos del Análisis Ético (Pág. 146)	2	1
⭐ 3.15.4 Métodos y Herramientas de Análisis Ético (Pág. 147)	1	1
⭐ 3.15.5 Valores, Naturaleza y Actualización Profesional (Pág. 147)	1	1
⭐ 3.15.6 Códigos de Ética y Responsabilidad Legal (Pág. 148)	1	1
⭐ 3.15.7 Responsabilidad Profesional y Política Pública (Pág. 148)	1	1
⭐ 3.15.8 Disidencia Ética y Contexto Cultural (Pág. 149)	1	1
⭐ 3.15.9 Ética en el Lugar de Trabajo (Pág. 149)	1	1
⭐ 3.15.10 Fundamentos de la Propiedad Intelectual (Pág. 149)	1	1
⭐ 3.15.11 Licencias, Plagio y Autoría (Pág. 150)	1	1
⭐ 3.15.12 Derechos de PI Digital y Derechos de Autor (Pág. 150)	1	1
⭐ 3.15.13 Open Source, Piratería y DRM (Pág. 151)	1	1
⭐ 3.15.14 Datos, Vigilancia y Tecnología de Privacidad (Pág. 151)	1	1
⭐ 3.15.15 Fundamentos Filosóficos y Legislación de Privacidad (Pág. 151)	1	1
⭐ 3.15.16 Libertades Civiles y Expresión (Pág. 152)	1	1
⭐ 3.15.17 Comunicación Técnica y Profesional (Pág. 152)	1	1
⭐ 3.15.18 Comunicación en Equipo y Cultura (Pág. 153)	1	1
⭐ 3.15.19 Huella Ambiental de los Sistemas Informáticos (Pág. 153)	1	1
⭐ 3.15.20 Efectos Sistémicos y Contexto Social (Pág. 154)	1	1
⭐ 3.15.21 Diseño Sostenible y Computación Ubicua (Pág. 154)	1	1
⭐ 3.15.22 Fundamentos e Historia Temprana de la Informática (Pág. 154)	1	1
⭐ 3.15.23 Historia Moderna: PC, Internet e IA (Pág. 155)	1	1
⭐ 3.15.24 Modelos y Estrategias Económicas (Pág. 155)		1
⭐ 3.15.25 Automatización, IA y Ética Económica (Pág. 156)		1
⭐ 3.15.26 Delitos y Ataques Informáticos (Pág. 156)	1	1
⭐ 3.15.27 Leyes, Políticas y Responsabilidad en Seguridad (Pág. 157)	1	1
⭐ 3.15.28 Identidad y Diversidad en la Informática (Pág. 157)	1	1
⭐ 3.15.29 Contexto Histórico e Intercultural de DEIA (Pág. 157)	1	1
⭐ 3.15.30 Inclusión y Prácticas en Informática (Pág. 158)	1	1
⭐ 3.15.31 Diseño Accesible (Pág. 158)	1	1

Cuadro 3.15: Lista de KUs del área de Sociedad, ética y la Profesión.

3.15.1. SEP/Impacto Social e IA (Core Tier1: 2 hrs, Core Tier2: 1 hr)

El Contexto Social proporciona la base para todas las demás unidades de conocimiento en SEP, particularmente ética Profesional. Cubre las implicaciones sociales de la informática en un mundo hiperconectado, el impacto de las aplicaciones informáticas en el bienestar y la seguridad individual, las consecuencias de involucrar tecnologías informáticas en la vida cívica y cómo los déficits en diversidad y accesibilidad en la informática afectan a la sociedad.

Temas:

Core

- Implicaciones sociales (por ejemplo, ideologías políticas y culturales) en un mundo hiperconectado donde las capacidades y el impacto de las redes sociales, la inteligencia artificial y la informática en general están evolucionando rápidamente.
- Impacto de las aplicaciones informáticas (por ejemplo, redes sociales, aplicaciones de inteligencia artificial) en el bienestar individual y la seguridad de todo tipo (por ejemplo, física, emocional, económica).
- Consecuencias de involucrar tecnologías informáticas, particularmente inteligencia artificial, tecnologías biométricas y sistemas de toma de decisiones algorítmicos, en la vida cívica (por ejemplo, tecnología de reconocimiento facial, etiquetas biométricas, algoritmos de distribución de recursos, software policial) y cómo la agencia y supervisión humana son cruciales.
- Crecimiento y control de internet, datos, informática e inteligencia artificial.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las diferentes formas en que la tecnología informática (redes, computación móvil, inteligencia artificial) media la interacción social a nivel personal y colectivo [Describir]
2. Interpretar el contexto social de un diseño dado y su implementación [Interpretar]
3. Analizar la eficacia de un diseño e implementación dados utilizando datos empíricos [Analizar]
4. Comprender las implicaciones del uso de la tecnología (por ejemplo, redes sociales) para diferentes identidades, culturas y comunidades [Explicar]
5. Describir el papel de internet en facilitar la comunicación entre ciudadanos, gobiernos y entre sí [Describir]
6. Analizar los efectos de la dependencia de la informática en la implementación de la democracia (por ejemplo, entrega de servicios sociales, votación electrónica) [Analizar]
7. Describir el impacto de la falta de representación apropiada de personas de poblaciones históricamente minorizadas en la profesión de la informática (por ejemplo, cultura de la industria, diversidad de productos) [Describir]

3.15.2. SEP/Equidad Digital y Accesibilidad (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Cómo los déficits en diversidad y accesibilidad en la informática afectan a la sociedad y qué pasos se pueden tomar para mejorar la equidad en la informática.
- A menudo referido como la brecha digital, diferencias en el acceso a recursos de tecnología digital y sus ramificaciones resultantes para género, clase, etnia, geografía y/o países en desarrollo, incluyendo la consideración de la responsabilidad hacia aquellos que podrían ser menos ricos, bajo amenaza o que lucharían para que sus voces sean escuchadas.
- Problemas de accesibilidad, incluyendo requisitos legales como las Pautas de Accesibilidad al Contenido Web (www.w3.org/TR/WCAG21).
- Computación consciente del contexto (context-aware computing).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar los supuestos y valores de los desarrolladores incorporados en el diseño de hardware y software, especialmente en lo que respecta a la usabilidad para poblaciones diversas, incluidos los desatendidos y aquellos con discapacidades [Analizar]
2. Discutir las implicaciones de la conciencia del contexto en sistemas de computación ubicua [Debatir]
3. Expresar cómo el acceso a internet y a las tecnologías informáticas afecta a diferentes sociedades [Explicar]
4. Identificar por qué/cómo el acceso a internet puede ser visto como un derecho humano [Analizar]

3.15.3. SEP/Fundamentos y Marcos del Análisis Ético (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Las teorías y principios éticos son los fundamentos del análisis ético porque son los puntos de vista que pueden proporcionar orientación a lo largo del camino hacia una decisión. Cada teoría enfatiza

diferentes supuestos y métodos para determinar la ética de una acción dada. Es importante que los estudiantes reconozcan que las decisiones en diferentes contextos pueden requerir diferentes teorías éticas (incluyendo combinaciones) para llegar a resultados éticamente aceptables, y lo que constituye 'aceptable' depende de una variedad de factores como el contexto cultural. Aplicar métodos para el análisis ético requiere tanto una comprensión de los principios y supuestos subyacentes que guían una herramienta dada como una conciencia del contexto social para esa decisión. Los marcos éticos tradicionales proporcionados por la filosofía occidental pueden ser útiles, pero no son inclusivos. Se debe hacer un esfuerzo para incluir perspectivas éticas decoloniales, indígenas e históricamente marginadas siempre que sea posible.

Temas:

Core

- Evitar falacias y tergiversaciones en la argumentación.
- Teorías éticas y toma de decisiones (marcos filosóficos y sociales, por ejemplo [1]).
- Reconocimiento del papel que juega la cultura en nuestra comprensión, adopción, diseño y uso de la tecnología informática.
- Por qué la ética es importante en la informática y cómo la ética es similar y diferente de las leyes y normas sociales.
- Introducción a los marcos éticos (por ejemplo, consecuencialismo como el utilitarismo, no consecuencialismo como el deber, derechos o justicia, centrado en el agente como la virtud o el feminismo, contractualismo, ética del cuidado) y su uso para analizar un dilema ético.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir cómo un contexto cultural dado impacta la toma de decisiones [Describir]
2. Expresar el uso de ejemplos y analogías en la argumentación ética [Analizar]
3. Analizar (y evitar) falacias lógicas básicas en un argumento [Analizar]
4. Analizar un argumento para identificar premisas y conclusión [Analizar]
5. Evaluar cómo y por qué la ética es tan importante en la informática y cómo se relaciona con las normas culturales, valores y leyes [Evaluar]
6. Justificar una decisión tomada sobre bases éticas [Justificar]
7. Discutir las ventajas y desventajas de los marcos éticos tradicionales [Debatir]
8. Analizar dilemas éticos relacionados con la creación y uso de tecnología desde múltiples perspectivas utilizando marcos éticos [Analizar]

3.15.4. SEP/Métodos y Herramientas de Análisis Ético (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Listas de verificación (checklists) profesionales.
- Rúbricas de evaluación.
- Análisis de partes interesadas (stakeholder analysis).
- Teoría del punto de vista (standpoint theory).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Distinguir todas las posiciones de las partes interesadas en relación con su contexto cultural en una situación dada [Distinguir]
2. Analizar el potencial de introducir o perpetuar deuda ética (consideración diferida de impactos o implicaciones éticas) en decisiones técnicas [Analizar]

3.15.5. SEP/Valores, Naturaleza y Actualización Profesional (Core Tier1: 1 hr, Core Tier2: 1 hr)

Los profesionales de la informática tienen responsabilidades que van más allá de sus habilidades técnicas. Esta unidad explora los valores comunitarios y legales que guían la profesión, la naturaleza del profesionalismo —incluyendo el cuidado, la disciplina y la tutoría— y la responsabilidad del profesional de mantenerse actualizado en un campo en constante evolución.

Temas:
Core
- Valores de la comunidad y las leyes por las que vivimos.
- La naturaleza de ser un profesional, incluyendo cuidado, atención, disciplina, responsabilidad fiduciaria y tutoría (mentoring).
- Mantenerse actualizado como profesional de la informática en términos de familiaridad, herramientas, habilidades, marcos legales y profesionales, así como la capacidad y responsabilidad de autoevaluarse y progresar en el campo de la informática.

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Identificar problemas éticos que surgen en el diseño de software, prácticas de desarrollo y despliegue de software [Analizar]
2. Discutir cómo abordar problemas éticos en situaciones específicas [Debatir]
3. Describir los mecanismos que típicamente existen para que un profesional se mantenga actualizado en asuntos éticos [Describir]

3.15.6. SEP/Códigos de Ética y Responsabilidad Legal (Core Tier1: 1 hr, Core Tier2: 1 hr)

Los códigos de ética profesional, como los de la ACM, IEEE y AAAI, definen estándares de conducta y responsabilidad en la informática. Esta unidad examina la certificación profesional, los códigos de ética relevantes, y las dimensiones legales y éticas de la responsabilidad del software, incluyendo la corrección, confiabilidad y seguridad.

Temas:
Core
- Certificación profesional, códigos de ética, conducta y práctica, como los de la ACM, IEEE, AAAI y otras sociedades internacionales.
- Responsabilidad (accountability), obligación (responsibility) y responsabilidad legal (liability) (por ejemplo, corrección del software, confiabilidad y seguridad, garantía, negligencia, responsabilidad estricta, enfoques éticos para la divulgación de vulnerabilidades de seguridad), incluyendo si un producto/servicio debe construirse, no solo hacerlo porque es técnicamente posible.

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Expresar la responsabilidad ética de garantizar la corrección, confiabilidad y seguridad del software, incluyendo de dónde surge esta responsabilidad (por ejemplo, Códigos de ética de ACM/IEEE/AAAI, leyes y regulaciones, políticas organizacionales) [Analizar]
2. Describir las fortalezas y debilidades de los códigos profesionales relevantes como expresiones de ser un profesional y guías para la toma de decisiones [Describir]

3.15.7. SEP/Responsabilidad Profesional y Política Pública (Core Tier1: 1 hr, Core Tier2: 1 hr)

Los profesionales de la informática operan dentro de marcos sociales, legales y culturales complejos. Esta unidad examina las teorías que explican cómo los seres humanos crean y utilizan la tecnología —incluyendo enfoques decoloniales y de justicia social— y explora las responsabilidades del profesional para reconocer, reportar y mantener conciencia de las consecuencias éticas de su trabajo y su participación en la política pública.

Temas:
Core
- Introducción a teorías que describen la creación y uso humano de la tecnología, incluyendo instrumentalismo, sociología de los sistemas tecnológicos, justicia de la discapacidad (disability justice), tesis de la neutralidad, pragmatismo y modelos decoloniales, incluyendo desarrollar y usar tecnología para corregir errores y hacer el bien.
- Estrategias para reconocer y reportar diseños, sistemas, software y conducta profesional (o sus resultados) que puedan violar la ley o códigos profesionales de ética.
- El papel del profesional de la informática y las sociedades profesionales en la política pública.

- Mantener la conciencia de las consecuencias.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar un problema global de informática, observando el papel de los profesionales y funcionarios gubernamentales en la gestión de este problema [Analizar]
2. Describir formas en que los profesionales y las organizaciones profesionales pueden contribuir a la política pública [Describir]
3. Describir las consecuencias del comportamiento profesional inapropiado [Describir]

3.15.8. SEP/Disidencia Ética y Contexto Cultural (Core Tier1: 1 hr, Core Tier2: 1 hr)

La disidencia ética es un componente crítico de la conducta profesional responsable. Esta unidad aborda el whistleblowing como mecanismo de rendición de cuentas, y explora cómo el contexto cultural regional moldea la comprensión y respuesta ante dilemas éticos en la profesión informática.

Temas:
Core

- Disidencia ética y denuncia de irregularidades (whistleblowing).
- La relación entre la cultura regional y los dilemas éticos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Estar familiarizado con la denuncia de irregularidades (*whistleblowing*) y tener acceso a conocimiento para guiarlo a través de un incidente [Evaluar (valorar)]
2. Identificar ejemplos de cómo la cultura regional interactúa con dilemas éticos [Analizar]

3.15.9. SEP/Ética en el Lugar de Trabajo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Tratar con acoso y discriminación.
- Formas de acreditación profesional.
- Ergonomía y entornos informáticos saludables.
- Tiempo de comercialización (time-to-market) y consideraciones de costo versus estándares profesionales de calidad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir formas de acoso y discriminación y vías de asistencia [Describir]
2. Evaluar varias formas de acreditación profesional [Evaluar (valorar)]
3. Establecer la relación entre la ergonomía en los entornos informáticos y la salud de las personas [Enunciar/Establecer]
4. Describir problemas asociados con el impulso de las industrias para centrarse en el tiempo de comercialización (time-to-market) versus la aplicación de estándares profesionales de calidad [Describir]

3.15.10. SEP/Fundamentos de la Propiedad Intelectual (Core Tier1: 1 hr, Core Tier2: 1 hr)

La propiedad intelectual se refiere a un rango de derechos intangibles de propiedad sobre cualquier producto del intelecto humano, como un programa de software. Las leyes, que varían según la localidad, proporcionan diferentes métodos para proteger estos derechos de propiedad basados en su tipo. Idealmente, las leyes de propiedad intelectual equilibran los intereses de los creadores y usuarios de la propiedad. Hay cuatro tipos de derechos de propiedad intelectual relevantes para el software: patentes, derechos de autor (copyrights), secretos comerciales y marcas registradas. Además, los derechos de propiedad a menudo están protegidos por licencias de usuario. Cada uno proporciona un tipo diferente de protección legal.

Temas:
Core

- Derechos de propiedad intelectual.
- Fundamentos legales para la protección de la propiedad intelectual.
- Fundamentos filosóficos de la propiedad intelectual.
- Formas de propiedad intelectual (por ejemplo, derechos de autor, patentes, secretos comerciales, marcas registradas) y los derechos que protegen.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Defender los usos legales y éticos de materiales con derechos de autor [Defender]
2. Discutir las bases filosóficas de la propiedad intelectual en un contexto apropiado (por ejemplo, país) [Debatir]
3. Contrastar las protecciones y obligaciones de los derechos de autor, patentes, secretos comerciales y marcas registradas [Contrastar]
4. Describir la justificación para la protección legal de la propiedad intelectual en el contexto apropiado (por ejemplo, país) [Describir]

3.15.11. SEP/Licencias, Plagio y Autoría (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Licencias de software comunes (por ejemplo, MIT, GPL y sus variantes, Apache, Mozilla, Creative Commons).
- Plagio y autoría.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Seleccionar una licencia de software apropiada para un proyecto dado [Analizar]
2. Interpretar la intención y implementación de las licencias de software [Interpretar]
3. Analizar los problemas éticos inherentes en varios mecanismos de detección de plagio [Analizar]
4. Identificar múltiples formas de plagio más allá de la copia textual de texto o software (por ejemplo, paráfrasis intencional, tergiversación de autoría y atribución inadecuada) [Analizar]

3.15.12. SEP/Derechos de PI Digital y Derechos de Autor (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Propiedad intelectual digital intangible (IDIP).
- Limitaciones a las protecciones de derechos de autor, incluyendo uso justo (fair use) y la doctrina de la primera venta (first sale doctrine).
- Leyes y tratados de propiedad intelectual que impactan la aplicación de los derechos de autor.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir y criticar la legislación y los precedentes destinados a las infracciones de derechos de autor digitales [Describir]
2. Identificar ejemplos contemporáneos de propiedad intelectual digital intangible [Analizar]
3. Discutir si un uso de material con derechos de autor es probable que sea uso justo (fair use) [Debatir]
4. Analizar el uso de obras con derechos de autor bajo los conceptos de uso justo (fair use) y la doctrina de la primera venta (first sale doctrine) [Analizar]
5. Discutir la naturaleza de las leyes anti-elusión (anti-circumvention) en el contexto de la protección de derechos de autor [Debatir]

3.15.13. SEP/Open Source, Piratería y DRM (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Piratería de software y métodos técnicos para hacer cumplir los derechos de propiedad intelectual, como la gestión de derechos digitales (digital rights management) y el software de código cerrado como secreto comercial.
- Fundamentos morales y legales del movimiento de código abierto.
- Sistemas que usan datos de otros (por ejemplo, modelos de lenguaje grande).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Distinguir los problemas conflictivos involucrados en la obtención de patentes de software [Distinguir]
2. Identificar los objetivos del movimiento de código abierto y su impacto en campos más allá de la informática, como el movimiento del derecho a reparar (right-to-repair) [Analizar]
3. Resumir la naturaleza global de la piratería de software [Resumir]
4. Criticar el uso de medidas técnicas de gestión de derechos digitales (por ejemplo, encriptación, marcas de agua, restricciones de copia y bloqueos regionales) desde las perspectivas de múltiples partes interesadas [Criticar]

3.15.14. SEP/Datos, Vigilancia y Tecnología de Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

La recopilación masiva de datos en sistemas transaccionales, plataformas de vigilancia, computación en la nube e inteligencia artificial plantea desafíos fundamentales para la privacidad. Esta unidad examina las implicaciones de estos sistemas, las concepciones de anonimato y seudonimidad, y las soluciones tecnológicas para proteger la privacidad, como la encriptación de extremo a extremo y la privacidad diferencial.

Temas:
Core

- Implicaciones de privacidad de la recopilación generalizada de datos, incluyendo pero no limitado a bases de datos transaccionales, almacenes de datos, sistemas de vigilancia, computación en la nube e inteligencia artificial.
- Concepciones de anonimato, seudonimidad e identidad.
- Soluciones basadas en tecnología para la protección de la privacidad (por ejemplo, encriptación de extremo a extremo y privacidad diferencial).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar soluciones a las amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Evaluar]
2. Describir el papel de la recopilación de datos en la implementación de sistemas de vigilancia generalizada (por ejemplo, RFID, reconocimiento facial, cobro de peajes, computación móvil) [Describir]
3. Distinguir los conceptos y objetivos del anonimato y la seudonimidad [Distinguir]
4. Describir las ramificaciones de las protecciones de privacidad basadas en tecnología, incluyendo la privacidad diferencial y la encriptación de extremo a extremo [Describir]

3.15.15. SEP/Fundamentos Filosóficos y Legislación de Privacidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

El derecho a la privacidad tiene raíces filosóficas y legales profundas que varían según el contexto cultural y jurídico. Esta unidad examina las concepciones filosóficas y legales de la privacidad, los fundamentos legales internacionales como el GDPR, y la legislación sectorial como HIPAA y la Ley de IA de la UE, evaluando críticamente su intención, valor e implementación.

Temas:
Core

- Concepciones filosóficas y legales de la naturaleza de la privacidad, incluyendo el derecho a la privacidad.
- Fundamentos legales de la protección de la privacidad en jurisdicciones relevantes (por ejemplo, GDPR en la UE).
- Legislación de privacidad en áreas de práctica (por ejemplo, HIPAA en EE.UU., Ley de IA en la UE).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir la base filosófica para la protección legal de la privacidad personal en un contexto apropiado (por ejemplo, país) [Debatir]
2. Criticar la intención, valor potencial e implementación de varias formas de legislación de privacidad y principios más allá de lo que la ley requiere [Críticar (análisis crítico)]

3.15.16. SEP/Libertades Civiles y Expresión (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Libertades civiles, derechos de privacidad y diferencias culturales.
- Principios básicos de la investigación con sujetos humanos y principios más allá de lo que la ley requiere (por ejemplo, Informe Belmont, Declaración Universal de Derechos Humanos de la ONU y cómo esto se relaciona con la tecnología).
- Libertad de expresión y sus limitaciones.
- Contenido generado por usuarios, moderación de contenido y responsabilidad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar diferencias culturales con respecto a la naturaleza y necesidad de la privacidad y otras libertades civiles [Analizar]
2. Identificar estrategias para permitir una libertad de expresión apropiada [Analizar]

3.15.17. SEP/Comunicación Técnica y Profesional (Core Tier1: 1 hr, Core Tier2: 1 hr)

La informática es una disciplina inherentemente colaborativa y social, lo que hace de la comunicación un aspecto esencial de la profesión. Gran parte, pero no toda, de esta comunicación ocurre en un entorno profesional donde los estilos, expectativas y normas de comunicación difieren de otros contextos donde se podría usar tecnología similar. Tanto la comunicación profesional como la informal transmiten información a varias audiencias que pueden tener diferentes objetivos y necesidades para esa información. Una buena comunicación también es necesaria para la transparencia y la confiabilidad. También es importante señalar que los profesionales de la informática no solo son comunicadores sino también oyentes que deben poder escuchar y utilizar reflexivamente los comentarios recibidos de diversas partes interesadas. Las habilidades de comunicación efectiva no son algo que uno 'simplemente sabe': se desarrollan y pueden aprenderse. Las habilidades de comunicación se enseñan mejor en contexto a lo largo del plan de estudios de pregrado.

Temas:
Core

- Comunicación oral, escrita y electrónica en equipo y grupo.
- Materiales de comunicación técnica (por ejemplo, código fuente y documentación, tutoriales, materiales de referencia, documentación de API).
- Comunicarse con diferentes partes interesadas, como clientes, liderazgo o el público.
- Compensaciones (tradeoffs) en factores competitivos que afectan los canales y elecciones de comunicación.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comprender la importancia de escribir documentos técnicos concisos y precisos siguiendo estándares bien definidos para el formato y para incluir tablas, figuras y referencias apropiadas [Producir]

2. Analizar documentación técnica escrita para verificar precisión técnica, concisión, falta de ambigüedad y conciencia de la audiencia [Analizar]
3. Componer y entregar una presentación formal consciente de la audiencia, accesible y organizada [Componer]
4. Identificar y describir cualidades de la comunicación efectiva (por ejemplo, virtual, cara a cara, intragrupo, documentos compartidos) [Analizar]
5. Elegir una forma apropiada de comunicar preocupaciones éticas delicadas [Evaluar (valorar)]

3.15.18. SEP/Comunicación en Equipo y Cultura (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Colaboración en equipo (incluyendo herramientas) y resolución de conflictos.
- Requisitos de accesibilidad e inclusividad para dirigirse a audiencias profesionales.
- Competencia cultural en la comunicación, incluyendo considerar el impacto de la diferencia en el lenguaje natural.
- Comunicarse para resolver problemas o hacer recomendaciones en el lugar de trabajo, como plantear preocupaciones éticas o abordar problemas de accesibilidad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Planificar interacciones (por ejemplo, virtuales, cara a cara, documentos compartidos) con otros de maneras que inviten a la participación inclusiva, modelen la consideración respetuosa de las contribuciones de otros y valoren explícitamente la diversidad de ideas [Planear]
2. Comprender cómo comunicarse efectiva y apropiadamente como miembro de un equipo, incluyendo técnicas de resolución de conflictos [Aplicar]
3. Discutir formas de influir en el desempeño y los resultados en equipos diversos y multiculturales [Debatir]
4. Evaluar fortalezas y debilidades personales para trabajar de forma remota como parte de un equipo que proviene de diversos antecedentes y experiencias [Evaluar (valorar)]

3.15.19. SEP/Huella Ambiental de los Sistemas Informáticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

El ciclo de vida del hardware y software tiene costos ambientales significativos. Esta unidad examina los impactos ambientales y sociales de los sistemas informáticos —desde residuos electrónicos y huella de carbono hasta el consumo energético de centros de datos y modelos de IA— evaluando su efecto en las decisiones de diseño y en la sociedad global.

Temas:
Core

- Impactos ambientales, sociales y culturales de las decisiones de implementación (por ejemplo, objetivos de sostenibilidad, sesgo/resultados algorítmicos, viabilidad económica y consumo de recursos).
- Impactos sociales y ambientales locales/regionales/globales de los sistemas informáticos y su uso (por ejemplo, huella de carbono, uso de recursos, residuos electrónicos) debido al hardware (por ejemplo, residuos electrónicos, centros de datos, utilización de elementos y recursos raros, reciclaje) y software (por ejemplo, servicios basados en la nube, blockchain, entrenamiento y uso de modelos de IA). Esto incluye el uso diario de hardware (hardware barato reemplazado con frecuencia) y software (navegación web, correo electrónico y otros servicios con demandas computacionales ocultas/remotas).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar los impactos ambientales del despliegue de un proyecto dado (por ejemplo, consumo de energía, contribución a los residuos electrónicos, impacto de la fabricación) [Evaluar (valorar)]
2. Describir los impactos sociales y ambientales globales del uso y disposición de computadoras [Describir]

3. Describir los impactos ambientales de las decisiones de diseño dentro del campo de la informática que se relacionan con el diseño de algoritmos, diseño de sistemas operativos, diseño de redes, diseño de bases de datos, etc [Describir]
4. Evaluar aplicaciones informáticas en relación con problemas ambientales (por ejemplo, energía, contaminación, uso de recursos, reciclaje y reutilización, gestión y producción de alimentos) [Evaluar (valorar)]

3.15.20. SEP/Efectos Sistémicos y Contexto Social (Core Tier1: 1 hr, Core Tier2: 1 hr)

Las tecnologías informáticas complejas como la IA generativa, las redes sociales y el trabajo remoto generan efectos sistémicos difíciles de predecir. Esta unidad analiza cómo la sostenibilidad del software depende de los sistemas sociales que lo rodean —usuarios, organizaciones, mercados y políticas— y cómo los diseños tecnológicos transforman a su vez esos sistemas.

Temas:

Core

- Efectos sistémicos de tecnologías y fenómenos informáticos complejos (por ejemplo, IA generativa, centros de datos, redes sociales, externalización (offshoring), trabajo remoto).
- Cómo la sostenibilidad de los sistemas de software es interdependiente con los sistemas sociales, incluyendo el conocimiento y las habilidades de sus usuarios, los procesos y políticas organizacionales y su contexto social (por ejemplo, fuerzas del mercado, políticas gubernamentales).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enumerar los efectos sostenibles de las prácticas y actividades modernas (por ejemplo, trabajo remoto, comercio electrónico, criptomonedas, modelos de IA, centros de datos) [Listar/Enumerar]
2. Analizar los impactos sociales y ambientales de nuevos diseños de sistemas [Analizar]

3.15.21. SEP/Diseño Sostenible y Computación Ubicua (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Pautas para estándares de diseño sostenible.
- Computación ubicua (pervasive computing): procesamiento de información que se ha integrado en objetos y actividades cotidianas, como sistemas inteligentes de energía, redes sociales y sistemas de retroalimentación para promover comportamientos sostenibles, transporte, monitoreo ambiental, ciencia ciudadana y activismo.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Identificar formas de ser un profesional sostenible en un área específica o con un proyecto específico [Analizar]
2. Diseñar pautas para el diseño o despliegue de TI sostenible en áreas como sistemas inteligentes de energía, redes sociales, transporte, agricultura, sistemas de cadena de suministro, monitoreo ambiental y activismo ciudadano [Diseñar]

3.15.22. SEP/Fundamentos e Historia Temprana de la Informática (Core Tier1: 1 hr, Core Tier2: 1 hr)

La historia es importante porque proporciona un mecanismo para comprender por qué nuestros sistemas informáticos operan como lo hacen, los contextos sociales en los que surgieron los enfoques actuales y cómo estos continúan resonando en la disciplina hoy. No solo la informática afecta a la sociedad, sino también viceversa, resultando en un contexto socio-técnico complejo que está en constante cambio, requiriendo la perspectiva de la historia para poner el presente, así como los futuros posibles, en la perspectiva apropiada. También informa decisiones basadas en éxitos y fracasos del pasado, incluyendo el daño causado y cómo no repetirlos. La historia de la informática a menudo se

enseña en contexto con conceptos fundamentales, como fundamentos de sistemas y fundamentos de desarrollo de software. Se debe poner un enfoque en aquellos que, debido a la marginación, no han figurado históricamente tan prominentemente como deberían.

Temas:

Core

- La historia de la informática: hardware, software y humano/organizacional.
- El papel de la historia en el presente, incluyendo dentro de diferentes contextos sociales, y la relevancia de esta historia en el futuro.
- Edad I (Pre-digital): Computación analógica antigua (Stonehenge, mecanismo de Anticitera, reloj de la Catedral de Salisbury, etc.), tablas numéricas calculadas por humanos, Euclides, Lovelace, Babbage, Gödel, Church, Turing, hardware pre-electrónico (electromecánico y mecánico).
- Edad II (Computación moderna temprana): ENIAC, UNIVAC, Bombas (Bletchley Park y descifradores de códigos), compañías informáticas (por ejemplo, IBM), mainframes, etc.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comprender la relevancia e impacto de la historia de la informática en eventos recientes, contexto presente y posibles resultados futuros, desde más de una perspectiva cultural [Evaluar (valorar)]
2. Discutir cómo las perspectivas sostenidas hoy han sido moldeadas por la historia, y que existen perspectivas alternativas (por ejemplo, temores de que la IA reemplace a los trabajadores humanos vs IA que aumenta el trabajo humano, varias visiones sobre la Ley de Moore) [Debatir]
3. Identificar tendencias formativas y consecuentes en la historia del campo de la informática [Analizar]
4. Identificar las contribuciones de individuos u organizaciones pioneras (laboratorios de investigación, compañías informáticas, oficinas gubernamentales) en el campo de la informática [Analizar]

3.15.23. SEP/Historia Moderna: PC, Internet e IA (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Edad III (Era de las PC): PCs, hardware y software informático moderno, Ley de Moore.
- Edad IV (Internet): Redes, arquitectura de internet, navegadores y su evolución, estándares, compañías y servicios nacidos en internet (por ejemplo, Google, Amazon, Microsoft, etc.), computación distribuida.
- Edad V (Móvil y Nube): Computación móvil y teléfonos inteligentes, computación en la nube y sus modelos (por ejemplo, SaaS), servidores remotos, seguridad y privacidad, redes sociales.
- Edad VI (IA): Sistemas de toma de decisiones, sistemas de recomendación, IA generativa y otras herramientas y tecnologías impulsadas por aprendizaje automático.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Discutir el contexto histórico para momentos importantes en la historia de la informática, como el cambio de tubos de vacío a transistores (TRADIC), sistemas operativos seminales tempranos (por ejemplo, OS 360), Xerox PARC y la primera computadora Apple con una GUI, la creación de paradigmas de lenguaje de programación específicos, el primer virus informático, la creación de internet, la creación de la WWW, la burbuja puntocom, Y2K, la introducción de teléfonos inteligentes, etc [Debatir]
2. Comparar la vida diaria antes y después del advenimiento de desarrollos históricos (por ejemplo, computadoras personales o internet) [Comparar]

3.15.24. SEP/Modelos y Estrategias Económicas (Core Tier2: 1 hr)

Las economías de la informática son importantes para aquellos que desarrollan y proporcionan recursos y servicios informáticos a otros, así como para la sociedad en general. Impactan a los usuarios de estos recursos y servicios, tanto profesionales como no profesionales. Los profesionales de la informática tienen el deber de conocer el impacto de estos temas en sus propios roles y actividades y

cómo las elecciones hechas impactarán a los usuarios y a la sociedad.

Temas:

Core

- Modelos económicos: regulados y no regulados, monopolios, efectos de red (network effects) y mercado abierto; economías del conocimiento y de la atención (knowledge and attention economies).
- Estrategias de precios y despliegue: obsolescencia programada (planned obsolescence), suscripciones, freemium, licencias de software, código abierto (open-source), software libre (free software), adware.
- Impactos de las diferencias en el acceso a recursos informáticos y el efecto de la oferta y demanda de mano de obra calificada en la calidad de los productos informáticos.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Resumir los efectos sociales de los modelos económicos (por ejemplo, las economías del conocimiento y de la atención) [Resumir]
2. Describir las diferencias y similitudes de estrategias competitivas (por ejemplo, suscripción vs freemium vs gratis) [Describir]
3. Discutir ejemplos de brechas digitales [Debatir]

3.15.25. SEP/Automatización, IA y Ética Económica (Core Tier2: 1 hr)

Temas:

Core

- Automatización, IA y sus efectos en los mercados laborales, desarrolladores y usuarios.
- Preocupaciones éticas en torno a la economía de la atención (attention economy) y otras economías de la informática (por ejemplo, consentimiento informado, recopilación de datos, uso de jerga legal extensa en acuerdos de usuario).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comprender los efectos de la automatización y la IA en la sociedad [Evaluar (valorar)]
2. Comprender las implicaciones éticas de las economías informáticas que dependen de la atención y los datos [Evaluar (valorar)]

3.15.26. SEP/Delitos y Ataques Informáticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Si bien las políticas de seguridad, las leyes y los delitos informáticos son temas importantes, es esencial que se vean con la base de otras unidades de conocimiento social y profesional, como Propiedad Intelectual, Privacidad y Libertades Civiles, Contexto Social y ética Profesional. Las computadoras, internet y la inteligencia artificial, quizás más que cualquier otra tecnología, han transformado la sociedad en los últimos 75 años. Al mismo tiempo, estas tecnologías han contribuido a amenazas sin precedentes a la privacidad; nuevas categorías de delitos informáticos y comportamientos antisociales; grandes interrupciones a organizaciones; y la concentración a gran escala de riesgo en los sistemas de información.

Temas:

Core

- Delitos informáticos, recurso legal para criminales informáticos e impacto en víctimas y sociedad.
- Ingeniería social, fraude habilitado por la informática, robo de identidad y recuperación de estos.
- Terrorismo cibernético, hacking criminal y hacktivismo.
- Malware, virus, gusanos.
- Ataques a infraestructura crítica como redes eléctricas y oleoductos.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enumerar ejemplos clásicos de delitos informáticos e incidentes de ingeniería social con impacto social [Listar/Enumerar]

2. Identificar problemas con las leyes que se aplican a los delitos informáticos [Analizar]
3. Describir la motivación y ramificaciones del terrorismo cibernético, robo de datos, hacktivismo, ransomware y otros ataques [Describir]
4. Examinar los problemas éticos y legales que rodean el mal uso del acceso y varias violaciones de seguridad [Examinar]

3.15.27. SEP/Leyes, Políticas y Responsabilidad en Seguridad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Fundamentos no técnicos de la seguridad (por ejemplo, ingeniería humana, política, confidencialidad).
- Beneficios y desafíos de las leyes de delitos informáticos existentes y propuestas.
- Políticas de seguridad y los desafíos del cambio y el cumplimiento.
- Responsabilidad por la seguridad a lo largo del ciclo de vida de la informática.
- Leyes internacionales y locales y cómo se cruzan.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Discutir el papel del profesional en la seguridad y las compensaciones y desafíos involucrados [Debatir]
2. Investigar medidas que pueden tomar tanto individuos como organizaciones, incluidos gobiernos, para prevenir o mitigar los efectos indeseables de los delitos informáticos [Investigar]
3. Diseñar una política de seguridad para toda la compañía, que incluya procedimientos para la gestión de contraseñas y monitoreo de empleados [Diseñar]
4. Comprender cómo la legislación de una región puede afectar las actividades en otra (por ejemplo, cómo el GDPR de la UE se aplica globalmente, cuando hay personas de la UE involucradas) [Evaluar (valorar)]

3.15.28. SEP/Identidad y Diversidad en la Informática (Core Tier1: 1 hr, Core Tier2: 1 hr)

La identidad —incluyendo raza, género, origen y capacidad— moldea tanto la experiencia de los profesionales de la informática como los productos que crean. Esta unidad examina cómo la identidad interactúa con las tecnologías y entornos informáticos, y los beneficios demostrables de los equipos de desarrollo diversos, así como los costos de la falta de diversidad.

Temas:
Core

- Cómo la identidad impacta y es impactada por las tecnologías y entornos informáticos (académicos y profesionales).
- Los beneficios de los equipos de desarrollo diversos y los impactos de los equipos que no son diversos.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir y distinguir equidad, igualdad, diversidad, inclusión y accesibilidad [Definir]
2. Identificar ejemplos de los beneficios que los equipos diversos pueden aportar a los productos de software y cómo la falta de diversidad tiene costos [Analizar]
3. Evaluar el impacto del poder y el privilegio en la profesión de la informática en lo que respecta a la cultura, industria, productos y sociedad [Evaluar (valorar)]

3.15.29. SEP/Contexto Histórico e Intercultural de DEIA (Core Tier1: 1 hr, Core Tier2: 1 hr)

Las desigualdades actuales en la informática tienen raíces históricas profundas en mecanismos sociales sistémicos. Esta unidad examina la marginación histórica en la informática, los desafíos de

infraestructura global, y las diferencias interculturales en DEIA, explorando cómo los cambios sistémicos pueden mejorar la equidad tanto en contextos familiares como desconocidos.

Temas:

Core

- Marginación histórica debido a mecanismos sociales sistémicos, supremacía tecnológica y desafíos de infraestructura global para la diversidad, equidad, inclusión y accesibilidad.
- Diferencias y necesidades interculturales en diversidad, equidad, inclusión y accesibilidad.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comparar la demografía de los estudiantes de ciencias de la computación y STEM de su institución con la demografía institucional general. Si difieren, identificar factores que contribuyen al acceso, compromiso y logro inequitativo en ciencias de la computación entre grupos marginados. Si no, evaluar por qué no [Comparar]
2. Desarrollar ejemplos de cambios sistémicos que podrían abordar positivamente la diversidad, equidad, inclusión y accesibilidad en un contexto familiar (es decir, en un curso introductorio de informática) y en un contexto no familiar, y cuándo estos podrían ser diferentes o iguales [Crear]
3. Comparar la demografía de su institución con la demografía general de la comunidad. Si difieren, identificar factores que contribuyen al acceso, compromiso y logro inequitativo entre grupos marginados. Si no, evaluar por qué no [Comparar]

3.15.30. SEP/Inclusión y Prácticas en Informática (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Lenguaje inclusivo y terminología cargada, y por qué su uso importa.
- Comportamientos inclusivos y por qué importan.
- Cómo los profesionales de la informática pueden influir e impactar la diversidad, equidad, inclusión y accesibilidad, incluyendo pero no solo a través del software que crean.
- Expertos y sus prácticas que reflejan las identidades del aula y del mundo a través de principios prácticos de DEIA.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Identificar lenguaje, prácticas y comportamientos que pueden hacer que alguien se sienta incluido en un lugar de trabajo y/o equipo, y por qué es relevante. Evitar terminología cargada (véase Words Matter, www.acm.org/diversity-inclusion/words-matter); esto incluye identificar y acomodar a usuarios que a menudo son excluidos sin pensarlo y no son considerados en absoluto [Analizar]
2. Demostrar colegialidad y respeto cuando se trabaja con miembros del equipo que no comparten su identidad. No es suficiente simplemente asignar proyectos en equipo. La facultad debe preparar a los estudiantes para el trabajo en equipo y monitorear, guiar y evaluar la efectividad de sus equipos estudiantiles a lo largo de un proyecto [Demostrar]
3. Identificar los supuestos y valores de los desarrolladores incorporados en el diseño de hardware y software, especialmente en lo que respecta a la usabilidad por parte de poblaciones diversas [Analizar]
4. Analizar el trabajo de expertos que reflejan las identidades del aula y del mundo [Analizar]

3.15.31. SEP/Diseño Accesible (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:

Core

- Diseñar y desarrollar tecnología con la accesibilidad en mente.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Evaluar la accesibilidad de su aula o laboratorio. Evaluar la accesibilidad de su página web. (Véase www.w3.org/WAI y www.w3.org/TR/WCAG21) [Evaluar]

3.16. Fundamentos de Sistemas (SF)

Un sistema informático es un conjunto de infraestructuras de hardware y software sobre las cuales se construyen aplicaciones. Los sistemas informáticos se han convertido en un pilar de la vida diaria de las personas. Como tal, es esencial que los estudiantes aprendan conocimientos sobre sistemas informáticos, adquieran las habilidades para usar y diseñar estos sistemas y comprendan la lógica y los principios fundamentales en los sistemas informáticos. Esto podría equipar a los estudiantes con la competencia necesaria para una carrera relacionada con las ciencias de la computación.

En el plan de estudios de Ciencias de la Computación, el estudio de los sistemas informáticos generalmente abarca múltiples áreas de conocimiento, incluyendo, entre otros, sistemas operativos, sistemas paralelos y distribuidos, redes de comunicaciones, arquitectura y organización de computadoras, e ingeniería de software. El área de conocimiento de Fundamentos de Sistemas, como sugiere su nombre, se centra en los conceptos fundamentales y principios de diseño en sistemas informáticos que son compartidos por estos cursos dentro de sus respectivos núcleos. El objetivo de esta área de conocimiento es presentar una visión integradora de estos conceptos fundamentales y principios de diseño de una manera unificada aunque simplificada, proporcionando una base común para los diferentes mecanismos y políticas especializados apropiados para el área de dominio específica. Los conceptos fundamentales en esta área de conocimiento incluyen una visión general de los sistemas informáticos, conceptos básicos como estado y transición de estado, asignación y planificación de recursos, y así sucesivamente. Además, esta área de conocimiento introduce principios básicos de diseño para mejorar la confiabilidad, disponibilidad, eficiencia y seguridad de los sistemas informáticos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.16.1 Componentes Hardware y Lógica (Pág. 159)	2	
⭐ 3.16.2 Capas, Abstracciones y Escalabilidad (Pág. 160)	2	
⭐ 3.16.3 Sistemas Digitales y Lógica (Pág. 160)	2	
⭐ 3.16.4 Máquinas de Estado (Pág. 161)	2	
⭐ 3.16.5 Procesamiento Secuencial y Paralelo (Pág. 161)	2	
⭐ 3.16.6 Gestión de Recursos (Pág. 162)	1	1
⭐ 3.16.7 Latencia, Caché y Jerarquía de Memoria (Pág. 162)	2	2
⭐ 3.16.8 Virtualización y Aislamiento (Pág. 162)	1	1
⭐ 3.16.9 Métricas y Benchmarks de Rendimiento (Pág. 163)	1	1
⭐ 3.16.10 Modelos Analíticos de Rendimiento (Pág. 163)	1	1
⭐ 3.16.11 Eventos, Herramientas y Experimentación (Pág. 164)	1	1
⭐ 3.16.12 Confiabilidad del Sistema (Pág. 164)	2	1
⭐ 3.16.13 Seguridad del Sistema (Pág. 165)	2	1
⭐ 3.16.14 Diseño del Sistema (Pág. 165)	2	1
3.16.15 Sociedad, ética y la Profesión (Pág. 165)	Electivo	

Cuadro 3.16: Lista de KUs del área de Fundamentos de Sistemas.

3.16.1. SF/Componentes Hardware y Lógica (Core Tier1: 2 hrs)

Introduce los bloques de construcción y componentes básicos de una computadora, el hardware como paradigma computacional y el concepto de segmentación (pipelining).

Temas:

Core

- Bloques de construcción y componentes básicos de una computadora (compuertas, flip-flops, registros, interconexiones; ruta de datos (datapath) + control + memoria).
- Hardware como paradigma computacional: Bloques de construcción lógicos fundamentales; Expresiones lógicas, minimización, formas de suma de productos Lógica Digital y Sistemas Digitales.

- Concepto básico de segmentación (pipelining), etapas de procesamiento superpuestas Organización Funcional.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir los bloques de construcción básicos de las computadoras y su papel en el desarrollo histórico de la arquitectura de computadoras [Describir]
2. Diseñar un circuito lógico simple usando los bloques de construcción fundamentales del diseño lógico para resolver un problema simple (por ejemplo, un sumador) [Diseñar]

3.16.2. SF/Capas, Abstracciones y Escalabilidad (Core Tier1: 2 hrs)

Cubre las abstracciones de programación, la distinción entre servicios de aplicación y del sistema operativo, el concepto de escalado y la organización en capas de los sistemas informáticos.
Temas:
Core

- Abstracciones de programación, interfaces, uso de bibliotecas Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades.
- Distinción e interacción entre servicios de aplicación y del sistema operativo, llamada a procedimiento remoto (*remote procedure call*) Rol y Propósito de los Sistemas Operativos.
- Concepto básico de escalado (scaling): rendimiento vs tamaño del problema.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir cómo los sistemas informáticos se construyen de capa sobre capa, basados en la separación de preocupaciones, con interfaces bien definidas, ocultando detalles de las capas bajas a las capas más altas [Describir]
2. Describir que el hardware, el sistema operativo, la máquina virtual y la aplicación son capas adicionales de interpretación/procesamiento [Describir]
3. Describir los mecanismos de cómo los errores se detectan, se señalan hacia atrás y se manejan a través de las capas [Describir]
4. Construir un programa simple (por ejemplo, un cliente/servidor TCP) usando métodos de capas, detección y recuperación de errores, y reflejo del estado de error a través de las capas [Crear]
5. Identificar errores en un programa por capas usando herramientas para seguimiento de programas, ejecución paso a paso (single stepping) y depuración [Analizar]
6. Comprender el concepto de escalado fuerte (strong) vs débil (weak), es decir, cómo el rendimiento se ve afectado por la escala del problema vs la escala de los recursos para resolver el problema. Esto puede motivarse con ejemplos simples del mundo real [Evaluar (valorar)]

3.16.3. SF/Sistemas Digitales y Lógica (Core Tier1: 2 hrs)

Cubre sistemas digitales vs analógicos, compuertas lógicas, expresiones booleanas, relojes y lógica combinacional y secuencial, como base para comprender el funcionamiento de los sistemas informáticos.
Temas:
Core

- Sistemas Digitales vs Analógicos/Discretos vs Continuos.
- Compuertas lógicas simples, expresiones lógicas, simplificación de lógica booleana.
- Relojes, Estado, Secuenciación.
- Lógica Combinacional, Lógica Secuencial, Registros, Memorias Lógica Digital y Sistemas Digitales.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las diferencias entre sistemas digitales y analógicos, y entre sistemas discretos y continuos. Puede dar ejemplos del mundo real de estos sistemas [Describir]

3.16.4. SF/Máquinas de Estado (Core Tier1: 2 hrs)

Introduce el concepto de estado y transición de estado, máquinas de estados finitos (FSM), y su aplicación en computadoras y protocolos de red como ejemplos de máquinas de estado.

Temas:

Core

- Estado y transición de estado (por ejemplo, estado inicial, estado final, ciclo de vida de estados) Lenguajes Formales y Autómatas.
- Máquinas de estados finitos (por ejemplo, NFA, DFA) Lenguajes Formales y Autómatas.
- Computadoras y Protocolos de Red como ejemplos de Máquinas de Estado Fundamentos de Redes y Comunicaciones.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir las computaciones como un sistema caracterizado por un conjunto conocido de configuraciones con transiciones de una configuración única (estado) a otra (estado) [Describir]
2. Describir la distinción entre sistemas cuya salida es solo una función de su entrada (sin estado - stateless) y aquellos con memoria/historial (con estado - stateful) [Describir]
3. Desarrollar descripciones de máquinas de estado para soluciones de enunciados de problemas simples (por ejemplo, secuenciación de semáforos, reconocedores de patrones) [Crear]
4. Describir una computadora como una máquina de estado que interpreta instrucciones de máquina [Describir]
5. Explicar cómo un programa o protocolo de red también puede expresarse como una máquina de estado y que pueden existir representaciones alternativas para la misma computación [Explicar]
6. Derivar el comportamiento de serie temporal de una máquina de estado a partir de su representación de máquina de estado (por ejemplo, máquina de estado de gestión de conexión TCP) [Analizar]

3.16.5. SF/Procesamiento Secuencial y Paralelo (Core Tier1: 2 hrs)

Cubre el procesamiento secuencial y paralelo, incluyendo modelos de un solo hilo y múltiples hilos a nivel de aplicación, y la evaluación del rendimiento de versiones paralelas de programas.

Temas:

Core

- Procesamiento secuencial vs paralelo. Concurrencia, Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades.
- Procesamiento secuencial a nivel de aplicación: hilo único (single thread) Concurrencia, Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades.
- Procesamiento paralelo simple a nivel de aplicación: a nivel de solicitud (servicios web/cliente-servidor/distribuido), hilo único por servidor, múltiples hilos con múltiples servidores, segmentación (pipelining) Concurrencia, Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Escribir un problema secuencial simple y una versión paralela simple del mismo programa [Escribir]
2. Evaluar el rendimiento de versiones secuenciales y paralelas simples de un programa con diferentes tamaños de problema y ser capaz de describir las aceleraciones (speed-ups) logradas [Evaluar]
3. Describir en una línea de tiempo de ejecución cómo los eventos y operaciones de paralelismo pueden tener lugar simultáneamente (es decir, al mismo tiempo). Explicar cómo el trabajo puede realizarse en menos tiempo transcurrido si esto puede explotarse [Describir]

3.16.6. SF/Gestión de Recursos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Esta unidad de conocimiento cubre diferentes tipos de recursos en sistemas informáticos y algoritmos comunes de asignación y planificación de recursos.

Temas:

Core

- Diferentes tipos de recursos (por ejemplo, participación del procesador, memoria, disco, ancho de banda de red) Fundamentos de Redes y Comunicaciones, API de Sistemas de Archivos e Implementación, Gestión de Memoria, Planificación.
- Algoritmos comunes de asignación/planificación de recursos (por ejemplo, primero en llegar primero en ser servido, planificación basada en prioridad, planificación justa (fair scheduling) y planificación con desalojo (preemptive scheduling)) Planificación.
- Ventajas y desventajas de los algoritmos de planificación comunes Planificación.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Definir cómo los recursos finitos de computadora (por ejemplo, participación del procesador, memoria, almacenamiento y ancho de banda de red) se gestionan mediante su cuidadosa asignación a las entidades existentes [Definir]
2. Describir cómo funcionan los algoritmos comunes de asignación/planificación de recursos [Describir]
3. Desarrollar algoritmos de planificación comunes y evaluar sus rendimientos [Crear]
4. Describir los pros y los contras de los algoritmos de planificación comunes [Describir]

3.16.7. SF/Latencia, Caché y Jerarquía de Memoria (Core Tier1: 2 hrs, Core Tier2: 2 hrs)

Cubre las latencias en los sistemas informáticos, la localidad espacial y temporal en las memorias caché, la coherencia de caché, y la jerarquía de memoria del procesador con su fórmula de tiempo promedio de acceso.

Temas:

Core

- Latencias en sistemas informáticos: 1. Velocidad de la luz y las computadoras (un pie por nanosegundo vs relojes de un GHz) Organización Funcional 2. Latencias de memoria vs disco vs memoria a través de la red Jerarquía de Memoria, Rendimiento y Eficiencia Energética
- Memorias caché y los efectos de la localidad espacial y temporal en el rendimiento en procesadores y sistemas Jerarquía de Memoria, Rendimiento y Eficiencia Energética, Gestión de Memoria.
- Memorias caché y coherencia de caché en bases de datos, sistemas operativos, sistemas distribuidos y arquitectura de computadoras Jerarquía de Memoria, Internos de DBMS, Gestión de Memoria.
- Introducción a la jerarquía de memoria del procesador Jerarquía de Memoria, Rendimiento y Eficiencia Energética.
- La fórmula del tiempo promedio de acceso a memoria Jerarquía de Memoria.

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir el desglose de la latencia de los sistemas informáticos en términos de memoria, disco y red [Describir]
2. Explicar la importancia de la localidad en la determinación del rendimiento del sistema [Explicar]
3. Calcular el tiempo promedio de acceso a memoria y describir las compensaciones (tradeoffs) en el rendimiento de la jerarquía de memoria en términos de capacidad, tasa de fallos/aciertos (miss/hit rate) y tiempo de acceso [Calcular]

3.16.8. SF/Virtualización y Aislamiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cubre la justificación del aislamiento y la virtualización para la protección y el rendimiento predecible, los niveles de indirección mediante memoria virtual, y los métodos para implementar memoria virtual y máquinas virtuales.

Temas:
Core
- Justificación de la virtualización y el aislamiento: protección y rendimiento predecible Virtualización.
- Niveles de indirección, ilustrados por la memoria virtual para gestionar recursos de memoria física Virtualización.
- Métodos para implementar memoria virtual y máquinas virtuales Virtualización.

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Explicar por qué es importante aislar y proteger la ejecución de programas individuales y entornos que comparten recursos subyacentes comunes [Explicar]
2. Describir cómo el concepto de indirección puede crear la ilusión de una máquina dedicada y sus recursos incluso cuando se comparten físicamente entre múltiples programas y entornos [Describir]
3. Evaluar el rendimiento de dos instancias de aplicación ejecutándose en máquinas virtuales separadas y determinar el efecto del aislamiento de rendimiento [Evaluar]

3.16.9. SF/Métricas y Benchmarks de Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cubre las figuras de mérito de rendimiento, las cargas de trabajo y benchmarks representativos, y los métodos de recolección y análisis para la evaluación cuantitativa del rendimiento de los sistemas informáticos.
Temas:
Core
- Figuras de mérito de rendimiento Rendimiento y Eficiencia Energética, Evaluación.
- Cargas de trabajo y puntos de referencia representativos (benchmarks), y métodos de recolección y análisis de figuras de mérito de rendimiento Rendimiento y Eficiencia Energética, Evaluación.

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Explicar cómo los componentes de la arquitectura del sistema contribuyen a mejorar su rendimiento [Explicar]
2. Explicar las circunstancias en las que una figura dada de una métrica de rendimiento del sistema es útil [Explicar]
3. Explicar el uso e insuficiencias de los puntos de referencia (benchmarks) como medida del rendimiento del sistema [Explicar]

3.16.10. SF/Modelos Analíticos de Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cubre la ecuación CPI como herramienta de diseño, la Ley de Amdahl y sus limitaciones, el análisis de orden de magnitud (Big O), el análisis de caminos críticos del sistema y las herramientas analíticas para la evaluación cuantitativa.
Temas:
Core
- Ecuación CPI (Ciclos por Instrucción) como herramienta para comprender las compensaciones (tradeoffs) en el diseño de conjuntos de instrucciones, tuberías de procesador (pipelines) y organizaciones del sistema de memoria Rendimiento y Eficiencia Energética, Evaluación.
- Ley de Amdahl: la parte de la computación que no se puede acelerar limita el efecto de las partes que sí se pueden Rendimiento y Eficiencia Energética, Evaluación.
- Análisis de orden de magnitud (notación Big O) Notación Asintótica y Clases de Complejidad, Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas, Marco de Análisis de Complejidad, Teoría de la Complejidad Computacional.
- Análisis de caminos lentos y rápidos de un sistema Organización Funcional.
- Herramientas analíticas para guiar la evaluación cuantitativa.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la Ley de Amdahl y discutir sus limitaciones [Describir]
2. Aplicar estudios de límite o cálculos simples para producir estimaciones de orden de magnitud para una métrica de rendimiento dada en un contexto dado [Aplicar]

3.16.11. SF/Eventos, Herramientas y Experimentación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cubre los eventos y su efecto en el rendimiento (detenciones, fallos de caché, fallos de página), la comprensión de sistemas por capas como plataformas de evaluación, las trampas de los microbenchmarks, y las herramientas y experimentos para medir el rendimiento del sistema.
Temas:
Core

- Eventos y su efecto en el rendimiento (por ejemplo, detenciones de instrucciones (stalls), fallos de caché, fallos de página) Organización Funcional, Gestión de Memoria.
- Comprender sistemas por capas, cargas de trabajo y plataformas, sus implicaciones para el rendimiento y los desafíos que representan para la evaluación.
- Trampas de los micro puntos de referencia (microbenchmark pitfalls).

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar herramientas de software para perfilar y medir el rendimiento del programa [Aplicar]
2. Diseñar y conducir un experimento orientado al rendimiento de un sistema común (por ejemplo, un sistema operativo y Spark) [Diseñar]
3. Diseñar un experimento de rendimiento en un sistema por capas para determinar el efecto de un parámetro del sistema en el rendimiento del sistema [Diseñar]

3.16.12. SF/Confiabilidad del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Esta unidad de conocimiento cubre la distinción entre errores (bugs), fallas (faults) y fallos (failures); confiabilidad vs disponibilidad; y confiabilidad mediante redundancia, incluyendo verificación y reintento, codificación redundante y duplicación.
Temas:
Core

- Distinción entre errores (bugs), fallas (faults) y fallos (failures) Coordinación, Conceptos de Confiabilidad, Ingeniería de Confiabilidad.
- Confiabilidad vs disponibilidad.
- Confiabilidad mediante redundancia: 1. verificación y reintento (*check and retry*) Soporte de Fiabilidad, Tolerancia a fallos 2. codificación redundante (códigos de corrección de errores, CRC, FEC, RAID) Jerarquía de Memoria, Bases de Datos Distribuidas/Computación en la Nube, Soporte de Fiabilidad 3. duplicación/reflejo (*mirroring*)/réplicas Bases de Datos Distribuidas/Computación en la Nube
- Otros enfoques para la confiabilidad (por ejemplo, registro por diario - journaling) Soporte de Fiabilidad, Tolerancia a fallos, Conceptos de Confiabilidad, Ingeniería de Confiabilidad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la distinción entre errores de programa, errores del sistema y fallas de hardware (por ejemplo, memoria corrupta) y excepciones (por ejemplo, intento de dividir por cero) [Explicar]
2. Articular la distinción entre detectar, manejar y recuperarse de fallas y los métodos para su implementación [Articular]
3. Describir el papel de los códigos de corrección de errores en proporcionar técnicas de verificación y corrección de errores en memorias, almacenamiento y redes [Describir]
4. Aplicar algoritmos simples para explotar información redundante con el propósito de corrección de datos [Aplicar]
5. Comparar diferentes métodos de detección y corrección de errores por su sobrecarga de datos, complejidad de implementación y tiempo de ejecución relativo para codificar, detectar y corregir

errores [Comparar]

3.16.13. SF/Seguridad del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Esta unidad de conocimiento cubre problemas comunes de seguridad del sistema y contramedidas, incluyendo criptografía y arquitectura de seguridad, así como sistemas de contramedida representativos, como sistemas de detección de intrusiones y sistemas antivirus.

Temas:

Core

- Problemas comunes de seguridad del sistema (por ejemplo, virus, ataques de denegación de servicio y escuchas - eavesdropping) Seguridad de Red, Protección y Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad, Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad.
- Contramedidas Seguridad de Red, Principios del Sistema Operativo, Protección y Seguridad: 1. Criptografía Protocolos, Aplicaciones y Criptografía Avanzada, Algoritmos Criptográficos, Fundamentos de Criptografía 2. Arquitectura de seguridad Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad
- Sistemas de contramedida representativos: 1. Sistemas de detección de intrusiones, cortafuegos (firewalls) Seguridad de Red 2. Sistemas antivirus

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir algunos problemas comunes de seguridad del sistema y dar ejemplos [Describir]
2. Describir algunas contramedidas contra problemas de seguridad del sistema [Describir]
3. Describir sistemas de contramedida representativos [Describir]

3.16.14. SF/Diseño del Sistema (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Esta unidad de conocimiento cubre criterios comunes de diseño de sistemas (por ejemplo, vivacidad (liveness), seguridad (safety), robustez, escalabilidad y seguridad (security)) y los diseños de sistemas representativos.

Temas:

Core

- Criterios comunes de diseño de sistemas (por ejemplo, vivacidad (liveness), seguridad (safety), robustez, escalabilidad y seguridad (security)) Evaluación.
- Diseños de sistemas representativos (por ejemplo, servidor web Apache, Spark y Linux).

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir criterios comunes de diseño de sistemas [Describir]
2. Dada la funcionalidad requerida de un sistema y sus criterios de diseño clave, proporcionar un diseño de alto nivel de este sistema [Generar]
3. Describir el diseño de algunos sistemas representativos [Describir]

3.16.15. SF/Sociedad, ética y la Profesión

Esta unidad de conocimiento cubre los derechos de propiedad intelectual de los sistemas informáticos, licencias de software comunes y delitos informáticos.

Temas:

Core

- Derechos de propiedad intelectual de los sistemas informáticos Open Source, Piratería y DRM, Derechos de PI Digital y Derechos de Autor, Fundamentos de la Propiedad Intelectual, Licencias, Plagio y Autoría.
- Licencias de software comunes Open Source, Piratería y DRM, Derechos de PI Digital y Derechos de Autor, Fundamentos de la Propiedad Intelectual, Licencias, Plagio y Autoría.

- Delitos informáticos Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir los derechos de propiedad intelectual de los sistemas informáticos [Describir]
2. Enumerar licencias de software representativas y comparar sus diferencias [Listar/Enumerar]
3. Enumerar delitos informáticos representativos [Listar/Enumerar]

3.17. Desarrollo de Plataformas Especializadas (SPD)

El área de conocimiento de Desarrollo de Plataformas Especializadas (SPD) se refiere a los atributos que involucran la creación de software dirigido a plataformas de hardware no tradicionales. El desarrollo para cada plataforma especializada, por ejemplo, robots, sistemas móviles, sistemas basados en web y sistemas embebidos, típicamente involucra consideraciones únicas.

Las necesidades sociales e industriales han creado una alta demanda de desarrolladores en plataformas especializadas, como aplicaciones móviles, plataformas web, plataformas robóticas y sistemas embebidos. Algunas habilidades profesionales únicas relevantes para esta área de conocimiento incluyen las siguientes:

- Crear aplicaciones que proporcionen una experiencia de usuario consistente en varios dispositivos, tamaños de pantalla y sistemas operativos.
- Desarrollar interfaces de programación de aplicaciones (APIs) para soportar la funcionalidad de cada plataforma especializada.
- Gestionar desafíos relacionados con limitaciones de recursos como cómputo, memoria, almacenamiento, y redes y comunicaciones.
- Aplicar preocupaciones transversales como optimización, seguridad, mejores prácticas de desarrollo, etc.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 3.17.1 Aspectos Comunes: Plataformas, APIs y Restricciones (Pág. 167)	3	2
⭐ 3.17.2 Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (Pág. 168)	2	1
⭐ 3.17.3 Frontend, Frameworks y Estándares Web (Pág. 168)		3
⭐ 3.17.4 Arquitectura Cloud y Gestión de Datos (Pág. 168)		3
⭐ 3.17.5 Plataformas Móviles: Fundamentos del Desarrollo (Pág. 169)		2
⭐ 3.17.6 Plataformas Móviles: Arquitectura y Restricciones (Pág. 169)		2
⭐ 3.17.7 Plataformas Móviles: Capacidades y Computación Asíncrona (Pág. 169)		1
⭐ 3.17.8 Plataformas Robóticas (Pág. 170)		4
⭐ 3.17.9 Características y Arquitecturas Embebidas (Pág. 170)		1
⭐ 3.17.10 Sistemas, Programación y Sensores (Pág. 171)		2
⭐ 3.17.11 Tiempo Real, Memoria y Verificación (Pág. 171)		3
⭐ 3.17.12 Historia, Tipos y Características de Plataformas (Pág. 171)		1
⭐ 3.17.13 Sensores, Controladores y Ecosistemas eSports (Pág. 172)		1
⭐ 3.17.14 Arquitecturas y Renderizado (Pág. 172)		1
⭐ 3.17.15 Algoritmos, Medios y Mundos Virtuales (Pág. 173)		1
⭐ 3.17.16 Lenguajes, APIs y Herramientas de Desarrollo (Pág. 173)		1
⭐ 3.17.17 Motores de Juego (Pág. 173)		1
⭐ 3.17.18 Fundamentos del Diseño de Juegos (Pág. 174)		1
⭐ 3.17.19 Géneros, Audiencias y Alcance Social (Pág. 174)		1
⭐ 3.17.20 Plataformas Interactivas de Datos y Análisis (Pág. 175)		2
⭐ 3.17.21 Programación con IA y Plataformas Cuánticas (Pág. 175)		1
3.17.22 SPD-SEP/Móvil (Pág. 175)	Electivo	
3.17.23 SPD-SEP Web (Pág. 176)	Electivo	
3.17.24 Propiedad Intelectual en los Juegos (Pág. 176)	Electivo	
3.17.25 Accesibilidad e Interfaz del Juego (Pág. 177)	Electivo	
3.17.26 Salud, Seguridad y Sostenibilidad (Pág. 177)	Electivo	
3.17.27 Representación, Inclusión e Igualdad (Pág. 178)	Electivo	
3.17.28 SPD-SEP/Robótica (Pág. 178)	Electivo	
3.17.29 SPD-SEP/Interactiva (Pág. 178)	Electivo	

Cuadro 3.17: Lista de KUs del área de Desarrollo de Plataformas Especializadas.

3.17.1. SPD/Aspectos Comunes: Plataformas, APIs y Restricciones (Core Tier1: 3 hrs, Core Tier2: 2 hrs)

Temas:
Core

- Visión general de las plataformas de desarrollo (es decir, web, móvil, juegos, robótica, embebida e interactiva). 1. Entrada/sensores/dispositivos de control/dispositivos hápticos 2. Limitaciones de recursos *a*) Computacional *b*) Almacenamiento de datos *c*) Memoria *d*) Comunicación 3. Requisitos - seguridad, disponibilidad de tiempo de actividad, tolerancia a fallos Conceptos de Confiabilidad, Ingeniería de Confiabilidad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad 4. Salida/actuadores/dispositivos hápticos
- Programación mediante Interfaces de Programación de Aplicaciones (API) específicas de la plataforma vs construcción de aplicaciones tradicional
- Programación bajo restricciones y requisitos de la plataforma (por ejemplo, herramientas de desarrollo disponibles, desarrollo, consideraciones de seguridad) Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar las restricciones de la programación móvil [Listar/Enumerar]

3.17.2. SPD/Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (Core Tier1: 2 hrs, Core Tier2: 1 hr)

Temas:
Core

- Visión general de los Lenguajes de plataforma (por ejemplo, Python, Swift, Lua, Kotlin)
- Técnicas para aprender y dominar un lenguaje de programación específico de la plataforma

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar las características de los lenguajes de scripting [Listar/Enumerar]
2. Describir el modelo de tres capas (three-tier) de la programación web [Describir]
3. Describir cómo se mantiene el estado en la programación web [Describir]

3.17.3. SPD/Frontend, Frameworks y Estándares Web (Core Tier2: 3 hrs)

Cubre los lenguajes de programación web, los frameworks, el modelo SaaS, los estándares web (DOM, accesibilidad) y las consideraciones de seguridad y privacidad.
Temas:
Core

- Lenguajes de programación web (por ejemplo, HTML5, JavaScript, Typescript, CSS)
- Plataformas, frameworks o meta-frameworks web 1. Servicios en la nube 2. API, Componentes Web (Web Components)
- Software como Servicio (SaaS)
- Estándares web como el modelo de objetos del documento (DOM), accesibilidad Accesibilidad y Diseño Inclusivo
- Consideraciones de Seguridad y Privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar y contrastar la programación web con la programación de propósito general [Comparar]
2. Describir las diferencias entre Software-como-Servicio (SaaS) y productos de software tradicionales [Describir]
3. Discutir cómo los estándares web impactan el desarrollo de software [Debatir]
4. Evaluar una aplicación web existente contra los estándares web actuales [Evaluar]

3.17.4. SPD/Arquitectura Cloud y Gestión de Datos (Core Tier2: 3 hrs)

Cubre el análisis de requisitos para aplicaciones web, los servicios de computación en la nube, la gestión de datos, las arquitecturas (microservicios, serverless, event-driven) y las soluciones de almacenamiento.
Temas:
Non Core

- Análisis de requisitos para aplicaciones web
- Servicios de computación Sistemas NoSQL 1. Alojamiento en la Nube (Cloud Hosting) 2. Escalabilidad (por ejemplo, Autoescalado (*Autoscaling*), Clústeres) 3. Estimación de costos para servicios
- Gestión de datos Conceptos Fundamentales de Sistemas de Bases de Datos 1. Residencia de datos (data residency): dónde se ubican los datos y qué rutas se pueden tomar para acceder a ellos 2. Integridad de datos (data integrity): garantizar que los datos sean accesibles y que se eliminen cuando sea requerido
- Arquitectura 1. Monolitos vs Microservicios 2. Micro-frontends 3. Arquitecturas Dirigidas por Eventos (Event-Driven) vs RESTful: ventajas y desventajas 4. Sin servidor (Serverless), computación en la nube bajo demanda
- Soluciones de almacenamiento Sistemas NoSQL, Bases de Datos Relacionales 1. Bases de datos relacionales 2. Bases de datos NoSQL

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar e implementar una aplicación basada en web utilizando un diseño de arquitectura de microservicios [Diseñar]
2. Describir las restricciones, como alojamiento, servicios y escalabilidad, relacionadas con las plataformas web [Describir]

3.17.5. SPD/Plataformas Móviles: Fundamentos del Desarrollo (Core Tier2: 2 hrs)

Temas:
Core

- Desarrollo con 1. Lenguajes de programación móvil 2. Entornos de programación móvil
- Restricciones de la plataforma móvil 1. Diseño de interfaz de usuario Comprender al Usuario: Objetivos individuales e interacciones con otros 2. Seguridad
- Acceso 1. Acceso a datos a través de APIs Construcción de Consultas 2. Diseño de puntos finales de API (endpoints) para aplicaciones móviles: trampas y consideraciones de diseño 3. Interfaces de red y web Modelado de Datos, Fundamentos de Redes y Comunicaciones

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar la programación móvil con la programación de propósito general [Comparar]
2. Evaluar los pros y los contras del desarrollo de aplicaciones móviles nativas y multiplataforma [Evaluar]

3.17.6. SPD/Plataformas Móviles: Arquitectura y Restricciones (Core Tier2: 2 hrs)

Temas:
Non Core

- Desarrollo 1. Desarrollo nativo (native) versus multiplataforma (cross-platform) 2. Patrones de diseño/arquitectura de software para aplicaciones móviles Diseño de API y Modelado, Principios y Arquitectura de Software, Calidad y Evaluación del Diseño
- Restricciones de la plataforma móvil 1. Diseño de interfaz de usuario responsivo (responsive) Accesibilidad y Diseño Inclusivo 2. Heterogeneidad y movilidad de los dispositivos 3. Diferencias en las experiencias de usuario (por ejemplo, entre aplicaciones móviles y basadas en web) 4. Compensación (tradeoff) entre energía y rendimiento
- Especificación y pruebas Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Diseñar la arquitectura de una aplicación móvil aplicando patrones de diseño de software apropiados, justificando la elección entre desarrollo nativo y multiplataforma [Diseñar]
2. Evaluar las restricciones de una plataforma móvil dada (interfaz responsiva, heterogeneidad de dispositivos, compensación energía-rendimiento) al especificar y probar una aplicación [Evaluar]

3.17.7. SPD/Plataformas Móviles: Capacidades y Computación Asíncrona (Core Tier2: 1 hr)

Temas:
Non Core

- Capacidades (*affordances*) de la computación móvil 1. Aplicaciones conscientes de la ubicación (*location-aware*) 2. Computación impulsada por sensores (por ejemplo, giroscopio, acelerómetro, datos de salud de un reloj) 3. Telefonía y mensajería instantánea 4. Realidad aumentada Inmersión
- Computación asíncrona Algoritmos 1. Diferencia con la programación síncrona tradicional 2. Manejo del éxito mediante devoluciones de llamada (*callbacks*) 3. Manejo de errores de forma asíncrona 4. Pruebas de código asíncrono y problemas típicos en las pruebas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear]
2. Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear]
3. Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]

3.17.8. SPD/Plataformas Robóticas (Core Tier2: 4 hrs)

Temas:
Core

- Tipos de plataformas y dispositivos robóticos Robótica
- Sensores, cómputo embebido y efectores (actuadores) Computación Tangible/Física
- Lenguajes y bibliotecas específicos de robótica Robótica
- Arquitectura de software robótico (por ejemplo, usando el Sistema Operativo de Robot (ROS - *Robot Operating System*))
- Restricciones y consideraciones de diseño de la plataforma robótica Robótica
- Interconexiones con sistemas físicos o simulados Simulación, Computación Tangible/Física
- Algoritmos Robóticos Robótica, Animación por Computadora 1. Cinemática directa (*forward kinematics*) 2. Cinemática inversa (*inverse kinematics*) 3. Dinámica 4. Navegación y planificación de rutas (*path planning*) 5. Agarre (*grasping*) y manipulación
- Consideraciones de seguridad e interacción Equidad Digital y Accesibilidad, Impacto Social e IA, Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar e implementar una aplicación en una plataforma robótica dada [Diseñar]
2. Integrar un kit de robot basado en Arduino y programarlo para navegar por un laberinto [Integrar]
3. Comparar lenguajes y técnicas específicos de robótica con aquellos utilizados para el desarrollo de software de propósito general [Comparar]
4. Explicar la lógica detrás del diseño de la plataforma robótica y sus interconexiones con sistemas físicos o simulados [Explicar]
5. Dada una aplicación de alto nivel, diseñar una arquitectura de software de robot usando ROS especificando todos los componentes e interconexiones (temas ROS - ROS topics) para lograr esa aplicación [Diseñar]
6. Discutir las restricciones que una plataforma robótica dada impone a los desarrolladores [Debatir]

3.17.9. SPD/Características y Arquitecturas Embebidas (Core Tier2: 1 hr)

Cubre las características únicas de los sistemas embebidos frente a otros sistemas, las APIs para arquitecturas personalizadas (GPU, FPGA) y la evaluación de arquitecturas según requisitos.

Temas:
Core

- Introducción a las características únicas de los sistemas embebidos 1. Sistemas de tiempo real (*real-time*) vs tiempo real suave (*soft real-time*) y no tiempo real 2. Restricciones de recursos, como perfiles de memoria y plazos (*deadlines*) Jerarquía de Memoria
- API para arquitecturas personalizadas 1. Tecnología GPU Arquitecturas Heterogéneas, Sombreado y Renderizado Avanzado 2. Arreglos de Compuertas Programables en Campo (FPGA - *Field Programmable Gate Arrays*) Lógica Digital y Sistemas Digitales 3. Sistemas multiplataforma (*cross-platform*)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las características únicas de los sistemas embebidos versus otros sistemas [Describir]
2. Identificar diferentes tipos de arquitecturas embebidas [Analizar]
3. Evaluar qué arquitectura es mejor para un conjunto dado de requisitos [Evaluar]

3.17.10. SPD/Sistemas, Programación y Sensores (Core Tier2: 2 hrs)

Cubre los sistemas embebidos con microcontroladores, interrupciones y eficiencia energética, la programación embebida, y la interfaz con sensores y actuadores.

Temas:

Core

- Sistemas Embebidos 1. Microcontroladores 2. Interrupciones y retroalimentación (*feedback*) 3. Manejadores de interrupciones en lenguajes de alto nivel Componentes Hardware y Lógica, Capas, Abstracciones y Escalabilidad 4. Interrupciones duras (*hard*) y suaves (*soft*) y salidas de trampa (*trap-exits*) Principios del Sistema Operativo 5. Interacción con hardware, actuadores y sensores 6. Eficiencia energética 7. Codificación débilmente temporizada (*loosely timed coding*) y sincronización 8. Adaptadores de software (software adapters)
- Programación embebida
- Sensores y actuadores

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Interfaz con sensores/actuadores [Programar]
2. Depurar un problema con una plataforma embebida existente [Analizar]
3. Diseñar y desarrollar software para interactuar y controlar hardware [Diseñar]

3.17.11. SPD/Tiempo Real, Memoria y Verificación (Core Tier2: 3 hrs)

Cubre los sistemas de tiempo real duro y suave, la gestión de recursos y memoria en tiempo real, las consideraciones de seguridad (safety), el análisis y verificación, y el diseño de aplicaciones embebidas.

Temas:

Core

- Sistemas de tiempo real duro (*hard*) vs sistemas de tiempo real suave (*soft*) Sistemas en Tiempo Real y Embebidos 1. Oportunidad (*timeliness*) 2. Sincronización/planificación del tiempo 3. Priorización 4. Latencia 5. Jitter de cómputo (*compute jitter*)
- Gestión de recursos en tiempo real
- Gestión de memoria 1. Mapeo de construcciones de programación (variable) a una ubicación de memoria Jerarquía de Memoria 2. Memoria compartida Gestión de Memoria 3. Gestión manual de memoria. 4. Recolección de basura (*garbage collection*) Traducción y Ejecución de Lenguajes
- Consideraciones de seguridad (*safety*) y análisis de seguridad Equidad Digital y Accesibilidad, Impacto Social e IA, Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo
- Análisis y verificación
- Diseño de aplicación

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Diseñar e implementar un pequeño sistema embebido para una plataforma dada (por ejemplo, un despertador inteligente o un dron) [Diseñar]
2. Diseñar métodos para sistemas de tiempo real [Diseñar]
3. Evaluar la planificación en tiempo real y el análisis de planificabilidad (*schedulability*) [Evaluar]
4. Evaluar la especificación y verificación formal de restricciones y propiedades de tiempo [Evaluar]

3.17.12. SPD/Historia, Tipos y Características de Plataformas (Core Tier2: 1 hr)

Cubre la evolución histórica de las plataformas de juegos, los tipos de plataformas típicas y sus características y restricciones técnicas.

Temas:
Core

- Evolución de las Plataformas de Juegos (por ejemplo, de Brown Box a Metaverso y más allá; mejora en las arquitecturas de computación (CPU y GPU); convergencia de plataformas y movilidad) Lógica Digital y Sistemas Digitales
- Plataformas de Juego Típicas (por ejemplo, Computadora Personal; Consola Doméstica; Consola Portátil; Máquina de Arcade; Televisión Interactiva; Teléfono Móvil; Tableta; Pantalla Montada en la Cabeza Integrada; Instalaciones Inmersivas y Simuladores; Dispositivos Habilitados para Internet de las Cosas; Sistemas CAVE; Navegadores Web; Sistemas de Transmisión Basados en la Nube)
- Características y Restricciones de Diferentes Plataformas de Juegos (por ejemplo, características: almacenamiento local, interconexión de redes, periféricos; rendimiento en tiempo de ejecución: frecuencia GPU/CPU, número de núcleos; conjuntos de chips: unidades de procesamiento de física, co-procesadores vectoriales; ancho de banda de expansión (PCIe); rendimiento de red (Ethernet); tipos y capacidades de memoria (DDR/GDDR); profundidad máxima de pila; consumo de energía; diseño térmico; Endian)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir las restricciones clave que una plataforma de juego específica probablemente impondrá a los desarrolladores [Describir]
2. Comparar las interfaces de diferentes plataformas de juegos, resaltando sus respectivas implicaciones para la interacción humano-computadora [Comparar]

3.17.13. SPD/Sensores, Controladores y Ecosistemas eSports (Core Tier2: 1 hr)

Cubre los sensores, controladores y actuadores típicos en plataformas de juego, y los ecosistemas de eSports incluyendo eventos, medios transmitidos y juego profesional.

Temas:
Core

- Sensores, Controladores y Actuadores Típicos (por ejemplo, diseños distintivos de sistemas de control - periféricos (ratón, teclado, joystick), controladores de juego, wearables, superficies interactivas; electrónica y hardware a medida; visión por computadora, seguimiento interno (*inside-out tracking*) y externo (*outside-in tracking*); electrónica y E/S habilitadas para IoT) Interacción

- Ecosistemas de eSports (por ejemplo, evolución del gameplay a través de plataformas; juegos y eSports; eventos de juegos como torneos LAN/arcade y eventos internacionales como la Serie Olímpica de eSports; medios transmitidos (*streamed*) y espectadores; tecnologías multimedia y gestión de transmisión; juego profesional; datos y aprendizaje automático para entrenamiento y coaching)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar cómo los eSports se transmiten (streamed) a grandes audiencias a través de internet [Explicar]

3.17.14. SPD/Arquitecturas y Renderizado (Core Tier2: 1 hr)

Cubre las arquitecturas de CPU y GPU para juegos y las tuberías (pipelines) para renderizado gráfico y simulación física.

Temas:
Core

- Arquitecturas de CPU y GPU (por ejemplo, taxonomía de Flynn; paralelización; conjuntos de instrucciones; componentes estándar - arreglo de cómputo gráfico (*graphics compute array*), controlador de memoria gráfica (*graphics memory controller*), sistema básico de E/S de matriz de gráficos de video (*video graphics array basic input/output system*); interfaz de bus; unidad de gestión de energía (*power management unit*); unidad de procesamiento de video (*video processing unit*); interfaz de pantalla (*display interface*)) Arquitecturas Heterogéneas

- Tuberías (*pipelines*) para Renderizado Gráfico y Simulación Física (por ejemplo, basado en mosaicos (*tile-based*), modo inmediato (*immediate-mode*)) Renderizado Aplicado y Técnicas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Recordar las características de las arquitecturas comunes de procesamiento gráfico de propósito general [Recordar/Evocar]
2. Identificar las etapas clave de la tubería de renderizado de modo inmediato (immediate-mode rendering pipeline) [Analizar]

3.17.15. SPD/Algoritmos, Medios y Mundos Virtuales (Core Tier2: 1 hr)

Cubre los algoritmos, estructuras de datos y funciones matemáticas comunes en juegos, y las representaciones de medios en mundos virtuales.
Temas:
Core

- Contextos Comunes para Algoritmos, Estructuras de Datos y Funciones Matemáticas (por ejemplo, bucles de juego (*game loops*); partición espacial, eliminación de viewport (*culling*) y nivel de detalle (*level of detail*); detección y resolución de colisiones (*collision detection and resolution*); simulación física; comportamiento para agentes inteligentes; generación procedural de contenido (*procedural content generation*)) Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, Matemáticas Discretas
- Representaciones de Medios en Mundos Virtuales (por ejemplo, E/S y técnicas de cómputo para mundos virtuales: audio; música; sprites; modelos y texturas; texto; diálogo; multimedia (por ejemplo, olfato, táctil)) Conceptos Fundamentales de Gráficos y Técnicas Interactivas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Traducir funciones matemáticas complejas a código fuente de buen rendimiento [Traducir]
2. Optimizar una función para la detección de colisiones en un entorno simulado [Crear]

3.17.16. SPD/Lenguajes, APIs y Herramientas de Desarrollo (Core Tier2: 1 hr)

Cubre los lenguajes de programación y sombreado, las APIs y bibliotecas gráficas, y las herramientas y entornos de desarrollo para juegos.
Temas:
Core

- Herramientas y Técnicas de Desarrollo de Juegos 1. Lenguajes de Programación (por ejemplo, C++; C#; Lua; Python; JavaScript) 2. Lenguajes de Sombreado (*Shader Languages*) (por ejemplo, HLSL, GLSL; Shader Graph) 3. Bibliotecas y APIs Gráficas (por ejemplo, DirectX; SDL; OpenGL; Metal; Vulkan; WebGL). Renderizado Aplicado y Técnicas, Diseño de Sistemas 4. Herramientas y Entornos de Desarrollo Comunes (por ejemplo, IDEs; Depuradores (*Debuggers*); Perfiladores (*Profilers*); Sistemas de Control de Versiones incluyendo aquellos que manejan activos binarios; Kits de Desarrollo y Kits de Producción/Consumidor; Emuladores) Prácticas de Desarrollo de Software, Entorno de Desarrollo e IDE, Control de Versiones y CI/CD

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Usar una API gráfica estándar de la industria para renderizar un modelo 3D en una escena virtual [Usar]
2. Modificar un sombreador (shader) para cambiar un efecto visual de acuerdo con los requisitos establecidos [Crear]
3. Recomendar un conjunto apropiado de herramientas y técnicas de desarrollo para implementar un juego de un género particular para una plataforma dada [Recomendar]

3.17.17. SPD/Motores de Juego (Core Tier2: 1 hr)

Cubre los motores de juego de código abierto y las técnicas de producción, garantía de calidad, optimización, portado e internacionalización.

Temas:
Core

- Motores de Juego (*Game Engines*) 1. Motores de Juego Abiertos (por ejemplo, Unreal; Unity; Godot; CryEngine; Phyre; Source 2; Pygame y Ren'Py; Phaser; Twine; Spring RTS) 2. Técnicas (por ejemplo, Ideación, Prototipado, Diseño e Implementación Iterativos, Compilación de Ejecutables (*Compiling Executable Builds*), Operaciones de Desarrollo y Garantía de Calidad - Pruebas de Juego (*Play Testing*) y Pruebas Técnicas (*Technical Testing*), Perfilado (*Profiling*); Optimización, Portado (*Porting*); Internacionalización y Localización, Redes) Rendimiento y Eficiencia Energética, Ingeniería de Requisitos, Evolución y Estimación de Requisitos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Implementar un juego para una plataforma particular según la especificación [Implementar]
2. Evaluar el rendimiento en tiempo de ejecución y de memoria de un juego usando una herramienta y entorno de desarrollo estándar de la industria [Evaluar (valorar)]
3. Discutir los desafíos clave en hacer un juego digital que sea compatible multiplataforma (cross-platform) [Debatir]

3.17.18. SPD/Fundamentos del Diseño de Juegos (Core Tier2: 1 hr)

Cubre el vocabulario y modelos del diseño de juegos, el pensamiento de diseño y el diseño de experiencia centrado en el usuario.

Temas:
Core

- Vocabulario del Diseño de Juegos (por ejemplo, definiciones de juego; modelo mecánica-dinámica-estética (*mechanics-dynamics-aesthetics model*); terminología de la industria; diseño de experiencia (*experience design*); modelos de experiencia y emoción)
- Pensamiento de Diseño (*Design Thinking*) y Diseño de Experiencia Centrado en el Usuario (por ejemplo, métodos de diseño de juegos; iteración, incremento y el doble diamante (*double-diamond*); fases de pre- y post-producción; garantía de calidad, incluyendo pruebas alfa y beta; participación de partes interesadas y clientes; gestión de la comunidad) Diseño de API y Modelado, Principios y Arquitectura de Software, Calidad y Evaluación del Diseño

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Expresar cómo los desarrolladores de juegos pueden mejorar la accesibilidad de una interfaz de juego [Demostrar]

3.17.19. SPD/Géneros, Audiencias y Alcance Social (Core Tier2: 1 hr)

Cubre los géneros de juegos, las audiencias y taxonomías de jugadores, y la proliferación de tecnologías de juegos a dominios más allá del entretenimiento.

Temas:
Core

- Géneros de Juegos (por ejemplo, aventura; simulador de caminata (*walking simulator*); disparos en primera persona (*first-person shooter*); estrategia en tiempo real (*real-time strategy*); arena de batalla multijugador en línea (MOBA); juego de rol (rpg))
- Audiencias y Taxonomías de Jugadores (por ejemplo, personas que juegan videojuegos; diversidad y ampliación de la participación; placeres, tipos de jugadores y preferencias; Bartle, Yee) Comprender al Usuario: Objetivos individuales e interacciones con otros
- Proliferación de Tecnologías de Juegos a Dominios Más Allá del Entretenimiento (por ejemplo, Educación y Entrenamiento; Juegos Serios (*Serious Games*); Producción Virtual; eSports; Gamificación; Diseño de Experiencias Inmersivas; Práctica de la Industria Creativa; Práctica Artística; Retórica Procedural (*Procedural Rhetoric*)) Aplicaciones e Impacto Social de la IA, Aplicaciones Adicionales e Impacto Social de la IA

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Crear formas novedosas de gameplay utilizando plataformas de juego de frontera [Crear]

3.17.20. SPD/Plataformas Interactivas de Datos y Análisis (Core Tier2: 2 hrs)

Cubre las plataformas de análisis de datos interactivo como cuadernos Jupyter, Google Colab, R, plataformas SQL en la nube y herramientas de diseño con bucles de retroalimentación de baja latencia.
Temas:
Non Core

- Plataformas de Cuadernos Computacionales (por ejemplo, Cuadernos Jupyter; Google Colab; R; SPSS; Observable)
- Plataformas de Análisis de Datos y SQL en la Nube (por ejemplo, BigQuery) Construcción de Consultas 1. Apache Spark 2. Visualizaciones de Datos Visualización
- Presentaciones Interactivas Respaldadas por Datos
- Herramientas de Diseño con Bucles de Retroalimentación de Baja Latencia (*Low-Latency Feedback Loops*) 1. Herramientas de renderizado 2. Herramientas de diseño gráfico

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Analizar grandes conjuntos de datos de forma interactiva [Analizar]
2. Crear una pista de acompañamiento para una interpretación musical, como la codificación en vivo (*live coding*) [Crear]
3. Crear cuadernos computacionales convincentes que construyan una narrativa para un objetivo/historia periodística dada [Crear]
4. Implementar código interactivo que use un conjunto de datos y genere gráficos exploratorios [Implementar]
5. Contextualizar los atributos de diferentes estilos de análisis de datos, como interactivo versus tubería (*pipeline*) de ingeniería [Contextualizar]
6. Escribir un programa usando una plataforma de computación de cuadernos (por ejemplo, búsqueda, ordenación o manipulación de grafos) [Escribir]

3.17.21. SPD/Programación con IA y Plataformas Cuánticas (Core Tier2: 1 hr)

Cubre la programación mediante prompts con modelos de lenguaje grande (LLMs) e IA generativa, y las plataformas cuánticas para programar operadores de lógica cuántica y servicios cuánticos disponibles.
Temas:
Non Core

- Programación mediante prompts (Prompt programming) 1. IA generativa (por ejemplo, ChatGPT de OpenAI, Codex de OpenAI, Copilot de GitHub) y los Modelos de Lenguaje Grande (LLMs) son accedidos/interactuados
- Plataformas Cuánticas Arquitecturas Cuánticas 1. Programar operadores de lógica cuántica en máquinas cuánticas. 2. Usar API para servicios cuánticos disponibles 3. Análisis de señales/Análisis de Fourier/Procesamiento de señales (para composición musical, análisis de audio/RF) Procesamiento de Imágenes

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Crear un programa que realice una tarea usando sistemas de Modelos de Lenguaje Grande (LLM) [Crear]
2. Contrastar un programa desarrollado por una plataforma de IA y por un humano [Contrastar]
3. Implementar un sistema que interactúe con un humano sin usar una pantalla [Implementar]
4. Demostrar un resultado de puerta cuántica (*quantum gate*) usando una plataforma cuántica [Demostrar]

3.17.22. SPD/SPD-SEP/Móvil

Temas:
Non Core

- Privacidad y protección de datos
- Accesibilidad en el diseño móvil
- Seguridad y ciberseguridad

- Impactos sociales de la tecnología móvil
- Uso ético de la IA y los algoritmos

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Comprender y defender las responsabilidades éticas para salvaguardar la privacidad del usuario y la protección de datos en aplicaciones móviles [Evaluar (valorar)]
2. Diseñar aplicaciones móviles con la accesibilidad en mente, asegurando un uso efectivo por personas con discapacidades [Diseñar]
3. Demostrar competencia en prácticas de codificación segura para mitigar riesgos asociados con diversas amenazas de seguridad en el desarrollo móvil [Demostrar]
4. Analizar los impactos sociales más amplios de la tecnología móvil, incluyendo su influencia en los patrones de comunicación, relaciones y salud mental [Analizar]
5. Comprender las consideraciones éticas del uso de IA en aplicaciones móviles, asegurando algoritmos imparciales y justos [Explicar]

3.17.23. SPD/SPD-SEP Web

Temas:
Non Core

- Preocupaciones de privacidad con las aplicaciones móviles
- Diseñar para la inclusividad y la accesibilidad
- Uso ético de la IA en aplicaciones móviles
- Desarrollo sostenible de aplicaciones y alojamiento de servidores
- Evitar spam o notificaciones intrusivas
- Abordar el ciberacoso y el acoso (*cyberbullying and harassment*)
- Promover comunidades en línea positivas
- Monetización y publicidad
- Uso ético de la gamificación

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Comprender cómo la computación móvil impacta las comunicaciones y el flujo de información dentro de la sociedad [Analizar]
2. Diseñar aplicaciones móviles que hayan hecho las tareas diarias más fáciles/rápidas [Diseñar]
3. Reconocer cómo la ubicuidad de la computación móvil ha afectado el equilibrio entre el trabajo y la vida personal [Reconocer]
4. Comprender cómo la computación móvil impacta el monitoreo de la salud y los servicios de atención médica [Analizar]
5. Definir cómo las aplicaciones móviles se usan para educar sobre y ayudar a lograr los objetivos de sostenibilidad de la ONU [Definir]

3.17.24. SPD/Propiedad Intelectual en los Juegos

Cubre los derechos de propiedad intelectual en las industrias creativas, incluyendo copyright, marcas, licenciamiento y cumplimiento.

Temas:
Non Core

- Derechos de Propiedad Intelectual en las Industrias Creativas 1. Propiedad de la Propiedad Intelectual: derechos de autor (*copyright*), marca registrada (*trademark*); derecho de diseño (*design right*), patente (*patent*), secreto comercial (*trade secret*), ley civil versus penal; acuerdos internacionales; generación procedural de contenido y las implicaciones de la inteligencia artificial generativa 2. Licenciamiento (Licensing): Excepciones de uso y uso justo (*fair usage*); acuerdos de licencia de código abierto (*open-source*); licenciamiento propietario y a medida (*bespoke*); cumplimiento (*enforcement*)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Discutir cómo los creadores pueden proteger su propiedad intelectual [Debatir]

2. Comparar opciones para el licenciamiento de código abierto de una herramienta de desarrollo de juegos [Comparar]

3.17.25. SPD/Accesibilidad e Interfaz del Juego

Cubre la usabilidad y accesibilidad de las interfaces de juego, incluyendo métodos de evaluación heurística y diseño para jugadores con discapacidades.

Temas:

Non Core

- Acceso Justo al Juego (*Fair Access to Play*) 1. Usabilidad de la Interfaz de Juego: requisitos del usuario, capacidades (*affordances*), diseño ergonómico, investigación de usuarios, medición de experiencia y métodos de evaluación heurística para juegos 2. Accesibilidad de la Interfaz de Juego: formas de discapacidad y deficiencia (impairment and disability); medios para facilitar el acceso al juego; diseño universal; requisitos legislados para plataformas de juegos; evaluación de cumplimiento; desafíos en mecánicas de juego y acceso

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Identificar trampas comunes en las interfaces de juego que excluyen a jugadores con visión deficiente o no funcional [Analizar]
2. Describir cómo la evaluación heurística puede usarse para identificar problemas de usabilidad en interfaces de juego [Describir]
3. Recomendar cambios a una interfaz de juego específica para mejorar el acceso a jugadores que son sordos o cuya audición está de otra manera deficiente [Recomendar]

3.17.26. SPD/Salud, Seguridad y Sostenibilidad

Cubre las lesiones relacionadas con el juego y su mitigación, la evaluación de riesgos para eventos y fabricación, la salud mental de los jugadores, y la cadena de suministro y sostenibilidad del hardware de plataforma.

Temas:

Non Core

- Salud y Seguridad Relacionadas con los Juegos 1. Lesiones en el Juego: formas de mitigar lesiones comunes en la parte superior del cuerpo, como lesiones por esfuerzo repetitivo (repetitive strain injury); psicología del ejercicio y fisioterapia en eSports 2. Evaluación de Riesgos para Eventos y Fabricación: control de sustancias peligrosas para la salud (COSHH); seguridad contra incendios; seguridad eléctrica y electrónica; evaluación de riesgos para juegos y eventos de juegos; evaluación de riesgos para la fabricación 3. Salud Mental: motivación para jugar; gamificación y diseño lúdico (*gameful design*); psicología del juego - trastorno por juego en internet (*internet gaming disorder*)
- Cadena de Suministro y Sostenibilidad del Hardware de Plataforma 1. Ciclo de Vida de la Plataforma: composición de la plataforma - materiales, ensamblaje; excavación y procesamiento de minerales; uso de energía; reciclaje; obsolescencia programada (*planned obsolescence*). 2. Esclavitud Moderna (Modern Slavery): cadenas de suministro; trabajo forzado y derechos civiles; condiciones laborales; detección y remisión (*remission*); organismos de certificación y esfuerzos benéficos.

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Explicar por qué las lesiones en la parte superior del cuerpo son comunes en los eSports [Explicar]
2. Modificar una política para un evento de LAN party para incluir mitigaciones que reduzcan el riesgo de incendio [Diseñar]
3. Diseñar una estrategia de gamificación para motivar el juego serio (*serious play*) para un juego de concientización [Diseñar]
4. Evaluar la idoneidad de dos fabricantes para producir una nueva consola de juegos [Evaluar (valorar)]
5. Discutir si los juegos son adictivos [Debatir]

3.17.27. SPD/Representación, Inclusión e Igualdad

Cubre la representación en los medios y la industria de los juegos, la inclusión de audiencias diversas, la igualdad y la movilidad social en la industria, y el uso de juegos para la defensa de causas sociales.

Temas:

Non Core

- Representación en los Medios y la Industria 1. Inclusión: identidad e identificación; inclusión de una amplia gama de personajes para audiencias diversas; representación en los medios y sus efectos; alfabetización mediática (*media literacy*); análisis de contenido; estereotipos; sexualización 2. Igualdad (Equality): historias y controversias, como gamergate, calidad de vida en la industria, discurso y conducta profesional en contextos empresariales, trayectorias hacia carreras en desarrollo de juegos, movilidad social, la experiencia de desarrolladores de diferentes orígenes e identidades, género y tecnología

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Discutir cómo reformar personajes y diálogos en una escena para reducir la amenaza del estereotipo (*stereotype threat*) [Debatir]
2. Ilustrar cómo la representación de la raza en un juego puede influir en el riesgo de exclusión social en la comunidad en línea asociada al juego [Aplicar]
3. Analizar el papel de las políticas de contratación de empresas y la defensa (*advocacy*) en la movilidad social [Analizar]
4. Sugerir cómo la representación de las mujeres en los videojuegos influye en cómo los jugadores perciben a los miembros de esos grupos [Evaluar]
5. Crear un videojuego que defienda con éxito la ciencia del clima [Crear]

3.17.28. SPD/SPD-SEP/Robótica

Temas:

Non Core

- Imparcialidad (*fairness*), transparencia y rendición de cuentas (*accountability*) en algoritmos robóticos
- Mitigar sesgos (biases) en la toma de decisiones de robots
- Seguridad pública en espacios compartidos con robots
- Cumplimiento con las leyes de protección de datos
- Consentimiento del paciente y confianza en los robots médicos

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Identificar instancias de sesgo en algoritmos robóticos y proponer estrategias para mitigarlos [Analizar]
2. Evaluar y criticar sistemas robóticos por consideraciones éticas y de imparcialidad, sugiriendo mejoras donde sea necesario [Evaluar]
3. Analizar ejemplos del mundo real de sesgos en la toma de decisiones de robots y desarrollar estrategias para reducir el sesgo en sistemas robóticos [Analizar]
4. Evaluar los riesgos potenciales asociados con robots en espacios públicos y proponer medidas de seguridad para mitigar esos riesgos [Evaluar (valorar)]
5. Evaluar el impacto del consentimiento del paciente y la confianza en la efectividad de los despliegues de robots médicos en contextos de atención médica [Evaluar]

3.17.29. SPD/SPD-SEP/Interactiva

Temas:

Non Core

- Pautas éticas al usar modelos de IA para ayudar en el periodismo y la creación de contenido
- Rendición de cuentas (*accountability*) por las salidas generadas por IA
- Comportamiento entre programadores de prompts (*prompt programmers*) y desarrolladores de IA

- Confianza con el público al usar modelos de IA

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Indicar un marco para la rendición de cuentas (*accountability*) en el despliegue de modelos de IA, incluyendo documentación clara y atribución [Evaluar]
2. Discutir códigos de conducta éticos y estándares profesionales relevantes para la programación de prompts (*prompt programming*) y el desarrollo de IA [Debatir]
3. Crear planes y materiales de comunicación para educar al público sobre las capacidades, limitaciones y salvaguardas éticas de la IA [Crear]

Capítulo 4

Cuerpo del conocimiento de Matemática

Las 12 áreas de conocimiento en Matemática son:

4.1 Cálculo y Análisis (CAN) (Pág. 182)
- 4.1.1 Límites y Continuidad (Pág. 182)
- 4.1.2 Cálculo Diferencial (Pág. 183)
- 4.1.3 Antiderivadas y Técnicas de Integración (Pág. 183)
- 4.1.4 Integrales Impropias (Pág. 183)
- 4.1.5 Sucesiones, Series y Series de Potencias (Pág. 184)
- 4.1.6 Aplicaciones de la Integral Definida (Pág. 184)
- 4.1.7 Geometría en el Espacio (Pág. 184)
- 4.1.8 Superficies Cuádricas (Pág. 185)
- 4.1.9 Ecuaciones Paramétricas y Coordenadas Polares (Pág. 185)
- 4.1.10 Cálculo Diferencial Multivariable (Pág. 185)
- 4.1.11 Integrales Múltiples (Pág. 186)
- 4.1.12 Integrales de Línea y Campos Vectoriales (Pág. 186)
- 4.1.13 Integrales de Superficie y Teoremas Fundamentales (Pág. 187)
- 4.1.14 Funciones Vectoriales de una Variable Real (Pág. 187)
- 4.1.15 Transformaciones Multivariables (Pág. 187)
- 4.1.16 Análisis Real e Integración de Lebesgue (Pág. 188)
- 4.1.17 Análisis Complejo y Aplicaciones Conformes (Pág. 188)
- 4.1.18 Análisis Funcional y Espacios de Banach/Hilbert (Pág. 188)
- 4.1.19 Análisis Armónico y Teoría de Fourier (Pág. 189)
- 4.1.20 Cálculo de Variaciones (Pág. 189)
- 4.1.21 Teoría de la Medida (Pág. 189)

4.2 álgebra y Teoría de Números (ANT) (Pág. 190)
- 4.2.1 Matrices y Sistemas de Ecuaciones Lineales (Pág. 190)
- 4.2.2 Espacios Vectoriales y Transformaciones Lineales (Pág. 191)
- 4.2.3 Valores Propios, Producto Interno y Formas Cuadráticas (Pág. 191)
- 4.2.4 Teoría de Grupos (Pág. 191)
- 4.2.5 Anillos, Cuerpos y Teoría de Galois (Pág. 192)
- 4.2.6 Teoría de Módulos y álgebra Conmutativa (Pág. 192)
- 4.2.7 Teoría Analítica de Números (Pág. 192)
- 4.2.8 Teoría Algebraica de Números (Pág. 193)
- 4.2.9 Teoría de Representaciones (Pág. 193)
- 4.2.10 Criptografía y Teoría de la Codificación (Pág. 193)

4.3 Geometría y Topología (GTO) (Pág. 194)
- 4.3.1 Geometría Euclidiana y Proyectiva (Pág. 194)
- 4.3.2 Topología General (Pág. 194)
- 4.3.3 Geometría Diferencial de Curvas y Superficies (Pág. 195)
- 4.3.4 Variedades Diferenciables y Formas Diferenciales (Pág. 195)
- 4.3.5 Geometría Riemanniana (Pág. 196)
- 4.3.6 Topología Algebraica (Pág. 196)
- 4.3.7 Geometría Simpléctica (Pág. 196)
- 4.3.8 Teoría Geométrica de Grupos (Pág. 197)

4.4 Matemáticas Discretas y Combinatoria (DMC) (Pág. 197)
- 4.4.1 Combinatoria Enumerativa (Pág. 197)
- 4.4.2 Teoría de Grafos (Pág. 198)
- 4.4.3 Combinatoria Extremal y Probabilística (Pág. 198)
- 4.4.4 Optimización Discreta (Pág. 198)
- 4.4.5 Teoría de Diseños y Códigos Combinatorios (Pág. 199)
- 4.4.6 Lógica Matemática y Teoría de Conjuntos (Pág. 199)

4.5 Probabilidad y Estadística (PST) (Pág. 200)
- 4.5.1 Axiomas de Probabilidad y Distribuciones (Pág. 200)
- 4.5.2 Teoremas Límite (Pág. 200)
- 4.5.3 Procesos Estocásticos (Pág. 201)
- 4.5.4 Teoría de la Estimación (Pág. 201)
- 4.5.5 Contraste de Hipótesis (Pág. 201)
- 4.5.6 Inferencia Bayesiana (Pág. 202)
- 4.5.7 Estadística de Alta Dimensión (Pág. 202)
- 4.5.8 Análisis de Series Temporales (Pág. 202)
- 4.5.9 Métodos No Paramétricos (Pág. 203)

4.6 Análisis Numérico y Científico (NSA) (Pág. 203)
- 4.6.1 Análisis de Error y Aritmética de Punto Flotante (Pág. 204)
- 4.6.2 Búsqueda de Raíces de Ecuaciones No Lineales (Pág. 204)
- 4.6.3 álgebra Lineal Numérica (Pág. 204)
- 4.6.4 Teoría de Aproximación e Interpolación (Pág. 205)
- 4.6.5 Integración Numérica y Cuadratura (Pág. 205)
- 4.6.6 Métodos Numéricos para Ecuaciones Diferenciales (Pág. 206)
- 4.6.7 Método de Elementos Finitos (Pág. 206)
- 4.6.8 Algoritmos de Optimización (Pág. 206)
- 4.6.9 Computación Paralela y de Alto Rendimiento (Pág. 207)

4.7 Modelado y Simulación Matemática (MMS) (Pág. 207)
- 4.7.1 Ecuaciones Diferenciales Ordinarias de Primer Orden (Pág. 208)
- 4.7.2 Ecuaciones Diferenciales de Orden Superior y Sistemas (Pág. 208)
- 4.7.3 Ecuaciones en Derivadas Parciales (Pág. 208)
- 4.7.4 Sistemas Dinámicos y Caos (Pág. 209)
- 4.7.5 Análisis Asintótico y Métodos de Perturbación (Pág. 209)
- 4.7.6 Dinámica de Fluidos Matemática (Pág. 209)
- 4.7.7 Teoría de Control Matemático (Pág. 210)
- 4.7.8 Ecuaciones Diferenciales Estocásticas (Pág. 210)
- 4.7.9 Modelado Basado en Agentes y Sistemas Complejos (Pág. 211)

4.8 Fundamentos Matemáticos de la Ciencia de Datos (FDS) (Pág. 211)
- 4.8.1 Teoría del Aprendizaje Estadístico (Pág. 211)
- 4.8.2 Optimización para Aprendizaje Automático (Pág. 212)
- 4.8.3 Análisis Topológico de Datos (Pág. 212)
- 4.8.4 Reducción de Dimensionalidad (Pág. 212)
- 4.8.5 Métodos de Kernel y Espacios de Hilbert con Núcleo Reproductor (Pág. 213)

- 4.8.6 Teoría de la Información (Pág. 213)
- 4.8.7 Fundamentos Matemáticos del Aprendizaje Profundo (Pág. 214)

4.9 Geometría y Topología Computacional (CGT) (Pág. 214)

- 4.9.1 Diagramas de Voronoi y Triangulaciones de Delaunay (Pág. 214)
- 4.9.2 Envolventes Convexas y Politopos (Pág. 215)
- 4.9.3 Topología Digital y Análisis de Imágenes (Pág. 215)
- 4.9.4 Homología Persistente y Topología Aplicada (Pág. 216)
- 4.9.5 Generación de Mallas y Algoritmos Geométricos (Pág. 216)

4.10 Información y Computación Cuántica (QIC) (Pág. 216)

- 4.10.1 Fundamentos de Mecánica Cuántica para Computación (Pág. 217)
- 4.10.2 Circuitos y Puertas Cuánticas (Pág. 217)
- 4.10.3 Algoritmos Cuánticos (Pág. 218)
- 4.10.4 Entrelazamiento y Desigualdades de Bell (Pág. 218)
- 4.10.5 Corrección de Errores Cuánticos (Pág. 218)
- 4.10.6 Redes de Tensores y Simulación Cuántica (Pág. 219)

4.11 Biología Matemática y Ciencias de la Vida (MBL) (Pág. 219)

- 4.11.1 Dinámica de Poblaciones (Pág. 219)
- 4.11.2 Epidemiología Matemática (Pág. 220)
- 4.11.3 Biología de Sistemas y Dinámica de Redes (Pág. 220)
- 4.11.4 Algoritmos de Bioinformática (Pág. 221)
- 4.11.5 Neurociencia Matemática (Pág. 221)

4.12 Matemáticas Financieras y Actuariales (FAM) (Pág. 222)

- 4.12.1 Teoría del Interés y Valor del Dinero en el Tiempo (Pág. 222)
- 4.12.2 Valoración Libre de Arbitraje y Modelos de Mercado (Pág. 222)
- 4.12.3 Modelo de Black-Scholes y Valoración de Opciones (Pág. 223)
- 4.12.4 Medidas de Riesgo y Teoría de Portafolios (Pág. 223)
- 4.12.5 Modelos Actuariales y Tablas de Mortalidad (Pág. 224)
- 4.12.6 Finanzas Computacionales y Métodos de Monte Carlo (Pág. 224)

4.1. Cálculo y Análisis (CAN)

Esta área cubre el estudio fundamental del cambio, los límites y las propiedades rigurosas de las funciones. Abarca desde la diferenciación e integración básicas hasta la teoría de la medida avanzada y los espacios funcionales, proporcionando el conjunto de herramientas esencial para todo modelado matemático.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.1.1 Límites y Continuidad (Pág. 182)	1	1
⭐ 4.1.2 Cálculo Diferencial (Pág. 183)	1	1
⭐ 4.1.3 Antiderivadas y Técnicas de Integración (Pág. 183)	1	1
⭐ 4.1.4 Integrales Impropias (Pág. 183)	1	1
⭐ 4.1.5 Sucesiones, Series y Series de Potencias (Pág. 184)	1	1
⭐ 4.1.6 Aplicaciones de la Integral Definida (Pág. 184)	1	1
⭐ 4.1.7 Geometría en el Espacio (Pág. 184)	1	1
⭐ 4.1.8 Superficies Cuádricas (Pág. 185)	1	1
⭐ 4.1.9 Ecuaciones Paramétricas y Coordenadas Polares (Pág. 185)	1	1
⭐ 4.1.10 Cálculo Diferencial Multivariable (Pág. 185)	1	1
⭐ 4.1.11 Integrales Múltiples (Pág. 186)	1	1
⭐ 4.1.12 Integrales de Línea y Campos Vectoriales (Pág. 186)	1	1
⭐ 4.1.13 Integrales de Superficie y Teoremas Fundamentales (Pág. 187)	1	1
⭐ 4.1.14 Funciones Vectoriales de una Variable Real (Pág. 187)	1	1
⭐ 4.1.15 Transformaciones Multivariables (Pág. 187)	1	1
⭐ 4.1.16 Análisis Real e Integración de Lebesgue (Pág. 188)	1	1
⭐ 4.1.17 Análisis Complejo y Aplicaciones Conformes (Pág. 188)	1	1
4.1.18 Análisis Funcional y Espacios de Banach/Hilbert (Pág. 188)	Electivo	
4.1.19 Análisis Armónico y Teoría de Fourier (Pág. 189)	Electivo	
4.1.20 Cálculo de Variaciones (Pág. 189)	Electivo	
⭐ 4.1.21 Teoría de la Medida (Pág. 189)	1	1

Cuadro 4.1: Lista de KUs del área de Cálculo y Análisis.

4.1.1. CAN/Límites y Continuidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estudio riguroso de los límites y la continuidad de funciones de una variable, incluyendo la definición épsilon-delta, las leyes de los límites y los teoremas fundamentales.

Temas:

Core

- Definición épsilon-delta de un límite y técnicas de demostración formal

- Leyes de los límites, límites laterales y límites en el infinito
- Continuidad: tipos, propiedades y el Teorema del Valor Intermedio
- Regla de L'Hôpital y formas indeterminadas
- Límites de sucesiones: convergencia, Teorema del Sándwich y Bolzano-Weierstrass

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar la definición épsilon-delta de un límite y explicar su significado geométrico [Familiarizarse]
2. Evaluar límites usando leyes de límites, la regla de L'Hôpital y análisis asintótico [Usar]
3. Demostrar continuidad o discontinuidad de una función y aplicar el Teorema del Valor Intermedio [Evaluar]

4.1.2. CAN/Cálculo Diferencial (Core Tier1: 1 hr, Core Tier2: 1 hr)

Derivadas de funciones de una variable: definición, reglas, derivadas de orden superior y aplicaciones a la optimización y análisis de curvas.

Temas:
Core

- Derivada como límite; interpretaciones geométrica y física; diferenciabilidad
- Reglas del producto, cociente y cadena; diferenciación implícita; razones relacionadas
- Teorema de Rolle, Teorema del Valor Medio y criterios de monotonía
- Derivadas de orden superior, polinomios de Taylor y estimación del resto
- Extremos: criterios de la primera y segunda derivada, trazado de curvas y optimización aplicada

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Interpretar la derivada como una tasa de cambio y como la pendiente de la recta tangente [Familiarizarse]
2. Aplicar reglas de diferenciación y diferenciación implícita para calcular derivadas de funciones compuestas e implícitas [Usar]
3. Resolver problemas de optimización y trazar curvas usando el Teorema del Valor Medio y los criterios de derivadas [Evaluar]

4.1.3. CAN/Antiderivadas y Técnicas de Integración (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- El Teorema Fundamental del Cálculo y la antiderivada
- Técnicas de integración: sustitución, partes, fracciones parciales y sustitución trigonométrica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar el Teorema Fundamental del Cálculo y explicar la relación entre diferenciación e integración [Familiarizarse]
2. Calcular integrales definidas e indefinidas usando técnicas estándar, incluyendo sustitución e integración por partes [Usar]

4.1.4. CAN/Integrales Impropias (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Integrales impropias: criterios de convergencia y pruebas de comparación

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Determinar la convergencia de integrales impropias usando criterios de comparación y otras pruebas de convergencia [Evaluar]

4.1.5. CAN/Sucesiones, Series y Series de Potencias (Core Tier1: 1 hr, Core Tier2: 1 hr)

Temas:
Core

- Sucesiones y series: pruebas de convergencia (cociente, raíz, comparación, integral)
- Series de potencias, radio de convergencia, series de Taylor y Maclaurin

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Determinar la convergencia de series de potencias y construir series de Taylor para funciones estándar [Evaluar]

4.1.6. CAN/Aplicaciones de la Integral Definida (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aplicaciones geométricas y físicas de la integral definida: áreas, volúmenes de sólidos de revolución, longitud de arco, área de superficie, trabajo y centroides.
Temas:
Core

- Área entre curvas mediante integración respecto a x o a y
- Volumen de sólidos de revolución por el método del disco y la arandela
- Volumen de sólidos de revolución por el método de la capa cilíndrica (shells)
- Longitud de arco de una curva plana
- Área de superficie de sólidos de revolución
- Trabajo mediante integración (resortes, bombeo de fluidos, cables)

Electivo

- Momentos y centroide de una región o sólido plano
- Fuerza y presión hidrostática mediante integración

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular el área entre dos curvas integrando respecto a x o a y, según convenga [Usar]
2. Calcular el volumen de un sólido de revolución usando el método del disco o la arandela [Usar]
3. Calcular el volumen de un sólido de revolución usando el método de la capa cilíndrica [Usar]
4. Calcular la longitud de arco de una curva dada por $y = f(x)$ [Usar]
5. Calcular el trabajo realizado por una fuerza variable usando integración [Usar]

Elective:

6. Calcular el centroide de una región plana usando momentos e integración [Usar]
7. Calcular la fuerza hidrostática sobre una superficie sumergida usando integración [Evaluar]

4.1.7. CAN/Geometría en el Espacio (Core Tier1: 1 hr, Core Tier2: 1 hr)

Vectores en el espacio tridimensional, el producto punto y el producto cruz, y su aplicación a rectas y planos en $\mathbb{R}^3$.
Temas:
Core

- Vectores en $\mathbb{R}^3$: operaciones, magnitud y vectores unitarios
- Producto punto (escalar): definición, propiedades y ángulo entre vectores
- Producto cruz (vectorial): definición, propiedades e interpretación geométrica
- Producto mixto (triple escalar) y su interpretación como volumen
- Ecuaciones vectorial, paramétrica y simétrica de una recta en el espacio
- Ecuación del plano: vector normal, forma general y distancia punto-plano

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular el producto punto y el producto cruz de vectores en $\mathbb{R}^3$ e interpretar sus resultados geométricamente [Usar]

2. Determinar las ecuaciones vectorial, paramétrica y simétrica de una recta dado un punto y un vector director [Usar]
3. Determinar la ecuación de un plano dado un punto y un vector normal, o tres puntos no colineales [Usar]
4. Calcular distancias entre puntos, rectas y planos en el espacio usando productos vectoriales [Evaluar]

4.1.8. CAN/Superficies Cuádricas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Clasificación y graficación de superficies cuádricas en $\mathbb{R}^3$ a partir de su ecuación general.

Temas:

Core

- Superficies cilíndricas y trazas en los planos coordenados
- Elipsoides y esferas: ecuación canónica y trazas
- Paraboloides elípticos e hiperbólicos
- Hiperboloides de una y de dos hojas
- Conos elípticos

Electivo

- Coordenadas cilíndricas y esféricas para describir superficies

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Clasificar una superficie cuádrica a partir de su ecuación general usando el método de trazas [Usar]
2. Graficar una superficie cuádrica identificando sus trazas en los planos coordenados [Usar]

Elective:

3. Convertir puntos y ecuaciones entre coordenadas cartesianas, cilíndricas y esféricas [Usar]

4.1.9. CAN/Ecuaciones Paramétricas y Coordenadas Polares (Core Tier1: 1 hr, Core Tier2: 1 hr)

Curvas paramétricas y su cálculo diferencial e integral, y coordenadas polares con sus cálculos asociados de área y longitud de arco.

Temas:

Core

- Curvas definidas paramétricamente: $x = f(t)$, $y = g(t)$
- Derivadas y rectas tangentes a curvas paramétricas; longitud de arco paramétrica
- Sistema de coordenadas polares y conversión con coordenadas cartesianas
- Graficación de curvas polares (cardioides, rosas, espirales, círculos)
- Área y longitud de arco en coordenadas polares

Electivo

- Secciones cónicas en coordenadas polares (excentricidad y directriz)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la derivada dy/dx y encontrar rectas tangentes a una curva paramétrica [Usar]
2. Calcular la longitud de arco de una curva paramétrica [Usar]
3. Convertir puntos y ecuaciones entre coordenadas polares y cartesianas [Usar]
4. Graficar curvas polares comunes identificando simetrías y valores especiales [Usar]
5. Calcular el área de una región delimitada por una curva polar [Evaluar]

Elective:

6. Identificar y clasificar secciones cónicas dadas en forma polar [Familiarizarse]

4.1.10. CAN/Cálculo Diferencial Multivariable (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cálculo diferencial de funciones escalares de varias variables: límites, derivadas parciales, gradiente, plano tangente, optimización y teorema de Taylor.

Temas:
Core

- Límites y continuidad de funciones de varias variables en $\mathbb{R}^n$
- Derivadas parciales, derivadas direccionales y el vector gradiente
- Plano tangente, diferencial total y aproximación lineal
- Extremos de funciones multivariables y multiplicadores de Lagrange
- Teorema de Taylor para funciones de varias variables

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar límites y determinar la continuidad de funciones de varias variables [Familiarizarse]
2. Calcular gradientes, derivadas direccionales y planos tangentes para funciones multivariables [Usar]
3. Aplicar multiplicadores de Lagrange para resolver problemas de optimización con restricciones [Usar]
4. Expandir funciones de varias variables usando el teorema de Taylor y estimar cotas de error [Evaluar]

4.1.11. CAN/Integrales Múltiples (Core Tier1: 1 hr, Core Tier2: 1 hr)

Integración de funciones de varias variables: integrales dobles y triples en coordenadas cartesianas, polares, cilíndricas y esféricas, con aplicaciones.
Temas:
Core

- Integrales dobles sobre rectángulos y regiones generales; integrales iteradas (teorema de Fubini)
- Integrales dobles en coordenadas polares y fórmula de cambio de variable
- Integrales triples en coordenadas cartesianas y aplicaciones a volumen y masa
- Integrales triples en coordenadas cilíndricas y esféricas
- Aplicaciones: área, volumen, centros de masa y momentos de inercia

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar integrales dobles y triples sobre regiones generales usando integrales iteradas [Usar]
2. Aplicar cambios de coordenadas en sistemas polares, cilíndricos y esféricos para simplificar integrales múltiples [Usar]
3. Calcular áreas, volúmenes y centros de masa usando integrales múltiples [Evaluar]

4.1.12. CAN/Integrales de Línea y Campos Vectoriales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Integrales de línea escalares y vectoriales, campos conservativos, funciones potencial, independencia de la trayectoria y el Teorema de Green.
Temas:
Core

- Campos vectoriales: definición, visualización, divergencia y rotacional
- Integrales de línea de funciones escalares: longitud de arco y masa
- Integrales de línea de campos vectoriales: trabajo y circulación
- Campos conservativos, funciones potencial e independencia de la trayectoria
- Teorema de Green y sus aplicaciones en el plano

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular integrales de línea escalares y vectoriales sobre curvas orientadas [Usar]
2. Determinar si un campo vectorial es conservativo y construir su función potencial [Usar]
3. Aplicar el Teorema de Green para evaluar integrales de línea mediante integrales dobles [Evaluar]

4.1.13. CAN/Integrales de Superficie y Teoremas Fundamentales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Superficies paramétricas, integrales de superficie escalares y de flujo, rotacional, divergencia y los teoremas fundamentales de Stokes y Gauss.

Temas:

Core

- Superficies paramétricas, plano tangente y área de superficie
- Integrales de superficie de funciones escalares
- Integrales de flujo: integrales de superficie de campos vectoriales
- Teorema de Stokes: relación entre integrales de línea e integrales de superficie
- Teorema de la Divergencia (Gauss): relación entre flujo e integral de volumen

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Parametrizar superficies y evaluar integrales de superficie escalares y de flujo [Usar]
2. Aplicar los Teoremas de Stokes y de la Divergencia para reducir integrales complejas [Evaluar]
3. Relacionar propiedades diferenciales locales (rotacional, divergencia) con el comportamiento global mediante teoremas integrales [Evaluar]

4.1.14. CAN/Funciones Vectoriales de una Variable Real (Core Tier1: 1 hr, Core Tier2: 1 hr)

Curvas en el espacio como funciones vectoriales de una variable real, incluyendo geometría diferencial de curvas: triedro de Frenet, curvatura y torsión.

Temas:

Core

- Definición, dominio y rango de funciones vectoriales de una variable real
- Límites, continuidad y derivadas de funciones vectoriales
- Longitud de arco y parametrización natural de una curva
- Vectores tangente, normal y binormal; triedro móvil de Frenet
- Curvatura, torsión y fórmulas de Frenet-Serret

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Identificar límites, continuidad y derivadas de funciones vectoriales de una variable real [Familiarizarse]
2. Calcular vectores tangente, normal y binormal y reparametrizar una curva por longitud de arco [Usar]
3. Calcular curvatura y torsión usando las fórmulas de Frenet-Serret [Evaluar]

4.1.15. CAN/Transformaciones Multivariables (Core Tier1: 1 hr, Core Tier2: 1 hr)

Funciones vectoriales de variables vectoriales: diferenciabilidad, matriz jacobiana, cambio de variable y los teoremas de la función inversa y de la función implícita.

Temas:

Core

- Transformaciones de $\mathbb{R}^n$ a $\mathbb{R}^m$ y su diferenciabilidad
- Matriz jacobiana y determinante jacobiano
- Cambio de variable en coordenadas polares, cilíndricas y esféricas
- Teorema de la función inversa
- Teorema de la función implícita en contextos multivariables

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir transformaciones entre sistemas de coordenadas usando la matriz jacobiana [Familiarizarse]
2. Calcular el jacobiano y aplicar cambio de variable para evaluar integrales múltiples [Usar]

3. Aplicar los teoremas de la función inversa y de la función implícita para analizar invertibilidad local [Evaluar]

4.1.16. CAN/Análisis Real e Integración de Lebesgue (Core Tier1: 1 hr, Core Tier2: 1 hr)

Tratamiento riguroso de espacios métricos, teoría de la medida y la integral de Lebesgue, incluyendo espacios Lp y teoremas de convergencia.

Temas:

Core

- Espacios métricos: completitud, compacidad y el teorema de categoría de Baire
- Medida de Lebesgue en $\mathbb{R}^n$: conjuntos medibles y conjuntos nulos
- Teoremas de convergencia: Convergencia Monótona, Convergencia Dominada y Lema de Fatou
- Espacios L^p: definición, desigualdades de Hölder y Minkowski, y dualidad
- Convergencia uniforme e intercambio de límite, diferenciación e integración

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Contrastar la integral de Lebesgue con la integral de Riemann e identificar su alcance relativo [Familiarizarse]
2. Demostrar propiedades básicas de funciones medibles usando los teoremas de convergencia [Usar]
3. Verificar criterios de completitud y compacidad en espacios métricos y L^p [Evaluar]

4.1.17. CAN/Análisis Complejo y Aplicaciones Conformes (Core Tier1: 1 hr, Core Tier2: 1 hr)

Funciones de variable compleja: analiticidad, integración de contorno, series de Laurent, residuos y aplicaciones conformes.

Temas:

Core

- Funciones holomorfas, ecuaciones de Cauchy-Riemann y funciones armónicas
- Teorema y fórmula integral de Cauchy; técnicas de integración de contorno
- Series de Laurent y clasificación de singularidades aisladas (evitable, polo, esencial)
- Teorema de los residuos y evaluación de integrales reales

Electivo

- Aplicaciones conformes, transformaciones de Möbius y el teorema de la aplicación de Riemann

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar la analiticidad de funciones complejas usando las ecuaciones de Cauchy-Riemann [Familiarizarse]
2. Evaluar integrales reales y sumar series usando el Teorema de los Residuos [Usar]

Elective:

3. Construir aplicaciones conformes entre dominios estándar usando transformaciones de Möbius [Evaluar]

4.1.18. CAN/Análisis Funcional y Espacios de Banach/Hilbert

Espacios normados de dimensión infinita, operadores lineales acotados y teoría espectral, con aplicaciones a EDPs y mecánica cuántica.

Temas:

Electivo

- Espacios vectoriales normados, espacios de Banach y el teorema de Hahn-Banach
- Espacios de Hilbert, productos internos, bases ortonormales y el Teorema de Representación de Riesz
- Operadores lineales acotados y compactos; espacios duales
- Teorema de la aplicación abierta, teorema del gráfico cerrado y principio de acotación uniforme
- Teoría espectral de operadores compactos autoadjuntos y la alternativa de Fredholm

Aprendizaje esperado (*Learning Outcomes*):
Elective:

1. Explicar la completitud en espacios de Banach y Hilbert y dar ejemplos canónicos de cada uno [Familiarizarse]
2. Aplicar el Teorema de Representación de Riesz para identificar espacios duales de espacios de Hilbert [Usar]
3. Analizar el espectro de un operador compacto y caracterizar su estructura de valores propios [Evaluar]

4.1.19. CAN/Análisis Armónico y Teoría de Fourier

Series y transformadas de Fourier, teoría L^2, distribuciones y wavelets, con aplicaciones al procesamiento de señales y EDPs.

Temas:
Electivo

- Series de Fourier: convergencia, teorema de Parseval y completitud L^2
- La transformada de Fourier en L^1 y L^2: propiedades, inversión y convolución
- Teorema de Plancherel y distribuciones temperadas (espacio de Schwartz)
- Teoría de wavelets: análisis multirresolución y la transformada wavelet discreta
- Aplicaciones al procesamiento de señales, resolución de EDPs y compresión de datos

Aprendizaje esperado (*Learning Outcomes*):
Elective:

1. Descomponer funciones periódicas en series de Fourier y evaluar la convergencia puntual y en L^2 [Familiarizarse]
2. Aplicar la transformada de Fourier para resolver EDPs lineales y calcular convoluciones [Usar]
3. Analizar la tasa de decaimiento de los coeficientes de Fourier en relación con la suavidad de una función [Evaluar]

4.1.20. CAN/Cálculo de Variaciones

Optimización de funcionales: ecuaciones de Euler-Lagrange, condiciones de segundo orden, restricciones isoperimétricas y métodos directos.

Temas:
Electivo

- Ecuaciones de Euler-Lagrange y problemas extremales clásicos (braquistócrona, geodésicas)
- Restricciones isoperimétricas y multiplicadores de Lagrange en espacios funcionales
- Segunda variación y condiciones suficientes para mínimos locales
- Formulaciones débiles y espacios de Sobolev en problemas variacionales
- Métodos directos en el cálculo de variaciones: existencia mediante coercitividad y semicontinuidad inferior

Aprendizaje esperado (*Learning Outcomes*):
Elective:

1. Enunciar la ecuación de Euler-Lagrange y derivarla para funcionales estándar [Familiarizarse]
2. Derivar ecuaciones de movimiento para sistemas mecánicos usando el principio de mínima acción [Usar]
3. Resolver problemas variacionales clásicos (braquistócrona, isoperimétrico) usando la teoría de Euler-Lagrange [Evaluar]

4.1.21. CAN/Teoría de la Medida (Core Tier1: 1 hr, Core Tier2: 1 hr)

Teoría abstracta de la medida e integración: sigma-álgebras, medidas abstractas, espacios producto y el teorema de Radon-Nikodym.

Temas:
Core

- Sigma-álgebras, espacios medibles y conjuntos de Borel
- Medidas abstractas, medidas exteriores y el teorema de extensión de Carathéodory

- Medidas producto y teorema de Fubini-Tonelli
- Teorema de Radon-Nikodym, continuidad absoluta y descomposición de medidas
- Medidas con signo, descomposición de Hahn y descomposición de Jordan

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Definir sigma-álgebras y enunciar las condiciones para que una función de conjuntos sea una medida [Familiarizarse]
2. Aplicar el teorema de Fubini-Tonelli para evaluar integrales iteradas en espacios producto [Usar]
3. Demostrar el teorema de Radon-Nikodym y aplicarlo para derivar reglas de cadena para densidades [Evaluar]

4.2. álgebra y Teoría de Números (ANT)

Esta área cubre las estructuras algebraicas subyacentes a las matemáticas modernas y sus aplicaciones, desde los fundamentos del álgebra lineal y la teoría de grupos hasta la aritmética de los cuerpos de números, la codificación algebraica y la criptografía.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.2.1 Matrices y Sistemas de Ecuaciones Lineales (Pág. 190)	1	1
⭐ 4.2.2 Espacios Vectoriales y Transformaciones Lineales (Pág. 191)	1	1
⭐ 4.2.3 Valores Propios, Producto Interno y Formas Cuadráticas (Pág. 191)	1	1
⭐ 4.2.4 Teoría de Grupos (Pág. 191)	1	1
⭐ 4.2.5 Anillos, Cuerpos y Teoría de Galois (Pág. 192)	1	1
4.2.6 Teoría de Módulos y álgebra Conmutativa (Pág. 192)	Electivo	
4.2.7 Teoría Analítica de Números (Pág. 192)	Electivo	
4.2.8 Teoría Algebraica de Números (Pág. 193)	Electivo	
4.2.9 Teoría de Representaciones (Pág. 193)	Electivo	
⭐ 4.2.10 Criptografía y Teoría de la Codificación (Pág. 193)	1	1

Cuadro 4.2: Lista de KUs del área de álgebra y Teoría de Números.

4.2.1. ANT/Matrices y Sistemas de Ecuaciones Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr)

álgebra de matrices, sistemas de ecuaciones lineales y sus métodos de resolución, determinantes y descomposiciones matriciales.
Temas:
Core

- álgebra de matrices: suma, multiplicación por escalar, producto de matrices, transpuesta, tipos de matrices e inversa de una matriz
- Resolución de sistemas de ecuaciones lineales mediante eliminación gaussiana y Gauss-Jordan; forma escalonada, rango y consistencia de sistemas
- Determinantes: definición, propiedades, regla de Cramer e interpretación geométrica
- Descomposiciones matriciales: LU, QR y descomposición en valores singulares (SVD)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar tipos de matrices y realizar operaciones básicas de álgebra matricial, incluyendo el cálculo de la inversa de una matriz [Familiarizarse]
2. Resolver sistemas de ecuaciones lineales mediante eliminación gaussiana o Gauss-Jordan, determinando el rango y la consistencia del sistema [Usar]
3. Aplicar la regla de Cramer y las descomposiciones LU, QR y SVD para resolver sistemas lineales y factorizar matrices [Evaluar]

4.2.2. ANT/Espacios Vectoriales y Transformaciones Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Espacios vectoriales, subespacios, bases y aplicaciones lineales entre espacios vectoriales.

Temas:

Core

- Espacios vectoriales y subespacios; bases, dimensión e independencia lineal
- Aplicaciones lineales, núcleo e imagen, teorema de rango-nulidad y representaciones matriciales

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Identificar si un conjunto con operaciones dadas forma un espacio vectorial y determinar una base [Familiarizarse]
2. Determinar el núcleo, la imagen y la representación matricial de una aplicación lineal, verificando el teorema de rango-nulidad [Usar]
3. Aplicar transformaciones lineales para modelar y resolver problemas de cambio de base y composición de aplicaciones [Evaluar]

4.2.3. ANT/Valores Propios, Producto Interno y Formas Cuadráticas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Valores propios, diagonalización, espacios con producto interno y formas cuadráticas, con aplicaciones a la optimización y aproximación de datos.

Temas:

Core

- Valores propios y vectores propios; diagonalización y el teorema espectral
- Espacios con producto interno, ortogonalidad, proceso de Gram-Schmidt y mínimos cuadrados
- Formas cuadráticas y definitud: clasificación en positiva, negativa e indefinida mediante valores propios

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Identificar los valores propios y vectores propios de una matriz y determinar si es diagonalizable [Familiarizarse]
2. Calcular la diagonalización de matrices mediante el teorema espectral y aplicar el proceso de Gram-Schmidt para obtener bases ortogonales [Usar]
3. Aplicar los métodos de mínimos cuadrados y la clasificación de formas cuadráticas mediante valores propios para resolver problemas de aproximación y optimización [Evaluar]

4.2.4. ANT/Teoría de Grupos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Grupos, subgrupos, homomorfismos, grupos cocientes y la clasificación de grupos finitos.

Temas:

Core

- Grupos, subgrupos, grupos cíclicos y orden de los elementos
- Clases laterales, teorema de Lagrange y subgrupos normales
- Homomorfismos de grupos, isomorfismos y los teoremas de isomorfía
- Acciones de grupo, órbitas, estabilizadores y lema de Burnside
- Teoremas de Sylow y clasificación de grupos finitos de orden pequeño

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar el teorema de Lagrange y explicar la relación entre subgrupos y clases laterales [Familiarizarse]
2. Aplicar los teoremas de isomorfía para analizar imágenes homomórficas y grupos cocientes [Usar]
3. Usar los teoremas de Sylow para determinar la estructura y el número de subgrupos de un grupo finito [Evaluar]

4.2.5. ANT/Anillos, Cuerpos y Teoría de Galois (Core Tier1: 1 hr, Core Tier2: 1 hr)

Anillos, ideales, anillos de polinomios, extensiones de cuerpos y la correspondencia de Galois entre extensiones de cuerpos y grupos de permutaciones.

Temas:

Core

- Anillos, ideales, anillos cociente y los teoremas de isomorfía para anillos
- Anillos de polinomios: irreducibilidad, factorización y dominios de factorización única
- Extensiones de cuerpos: elementos algebraicos y trascendentes, grado y cuerpos de descomposición
- Grupos de Galois, la correspondencia de Galois y solubilidad por radicales
- Cuerpos finitos: estructura, existencia y aplicaciones en teoría de codificación

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Clasificar anillos como dominios enteros, DIPs o DFUs y justificar la clasificación [Familiarizarse]
2. Determinar el grupo de Galois de un polinomio y aplicar la correspondencia de Galois [Usar]
3. Demostrar que las ecuaciones quínticas generales no son solubles por radicales usando la teoría de Galois [Evaluar]

4.2.6. ANT/Teoría de Módulos y álgebra Conmutativa

Módulos sobre anillos, módulos libres y proyectivos, el teorema de estructura para módulos finitamente generados y álgebra conmutativa.

Temas:

Non Core

- Módulos sobre anillos: definiciones, submódulos, cocientes y homomorfismos
- Módulos libres y proyectivos; sucesiones exactas y sucesiones escindidas
- Teorema de estructura para módulos finitamente generados sobre un DIP (forma normal de Jordan)
- Productos tensoriales de módulos: definición, propiedades y aplicaciones
- Localización, ideales primos y maximales, anillos noetherianos y teorema de la base de Hilbert

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Explicar el teorema de estructura para módulos finitamente generados sobre un DIP y derivar la forma normal de Jordan [Familiarizarse]
2. Construir productos tensoriales y aplicarlos a problemas de álgebra multilineal [Usar]
3. Analizar un anillo conmutativo identificando su espectro primo y aplicando localización [Evaluar]

4.2.7. ANT/Teoría Analítica de Números

Distribución de números primos, series de Dirichlet, la función zeta de Riemann y funciones L.

Temas:

Non Core

- Funciones aritméticas, la función contadora de primos y el Teorema de los Números Primos
- Series de Dirichlet: convergencia, producto de Euler y funciones multiplicativas
- La función zeta de Riemann: continuación analítica, ecuación funcional y la banda crítica
- Caracteres de Dirichlet, funciones L y primos en progresiones aritméticas
- Métodos de criba: Eratóstenes, Legendre y la criba grande

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Enunciar el Teorema de los Números Primos y describir el papel de la función zeta de Riemann en su demostración [Familiarizarse]
2. Aplicar series de Dirichlet y funciones multiplicativas para derivar estimaciones asintóticas [Usar]
3. Analizar la distribución de primos en progresiones aritméticas usando el teorema de Dirichlet [Evaluar]

4.2.8. ANT/Teoría Algebraica de Números

Cuerpos de números, anillos de enteros, factorización de ideales, grupos de clases y geometría de los números.

Temas:

Non Core

- Cuerpos de números: enteros algebraicos, anillos de enteros y sus propiedades básicas
- Factorización única de ideales en dominios de Dedekind
- Grupo de clases de ideales, teorema de Minkowski y finitud del número de clases
- Unidades, teorema de las unidades de Dirichlet y ramificación de primos
- Números p-ádicos y valoraciones; cuerpos locales

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Describir el anillo de enteros de un cuerpo de números y explicar por qué la factorización única de elementos puede fallar [Familiarizarse]
2. Calcular el grupo de clases y el grupo de unidades de cuerpos cuadráticos usando la cota de Minkowski [Usar]
3. Aplicar la factorización de ideales en dominios de Dedekind para resolver ecuaciones diofánticas [Evaluar]

4.2.9. ANT/Teoría de Representaciones

Representaciones de grupos finitos, caracteres, lema de Schur, tabla de caracteres y aplicaciones a la física y la combinatoria.

Temas:

Non Core

- Representaciones lineales de grupos: definiciones, subrepresentaciones y sumas directas
- Lema de Schur, representaciones irreducibles y teorema de Maschke
- Caracteres, tablas de caracteres, relaciones de ortogonalidad y número de representaciones irreducibles
- Representaciones inducidas, reciprocidad de Frobenius y teorema de Mackey
- Aplicaciones a la física (simetría molecular), combinatoria y teoría de representaciones modulares

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Explicar el lema de Schur y el teorema de Maschke y su papel en la descomposición de representaciones [Familiarizarse]
2. Construir la tabla de caracteres de un grupo finito y verificar las relaciones de ortogonalidad [Usar]
3. Aplicar la teoría de representaciones para determinar los modos vibracionales de una molécula simétrica [Evaluar]

4.2.10. ANT/Criptografía y Teoría de la Codificación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fundamentos matemáticos de la criptografía moderna y los códigos correctores de errores, incluyendo RSA, curvas elípticas y códigos lineales.

Temas:

Core

- Aritmética modular, criptosistema RSA y problema del logaritmo discreto
- Curvas elípticas sobre cuerpos finitos y criptografía de curvas elípticas (ECC)
- Códigos lineales: matrices generadora y de verificación de paridad, distancia de Hamming y corrección de errores
- Códigos cíclicos: representación polinomial, códigos BCH y códigos Reed-Solomon

Non Core

- Criptografía postcuántica: esquemas basados en retículos y el problema Learning With Errors (LWE)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las hipótesis de dureza matemática subyacentes a RSA y la criptografía de curvas elípticas [Familiarizarse]
2. Construir un código lineal, calcular sus parámetros y determinar su capacidad de corrección de errores [Usar]

NonCore:

3. Analizar los supuestos de seguridad de los esquemas criptográficos basados en retículos en el contexto postcuántico [Evaluar]

4.3. Geometría y Topología (GTO)

Esta área cubre el estudio de la forma, el espacio y la continuidad, desde la geometría euclidiana clásica hasta la teoría moderna de variedades, la topología algebraica y la geometría simpléctica, con profundas conexiones con la física y la ciencia de datos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.3.1 Geometría Euclidiana y Proyectiva (Pág. 194)	1	1
⭐ 4.3.2 Topología General (Pág. 194)	1	1
⭐ 4.3.3 Geometría Diferencial de Curvas y Superficies (Pág. 195)	1	1
⭐ 4.3.4 Variedades Diferenciables y Formas Diferenciales (Pág. 195)	1	1
4.3.5 Geometría Riemanniana (Pág. 196)	Electivo	
4.3.6 Topología Algebraica (Pág. 196)	Electivo	
4.3.7 Geometría Simpléctica (Pág. 196)	Electivo	
4.3.8 Teoría Geométrica de Grupos (Pág. 197)	Electivo	

Cuadro 4.3: Lista de KUs del área de Geometría y Topología.

4.3.1. GTO/Geometría Euclidiana y Proyectiva (Core Tier1: 1 hr, Core Tier2: 1 hr)

Geometría euclidiana axiomática, isometrías, grupos de simetría y el plano proyectivo.

Temas:
Core

- Geometría euclidiana: axiomas, congruencia, semejanza y teoremas clásicos
- Isometrías del plano y el espacio: reflexiones, rotaciones, traslaciones y reflexiones con deslizamiento
- Grupos de simetría: grupos de friso, grupos de papel tapiz y clasificación de grupos finitos de isometrías
- Plano proyectivo: coordenadas homogéneas, dualidad y razón cruzada
- Secciones cónicas en geometría proyectiva: clasificación y transformaciones proyectivas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la base axiomática de la geometría euclidiana e identificar el papel del postulado de las paralelas [Familiarizarse]
2. Clasificar isometrías del plano y determinar el grupo de simetría de una figura geométrica dada [Usar]
3. Aplicar coordenadas proyectivas y dualidad para demostrar teoremas clásicos (Pappus, Desargues) [Evaluar]

4.3.2. GTO/Topología General (Core Tier1: 1 hr, Core Tier2: 1 hr)

Espacios topológicos, continuidad, compacidad, conexidad, axiomas de separación y espacios cociente.

Temas:
Core
- Espacios topológicos: conjuntos abiertos y cerrados, bases y subbases
- Funciones continuas, homeomorfismos y equivalencia topológica
- Compacidad (Heine-Borel), conexidad y conexidad por caminos
- Axiomas de separación (T_1, Hausdorff, normal) y lema de Urysohn
- Espacios cociente, topología producto y teorema de Tychonoff

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Identificar si un espacio dado es compacto, conexo o Hausdorff usando criterios estándar [Familiarizarse]
2. Construir nuevos espacios topológicos mediante cocientes y productos y determinar sus propiedades topológicas [Usar]
3. Demostrar teoremas básicos sobre funciones continuas, incluyendo los teoremas del valor extremo y del valor intermedio en entornos topológicos [Evaluar]

4.3.3. GTO/Geometría Diferencial de Curvas y Superficies (Core Tier1: 1 hr, Core Tier2: 1 hr)

Teoría local y global de curvas y superficies en $\mathbb{R}^3$: curvatura, aplicación de Gauss y teorema de Gauss-Bonnet.
Temas:
Core
- Curvas en el espacio: curvatura, torsión, triedro de Frenet-Serret y el teorema fundamental de curvas
- Superficies regulares en $\mathbb{R}^3$: parametrizaciones, planos tangentes y primera forma fundamental
- Segunda forma fundamental, curvaturas principales, curvatura gaussiana y media, y la aplicación de Gauss
- Geodésicas en superficies y el teorema de Gauss-Bonnet
- Teorema Egregium de Gauss: naturaleza intrínseca de la curvatura gaussiana

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Calcular la curvatura y torsión de una curva en el espacio a partir de sus ecuaciones paramétricas [Familiarizarse]
2. Calcular la curvatura gaussiana y media de superficies paramétricas usando la primera y segunda formas fundamentales [Usar]
3. Aplicar el teorema de Gauss-Bonnet para relacionar la curvatura de una superficie con su característica de Euler topológica [Evaluar]

4.3.4. GTO/Variedades Diferenciables y Formas Diferenciales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Variedades diferenciables, fibrados tangentes, formas diferenciales, cálculo exterior y cohomología de de Rham.
Temas:
Core
- Variedades diferenciables: cartas, atlas, aplicaciones diferenciables y el teorema de la función implícita
- Fibrados tangente y cotangente; campos vectoriales y sus flujos
- Formas diferenciales, producto cuña, derivada exterior y teorema de Stokes en variedades
- Cohomología de de Rham y lema de Poincaré
- Grupos de Lie y álgebras de Lie: definiciones y ejemplos fundamentales

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Definir variedades diferenciables y aplicaciones diferenciables usando cartas y atlas [Familiarizarse]

2. Aplicar el teorema de Stokes en variedades para unificar los teoremas integrales clásicos del cálculo vectorial [Usar]
3. Calcular grupos de cohomología de de Rham de variedades estándar e interpretar su significado topológico [Evaluar]

4.3.5. GTO/Geometría Riemanniana

Métricas riemannianas, conexiones, geodésicas, tensores de curvatura y geometría de comparación.

Temas:

Non Core

- Métricas riemannianas: longitud, distancia y volumen en variedades
- Conexión de Levi-Civita, transporte paralelo y derivada covariante
- Geodésicas: existencia, minimalidad y aplicación exponencial
- Tensor de curvatura de Riemann, curvatura de Ricci y curvatura escalar
- Teoremas de comparación: Bonnet-Myers, Hadamard-Cartan y sus consecuencias topológicas

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Describir la conexión de Levi-Civita y explicar qué la distingue de una conexión arbitraria [Familiarizarse]
2. Calcular el tensor de curvatura de Riemann para espacios estándar (esfera, espacio hiperbólico) [Usar]
3. Aplicar teoremas de comparación para deducir consecuencias topológicas a partir de cotas de curvatura [Evaluar]

4.3.6. GTO/Topología Algebraica

Homotopía, grupo fundamental, espacios recubridores, homología y cohomología singular, y la sucesión de Mayer-Vietoris.

Temas:

Non Core

- Grupo fundamental, equivalencia homotópica y teorema de Van Kampen
- Espacios recubridores y su clasificación mediante el grupo fundamental
- Homología singular: complejos de cadenas, sucesiones exactas y la sucesión de Mayer-Vietoris
- Cohomología singular, el Teorema del Coeficiente Universal y productos copa
- Grupos de homotopía superiores y el teorema de Hurewicz

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Calcular el grupo fundamental de espacios estándar usando el teorema de Van Kampen [Familiarizarse]
2. Aplicar la sucesión de Mayer-Vietoris para calcular grupos de homología singular [Usar]
3. Distinguir espacios topológicos calculando y comparando sus anillos de homología y cohomología [Evaluar]

4.3.7. GTO/Geometría Simpléctica

Variedades simplécticas, sistemas hamiltonianos, subvariedades lagrangianas y conexiones con la mecánica clásica y la simetría especular.

Temas:

Electivo

- Variedades simplécticas y la forma simpléctica; teorema de Darboux
- Campos vectoriales hamiltonianos, corchetes de Poisson y flujos simplécticos
- Subvariedades lagrangianas y su papel en la cuantización y la simetría especular
- Aplicaciones momento y reducción simpléctica (Marsden-Weinstein)
- Geometría de contacto: variedades de contacto, campos vectoriales de Reeb y conexiones con la dinámica

Aprendizaje esperado (*Learning Outcomes*):
Elective:

1. Enunciar el teorema de Darboux y explicar por qué todas las variedades simplécticas de la misma dimensión son localmente equivalentes [Familiarizarse]
2. Formular sistemas hamiltonianos en variedades simplécticas y calcular flujos simplécticos [Usar]
3. Aplicar la reducción simpléctica para analizar sistemas mecánicos con simetrías [Evaluar]

4.3.8. GTO/Teoría Geométrica de Grupos

Grupos como objetos geométricos: grafos de Cayley, cuasi-isometría, métricas de palabras, grupos hiperbólicos y crecimiento.
Temas:
Electivo

- Grafos de Cayley, métrica de palabras y la geometría gruesa de grupos
- Cuasi-isometría: definición, ejemplos y lema de Švarc-Milnor
- Grupos hiperbólicos (Gromov): definición, triángulos delgados y frontera en el infinito
- Funciones de crecimiento de grupos: crecimiento polinomial (teorema de Gromov) y crecimiento exponencial
- Espacios CAT(0) y grupos que actúan en ellos

Aprendizaje esperado (*Learning Outcomes*):
Elective:

1. Construir el grafo de Cayley de un grupo finitamente generado y calcular su métrica de palabra [Familiarizarse]
2. Verificar cuasi-isometría entre grupos que actúan en el mismo espacio métrico mediante el lema de Švarc-Milnor [Usar]
3. Analizar la geometría de un grupo hiperbólico usando triángulos delgados y la frontera en el infinito [Evaluar]

4.4. Matemáticas Discretas y Combinatoria (DMC)

Esta área cubre las matemáticas de estructuras finitas y numerables, incluyendo la enumeración combinatoria, la teoría de grafos, la optimización discreta y los fundamentos lógicos de las matemáticas. Sustenta la informática teórica, la investigación de operaciones y la ciencia de datos moderna.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.4.1 Combinatoria Enumerativa (Pág. 197)	1	1
⭐ 4.4.2 Teoría de Grafos (Pág. 198)	1	1
4.4.3 Combinatoria Extremal y Probabilística (Pág. 198)	Electivo	
⭐ 4.4.4 Optimización Discreta (Pág. 198)	1	1
4.4.5 Teoría de Diseños y Códigos Combinatorios (Pág. 199)	Electivo	
⭐ 4.4.6 Lógica Matemática y Teoría de Conjuntos (Pág. 199)	1	1

Cuadro 4.4: Lista de KUs del área de Matemáticas Discretas y Combinatoria.

4.4.1. DMC/Combinatoria Enumerativa (Core Tier1: 1 hr, Core Tier2: 1 hr)

Conteo sistemático de estructuras discretas mediante biyecciones, recursiones, funciones generadoras y el principio de inclusión-exclusión.
Temas:
Core

- Principios básicos de conteo: permutaciones, combinaciones y el teorema del binomio
- Principio de inclusión-exclusión y aplicaciones a desarreglos y la función totiente de Euler
- Funciones generadoras ordinarias y exponenciales; resolución de recurrencias
- Particiones de enteros, tablas de Young y la fórmula de la longitud del gancho
- Demostraciones biyectivas e inversión de Möbius en conjuntos parcialmente ordenados

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Aplicar principios básicos de conteo e inclusión-exclusión a problemas de enumeración [Familiarizarse]
2. Construir funciones generadoras ordinarias y exponenciales para enumerar estructuras combinatorias [Usar]
3. Derivar fórmulas de forma cerrada para sucesiones combinatorias usando biyecciones e inversión de Möbius [Evaluar]

4.4.2. DMC/Teoría de Grafos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estructura y propiedades de grafos: conectividad, coloraciones, emparejamientos, planaridad y teoría espectral de grafos.
Temas:
Core

- Grafos: definiciones, isomorfismo, sucesiones de grados, árboles y árboles generadores
- Conectividad, teorema de Menger y flujos en redes (flujo máximo-corte mínimo)
- Emparejamientos (teorema de Hall), coloraciones de grafos y el polinomio cromático
- Grafos planares, fórmula de Euler, teorema de Kuratowski y el teorema de los cuatro colores
- Matrices de adyacencia y laplaciana, valores propios y grafos expandidores

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar propiedades estructurales de grafos (conectividad, planaridad, bipartitud) y aplicar la fórmula de Euler [Familiarizarse]
2. Aplicar el teorema de matrimonio de Hall y algoritmos de flujo en redes a problemas de emparejamiento y enrutamiento [Usar]
3. Analizar el espectro de un grafo para acotar su número cromático y sus propiedades de conectividad [Evaluar]

4.4.3. DMC/Combinatoria Extremal y Probabilística

Problemas extremales en grafos y sistemas de conjuntos, el método probabilístico y grafos aleatorios.
Temas:
Non Core

- Problemas de tipo Turán: teorema de Turán y teorema de Kruskal-Katona
- El método probabilístico: métodos del primer y segundo momento, Lema Local de Lovász
- Grafos aleatorios de Erdős-Rényi: fenómenos de umbral y estructura de componentes
- Teoría de Ramsey: números de Ramsey, teorema de van der Waerden y Hales-Jewett
- Métodos algebraicos en combinatoria: método polinomial y teorema de Chevalley-Warning

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Enunciar el teorema de Turán y aplicarlo para acotar el número de aristas en un grafo libre de K_r [Familiarizarse]
2. Aplicar el método probabilístico y el Lema Local de Lovász para demostrar la existencia de estructuras combinatorias [Usar]
3. Analizar fenómenos de umbral en grafos aleatorios de Erdős-Rényi usando métodos del primer y segundo momento [Evaluar]

4.4.4. DMC/Optimización Discreta (Core Tier1: 1 hr, Core Tier2: 1 hr)

Programación entera, flujos en redes, matroides, algoritmos de emparejamiento y teoría de NP-completitud.
Temas:
Core

- Programación lineal entera: ramificación y acotamiento, planos de corte y matrices totalmente unimodulares
- Algoritmos de flujo en redes: caminos más cortos, flujo máximo, flujo de costo mínimo
- Teoría de matroides: conjuntos independientes, algoritmo voraz e intersección de matroides
- Teoría de complejidad: P vs. NP, NP-completitud y reducciones
- Algoritmos de aproximación y esquema de aproximación polinomial (PTAS)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la relación entre las relajaciones de programación lineal y las cotas de programación entera [Familiarizarse]
2. Aplicar algoritmos de flujo en redes y de matroides para resolver problemas de optimización combinatoria [Usar]
3. Demostrar la NP-completitud de un problema combinatorio mediante reducción polinomial [Evaluar]

4.4.5. DMC/Teoría de Diseños y Códigos Combinatorios

Diseños combinatorios, cuadrados latinos, geometrías finitas y sus conexiones con los códigos correctores de errores.

Temas:
Non Core

- Diseños de bloques: diseños de bloques incompletos balanceados (BIBD), desigualdad de Fisher y diseños simétricos
- Cuadrados latinos, ortogonalidad y aplicaciones al diseño experimental
- Planos proyectivos y afines finitos; conexión con diseños combinatorios
- Matrices de Hadamard y sus aplicaciones en codificación y procesamiento de señales
- Conexiones entre diseños combinatorios y códigos lineales (Hamming, Golay)

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Definir un BIBD y verificar que una configuración dada satisface sus parámetros [Familiarizarse]
2. Construir un diseño balanceado o cuadrados latinos ortogonales para un conjunto de parámetros dado [Usar]
3. Relacionar la estructura de un diseño combinatorio con la matriz de verificación de paridad de un código lineal correspondiente [Evaluar]

4.4.6. DMC/Lógica Matemática y Teoría de Conjuntos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Lógica proposicional y de primer orden, sistemas de demostración, teoremas de completitud e incompletitud de Gödel y teoría axiomática de conjuntos.

Temas:
Core

- Lógica proposicional: sintaxis, semántica, satisfacibilidad y resolución
- Lógica de primer orden: modelos, corrección y teorema de completitud de Gödel
- Teoremas de incompletitud de Gödel y sus implicaciones para los sistemas formales
- Teoría axiomática de conjuntos (ZFC): ordinales, cardinales y el axioma de elección
- Computabilidad: máquinas de Turing, decidibilidad y el problema de la parada

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar los teoremas de completitud y primer teorema de incompletitud de Gödel y explicar su significado [Familiarizarse]
2. Construir demostraciones formales en lógica de primer orden y determinar la satisfacibilidad de fórmulas [Usar]
3. Analizar la indecidibilidad de problemas mediante reducción al problema de la parada [Evaluar]

4.5. Probabilidad y Estadística (PST)

Esta área cubre la teoría matemática de la incertidumbre y la inferencia, desde los fundamentos axiomáticos de la probabilidad hasta la metodología estadística moderna, incluyendo inferencia bayesiana, estadística de alta dimensión y análisis de series temporales.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.5.1 Axiomas de Probabilidad y Distribuciones (Pág. 200)	1	1
⭐ 4.5.2 Teoremas Límite (Pág. 200)	1	1
⭐ 4.5.3 Procesos Estocásticos (Pág. 201)	1	1
⭐ 4.5.4 Teoría de la Estimación (Pág. 201)	1	1
⭐ 4.5.5 Contraste de Hipótesis (Pág. 201)	1	1
⭐ 4.5.6 Inferencia Bayesiana (Pág. 202)	1	1
4.5.7 Estadística de Alta Dimensión (Pág. 202)	Electivo	
⭐ 4.5.8 Análisis de Series Temporales (Pág. 202)	1	1
⭐ 4.5.9 Métodos No Paramétricos (Pág. 203)	1	1

Cuadro 4.5: Lista de KUs del área de Probabilidad y Estadística.

4.5.1. PST/Axiomas de Probabilidad y Distribuciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fundamentación axiomática de Kolmogorov de la probabilidad, variables aleatorias, distribuciones y esperanzas.

Temas:

Core

- Axiomas de Kolmogorov, sigma-álgebras y espacios de probabilidad
- Variables aleatorias: distribuciones discretas y continuas, CDF, PDF y PMF
- Esperanza, varianza, momentos y funciones generadoras de momentos
- Distribuciones conjuntas, marginales y condicionales; independencia y covarianza
- Familias estándar: Binomial, Poisson, Normal, Exponencial, Gamma y Beta

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar los axiomas de Kolmogorov y construir un espacio de probabilidad para un experimento estándar [Familiarizarse]
2. Calcular esperanzas, varianzas y funciones generadoras de momentos para distribuciones estándar [Usar]
3. Determinar distribuciones conjuntas y condicionales y evaluar la independencia de variables aleatorias [Evaluar]

4.5.2. PST/Teoremas Límite (Core Tier1: 1 hr, Core Tier2: 1 hr)

Leyes de los grandes números, teorema central del límite, modos de convergencia y grandes desviaciones.

Temas:

Core

- Modos de convergencia: casi segura, en probabilidad, en L^p y en distribución
- Leyes débiles y fuertes de los grandes números y sus aplicaciones
- Teorema Central del Límite: enunciado, demostración mediante funciones características y cota de Berry-Esseen
- Funciones características y el teorema de continuidad (Lévy)
- Grandes desviaciones: teorema de Cramér y teorema de Gärtner-Ellis

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Distinguir entre los modos de convergencia de variables aleatorias y proporcionar contraejemplos que muestren sus diferencias [Familiarizarse]

2. Aplicar el Teorema Central del Límite para aproximar distribuciones de sumas y medias [Usar]
3. Demostrar la ley fuerte de los grandes números y estimar probabilidades de grandes desviaciones usando el teorema de Cramér [Evaluar]

4.5.3. PST/Procesos Estocásticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cadenas de Markov, procesos de Poisson, movimiento browniano, martingalas y cálculo estocástico.

Temas:

Core

- Cadenas de Markov en tiempo discreto: clasificación de estados, distribuciones estacionarias y teorema ergódico
- Procesos de Poisson: definición, propiedades y generalizaciones (compuesto, no homogéneo)
- Movimiento browniano: definición, propiedades de las trayectorias y principio de reflexión
- Martingalas, tiempos de parada, teorema de muestreo opcional y desigualdades de Doob
- Integral de Itô, fórmula de Itô y ecuaciones diferenciales estocásticas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Clasificar estados de una cadena de Markov y determinar su distribución estacionaria [Familiarizarse]
2. Aplicar la teoría de martingalas y el teorema de muestreo opcional para resolver problemas de tiempos de parada [Usar]
3. Formular y resolver una ecuación diferencial estocástica usando la fórmula de Itô [Evaluar]

4.5.4. PST/Teoría de la Estimación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estimación puntual y por intervalos: suficiencia, insesgadez, eficiencia, máxima verosimilitud y teoría asintótica.

Temas:

Core

- Estadísticos suficientes, teorema de factorización y completitud
- Estimadores insesgados, cota de Cramér-Rao y UMVUE
- Estimación por máxima verosimilitud: propiedades, consistencia y normalidad asintótica
- Intervalos de confianza: cantidades pivote, razón de verosimilitud y métodos bootstrap
- Teoría asintótica: método delta, M-estimadores y matriz de información de Fisher

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar la cota de Cramér-Rao e identificar cuándo existe un UMVUE mediante el teorema de Lehmann-Scheffé [Familiarizarse]
2. Derivar estimadores de máxima verosimilitud y calcular su varianza asintótica [Usar]
3. Construir intervalos de confianza usando cantidades pivote y remuestreo bootstrap [Evaluar]

4.5.5. PST/Contraste de Hipótesis (Core Tier1: 1 hr, Core Tier2: 1 hr)

Marco de Neyman-Pearson, pruebas uniformemente más potentes, pruebas de razón de verosimilitud y contrastes múltiples.

Temas:

Core

- Lema de Neyman-Pearson: pruebas más potentes para hipótesis simples
- Pruebas uniformemente más potentes (UMP) y familias de razón de verosimilitud monótona
- Pruebas de razón de verosimilitud generalizada y teorema de Wilks
- p-valores, contrastes múltiples y control de la tasa de falsos descubrimientos (FDR)
- Pruebas no paramétricas: prueba de los signos, suma de rangos de Wilcoxon y Kolmogorov-Smirnov

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar el lema de Neyman-Pearson y explicar el equilibrio entre errores de tipo I y tipo II [Familiarizarse]
2. Aplicar la prueba de razón de verosimilitud y el teorema de Wilks a problemas de hipótesis compuestas [Usar]
3. Controlar la tasa de falsos descubrimientos en un escenario de contrastes múltiples usando el procedimiento de Benjamini-Hochberg [Evaluar]

4.5.6. PST/Inferencia Bayesiana (Core Tier1: 1 hr, Core Tier2: 1 hr)

Paradigma bayesiano: distribuciones a priori y a posteriori, familias conjugadas, modelos jerárquicos y métodos computacionales.

Temas:

Core

- Teorema de Bayes en entornos continuos; distribuciones a priori y a posteriori
- Familias conjugadas a priori: Beta-Binomial, Dirichlet-Multinomial, Normal-Normal
- Modelos bayesianos jerárquicos y Bayes empírico
- Métodos de Monte Carlo de Cadenas de Markov (MCMC): Metropolis-Hastings y muestreo de Gibbs
- Inferencia variacional y la cota inferior de la evidencia (ELBO)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar el teorema de Bayes para actualizar una distribución a priori con datos observados y obtener una distribución a posteriori [Familiarizarse]
2. Construir un modelo bayesiano jerárquico y derivar sus distribuciones conjunta y condicionales [Usar]
3. Implementar muestreo de Gibbs o Metropolis-Hastings para aproximar una distribución a posteriori [Evaluar]

4.5.7. PST/Estadística de Alta Dimensión

Teoría estadística en altas dimensiones: desigualdades de concentración, esparsidad, LASSO y teoría de matrices aleatorias.

Temas:

Non Core

- Desigualdades de concentración: cotas de Markov, Chebyshev, Chernoff y variables sub-Gaussianas
- Estimación dispersa: LASSO, regresión ridge y la propiedad de isometría restringida (RIP)
- Teoría de matrices aleatorias: ley del semicírculo de Wigner y distribución de Marchenko-Pastur
- Muestreo comprimido: condiciones de recuperación exacta y relajación convexa
- Cotas inferiores minimax y el equilibrio sesgo-varianza en altas dimensiones

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Aplicar desigualdades de concentración para acotar la desviación de sumas de variables aleatorias [Familiarizarse]
2. Analizar las propiedades estadísticas de los estimadores LASSO bajo condiciones de esparsidad y RIP [Usar]
3. Derivar cotas inferiores minimax para problemas de estimación usando métodos de teoría de la información [Evaluar]

4.5.8. PST/Análisis de Series Temporales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Procesos estacionarios, modelos ARMA, análisis espectral, modelos de espacio de estados y predicción.

Temas:

Core

- Estacionariedad, función de autocovarianza y función de autocorrelación (ACF)
- Modelos ARMA: identificación, estimación (Yule-Walker, MLE) y diagnóstico
- Densidad espectral, el periodograma y el teorema de Wiener-Khinchin
- Modelos de espacio de estados y el filtro de Kalman
- Modelos ARIMA, ajuste estacional y predicción a múltiples pasos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar la estacionariedad y determinar el orden de un modelo ARMA a partir de gráficos ACF y PACF [Familiarizarse]
2. Ajustar modelos ARIMA a datos de series temporales, validar residuos y producir predicciones [Usar]
3. Aplicar el filtro de Kalman para estimar estados ocultos en un modelo lineal gaussiano de espacio de estados [Evaluar]

4.5.9. PST/Métodos No Paramétricos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Inferencia libre de distribución, pruebas basadas en rangos, estimación de densidad por kernel y regresión no paramétrica.
Temas:
Core

- Función de distribución empírica, teorema de Glivenko-Cantelli y bootstrap
- Pruebas basadas en rangos: Wilcoxon, Kruskal-Wallis y correlación de Spearman
- Estimación de densidad por kernel: selección de ancho de banda, equilibrio sesgo-varianza
- Regresión no paramétrica: estimador de Nadaraya-Watson y splines de suavizado
- Pruebas de permutación e inferencia por aleatorización

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las ventajas y limitaciones de los métodos no paramétricos en comparación con las alternativas paramétricas [Familiarizarse]
2. Aplicar estimación de densidad por kernel y seleccionar el ancho de banda mediante validación cruzada [Usar]
3. Realizar pruebas de permutación y basadas en rangos e interpretar sus resultados para inferencias sin supuestos distributivos [Evaluar]

4.6. Análisis Numérico y Científico (NSA)

Esta área cubre el diseño, análisis e implementación de algoritmos para el cálculo numérico, incluyendo aritmética de punto flotante, sistemas lineales, aproximación, integración, ecuaciones diferenciales, optimización y computación de alto rendimiento.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.6.1 Análisis de Error y Aritmética de Punto Flotante (Pág. 204)	1	1
⭐ 4.6.2 Búsqueda de Raíces de Ecuaciones No Lineales (Pág. 204)	1	1
⭐ 4.6.3 álgebra Lineal Numérica (Pág. 204)	1	1
⭐ 4.6.4 Teoría de Aproximación e Interpolación (Pág. 205)	1	1
⭐ 4.6.5 Integración Numérica y Cuadratura (Pág. 205)	1	1
⭐ 4.6.6 Métodos Numéricos para Ecuaciones Diferenciales (Pág. 206)	1	1
⭐ 4.6.7 Método de Elementos Finitos (Pág. 206)	1	1
⭐ 4.6.8 Algoritmos de Optimización (Pág. 206)	1	1
⭐ 4.6.9 Computación Paralela y de Alto Rendimiento (Pág. 207)	1	1

Cuadro 4.6: Lista de KUs del área de Análisis Numérico y Científico.

4.6.1. NSA/Análisis de Error y Aritmética de Punto Flotante (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fuentes de error numérico, aritmética de punto flotante IEEE 754, redondeo, cancelación y análisis de error hacia adelante y hacia atrás.

Temas:

Core

- Representación de punto flotante IEEE 754: épsilon de la máquina, desbordamiento, subdesbordamiento y modos de redondeo
- Errores de redondeo, cancelación catastrófica y pérdida de significancia
- Análisis de error hacia adelante y hacia atrás; número de condición de un problema
- Estabilidad numérica: algoritmos estables vs. inestables; análisis de error hacia atrás de Wilkinson
- Propagación del error en operaciones aritméticas y evaluación de funciones

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar el estándar de punto flotante IEEE 754 e identificar las fuentes de errores de redondeo y cancelación [Familiarizarse]
2. Estimar el número de condición de un problema y predecir la precisión de un resultado numérico [Usar]
3. Analizar un algoritmo en cuanto a su estabilidad numérica utilizando cotas de error hacia adelante y hacia atrás [Evaluar]

4.6.2. NSA/Búsqueda de Raíces de Ecuaciones No Lineales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos iterativos para aproximar raíces de ecuaciones no lineales de una variable, su análisis de convergencia y su extensión a sistemas de ecuaciones no lineales.

Temas:

Core

- Método de bisección: acotamiento del intervalo, criterio de parada y cota del error
- Iteración de punto fijo y condiciones suficientes de convergencia
- Método de Newton-Raphson: derivación, convergencia cuadrática y casos de falla
- Método de la secante como alternativa al método de Newton sin derivada explícita
- Orden de convergencia y comparación de la eficiencia entre métodos iterativos
- Extensión del método de Newton a sistemas de ecuaciones no lineales mediante la matriz Jacobiana

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comparar las hipótesis y garantías de convergencia de los métodos de bisección, punto fijo, Newton-Raphson y secante [Familiarizarse]
2. Implementar el método de Newton-Raphson y el método de la secante para aproximar raíces de una función no lineal [Usar]
3. Analizar el orden de convergencia de un método iterativo dado y justificar la elección de método según el costo computacional y la disponibilidad de la derivada [Evaluar]

4.6.3. NSA/álgebra Lineal Numérica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Algoritmos para resolver sistemas lineales, problemas de mínimos cuadrados, problemas de valores propios y factorizaciones matriciales, con atención a la estabilidad y complejidad.

Temas:

Core

- Eliminación gaussiana con pivoteo parcial, factorización LU y análisis de complejidad
- Factorización QR mediante Householder y Gram-Schmidt; problemas de mínimos cuadrados
- Algoritmos para valores propios: iteración por potencias, algoritmo QR y método de Lanczos

- Descomposición en valores singulares: cálculo, truncamiento y aplicaciones (PCA, pseudoinversa)
- Métodos iterativos para sistemas grandes dispersos: CG, GMRES y precondicionamiento

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar solucionadores directos e iterativos y seleccionar el método apropiado para una estructura de matriz dada [Familiarizarse]
2. Aplicar factorización QR y SVD para resolver problemas de mínimos cuadrados y calcular aproximaciones de bajo rango [Usar]
3. Analizar la estabilidad numérica y el costo computacional de algoritmos para valores propios [Evaluar]

4.6.4. NSA/Teoría de Aproximación e Interpolación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Interpolación polinomial y por splines, mejor aproximación, polinomios de Chebyshev y aproximación racional.

Temas:
Core

- Interpolación polinomial: Lagrange, diferencias divididas de Newton y fenómeno de Runge
- Interpolación por splines: splines cúbicos, B-splines y métodos de funciones polinómicas a trozos
- Mejor aproximación en espacios normados: Chebyshev (minimax) y mínimos cuadrados
- Polinomios de Chebyshev: propiedades, ortogonalidad y convergencia espectral
- Aproximación trigonométrica y transformada rápida de Fourier (FFT)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar el fenómeno de Runge y justificar la elección de nodos de Chebyshev para mitigarlo [Familiarizarse]
2. Construir interpolantes por splines cúbicos y de Chebyshev para datos dados y estimar el error de interpolación [Usar]
3. Aplicar la FFT para calcular eficientemente aproximaciones trigonométricas de una función muestreada [Evaluar]

4.6.5. NSA/Integración Numérica y Cuadratura (Core Tier1: 1 hr, Core Tier2: 1 hr)

Reglas de cuadratura, integración adaptativa, cuadratura gaussiana e integración multidimensional.

Temas:
Core

- Reglas de Newton-Cotes: trapezoidal, Simpson y reglas compuestas; análisis de error
- Cuadratura adaptativa y control automático del error
- Cuadratura gaussiana: nodos y pesos óptimos, conexión con polinomios ortogonales
- Tratamiento numérico de integrales impropias y singulares
- Integración de Monte Carlo y métodos cuasi-Monte Carlo para integrales de alta dimensión

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Derivar la fórmula de error para la regla compuesta de Simpson e identificar su orden de precisión [Familiarizarse]
2. Seleccionar y aplicar una regla de cuadratura apropiada (Gaussiana, adaptativa) según la regularidad del integrando [Usar]
3. Aplicar integración de Monte Carlo para estimar integrales de alta dimensión y cuantificar el error estadístico [Evaluar]

4.6.6. NSA/Métodos Numéricos para Ecuaciones Diferenciales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos de diferencias finitas y Runge-Kutta para EDOs, análisis de estabilidad y métodos de diferencias finitas para EDPs.

Temas:

Core

- Métodos de Runge-Kutta: Euler, RK4 y métodos embebidos para control de error
- Métodos lineales multipaso: Adams-Bashforth, Adams-Moulton y métodos BDF
- Análisis de estabilidad: cero-estabilidad, regiones de estabilidad absoluta y EDOs rígidas
- Métodos de diferencias finitas para EDPs parabólicas y elípticas: estabilidad y convergencia (teorema de equivalencia de Lax)
- Problemas de valores en la frontera: método del disparo y discretización por diferencias finitas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comparar solucionadores de EDOs explícitos e implícitos y explicar su estabilidad relativa para problemas rígidos [Familiarizarse]
2. Implementar un método de Runge-Kutta con control de paso adaptativo y aplicarlo a un sistema de EDOs [Usar]
3. Analizar la estabilidad y convergencia de un esquema de diferencias finitas para una EDP parabólica usando el teorema de equivalencia de Lax [Evaluar]

4.6.7. NSA/Método de Elementos Finitos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Formulación variacional, métodos de Galerkin, espacios de elementos finitos, estimaciones de error y aplicaciones a EDPs elípticas.

Temas:

Core

- Formulación débil (variacional) de problemas de valor en la frontera y espacios de Sobolev
- Método de Galerkin: espacios de aproximación de dimensión finita y matriz de rigidez
- Espacios de elementos finitos triangulares y cuadriláteros: elementos de Lagrange y conformidad
- Estimaciones de error a priori: lema de Céa y cotas de error de interpolación

Non Core

- Estimaciones de error a posteriori y refinamiento adaptativo de malla

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Derivar la formulación débil de un PVE elíptico y mostrar su equivalencia con la forma fuerte [Familiarizarse]
2. Ensamblar la matriz de rigidez global y el vector de carga para una discretización lineal por elementos finitos [Usar]
3. Estimar el error en H^1 de una solución por elementos finitos usando el lema de Céa y la teoría de interpolación [Evaluar]

4.6.8. NSA/Algoritmos de Optimización (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos de gradiente, métodos de Newton y cuasi-Newton, optimización con restricciones y algoritmos de optimización convexa.

Temas:

Core

- Descenso por gradiente y métodos de búsqueda lineal: condiciones de Armijo-Wolfe y tasas de convergencia
- Método de Newton y métodos cuasi-Newton (BFGS, L-BFGS)
- Optimización con restricciones: condiciones KKT, métodos de penalización y programación cuadrática secuencial
- Algoritmos de optimización convexa: métodos de punto interior y método de multiplicadores de dirección alterna (ADMM)

- Descenso de gradiente estocástico (SGD), reducción de varianza y optimizador Adam

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar las garantías de convergencia del descenso por gradiente para funciones convexas suaves [Familiarizarse]
2. Aplicar los métodos de Newton y cuasi-Newton a problemas de optimización sin restricciones [Usar]
3. Formular un problema de optimización con restricciones como un sistema KKT y aplicar un método de punto interior [Evaluar]

4.6.9. NSA/Computación Paralela y de Alto Rendimiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Arquitecturas paralelas, modelos de rendimiento, algoritmos paralelos para álgebra lineal y computación con GPUs.
Temas:
Core

- Arquitecturas paralelas: memoria compartida (OpenMP), memoria distribuida (MPI) y GPU (CUDA)
- Modelos de rendimiento: ley de Amdahl, modelo de línea techada y complejidad de comunicación
- Algoritmos paralelos para álgebra lineal densa y dispersa (ScaLAPACK, PETSc)
- Descomposición de dominio y solucionadores paralelos de EDPs

Non Core

- Diferenciación automática (modo directo y reverso) y su rol en ML científico

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la ley de Amdahl e identificar los cuellos de botella que limitan la aceleración paralela en un algoritmo dado [Familiarizarse]
2. Implementar una rutina paralela de álgebra lineal usando MPI o OpenMP y medir la eficiencia paralela [Usar]

NonCore:

3. Aplicar diferenciación automática para calcular gradientes de funciones científicas complejas con fines de optimización [Evaluar]

4.7. Modelado y Simulación Matemática (MMS)

Esta área cubre las herramientas matemáticas para describir y simular fenómenos del mundo real, abarcando ecuaciones diferenciales ordinarias y parciales, sistemas dinámicos, métodos de perturbación, dinámica de fluidos, teoría de control y modelado basado en agentes.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.7.1 Ecuaciones Diferenciales Ordinarias de Primer Orden (Pág. 208)	1	1
⭐ 4.7.2 Ecuaciones Diferenciales de Orden Superior y Sistemas (Pág. 208)	1	1
⭐ 4.7.3 Ecuaciones en Derivadas Parciales (Pág. 208)	1	1
⭐ 4.7.4 Sistemas Dinámicos y Caos (Pág. 209)	1	1
⭐ 4.7.5 Análisis Asintótico y Métodos de Perturbación (Pág. 209)	1	1
4.7.6 Dinámica de Fluidos Matemática (Pág. 209)	Electivo	
⭐ 4.7.7 Teoría de Control Matemático (Pág. 210)	1	1
⭐ 4.7.8 Ecuaciones Diferenciales Estocásticas (Pág. 210)	1	1
4.7.9 Modelado Basado en Agentes y Sistemas Complejos (Pág. 211)	Electivo	

Cuadro 4.7: Lista de KUs del área de Modelado y Simulación Matemática.

4.7.1. MMS/Ecuaciones Diferenciales Ordinarias de Primer Orden (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos de solución, aplicaciones y análisis cualitativo de ecuaciones diferenciales ordinarias de primer orden: técnicas de solución exacta, modelado de fenómenos físicos y comportamiento cualitativo de soluciones.

Temas:

Core

- EDOs de primer orden: separables, lineales, ecuaciones exactas y factores integrantes
- Aplicaciones y modelado con EDOs de primer orden: crecimiento y decaimiento, problemas de mezcla, ley de enfriamiento de Newton y ley de Torricelli
- Análisis cualitativo de EDOs autónomas de primer orden: campos de direcciones, soluciones de equilibrio y estabilidad

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Clasificar EDOs de primer orden y seleccionar el método de solución apropiado (separación de variables, ecuación lineal, ecuación exacta o factor integrante) [Familiarizarse]
2. Formular y resolver modelos de primer orden para crecimiento y decaimiento, problemas de mezcla, la ley de enfriamiento de Newton y la ley de Torricelli [Usar]
3. Analizar cualitativamente EDOs autónomas de primer orden mediante campos de direcciones, e identificar puntos de equilibrio y su estabilidad [Evaluar]

4.7.2. MMS/Ecuaciones Diferenciales de Orden Superior y Sistemas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Teoría y métodos de solución para EDOs de orden superior y sistemas lineales: ecuaciones características, sistemas de valores propios, teoremas de existencia y unicidad, y métodos de transformada de Laplace.

Temas:

Core

- EDOs lineales de orden superior: ecuaciones características, variación de parámetros y coeficientes indeterminados
- Sistemas de EDOs lineales: exponencial de matriz, métodos de valores propios y retratos de fase
- Teoremas de existencia y unicidad (Picard-Lindelöf) y continuación de soluciones
- Métodos de transformada de Laplace para resolver EDOs con forzamiento discontinuo

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar el teorema de existencia y unicidad de Picard-Lindelöf e interpretar sus implicancias para la continuación de soluciones de una EDO de orden superior [Familiarizarse]
2. Resolver sistemas lineales de EDOs usando métodos de valores propios y esbozar sus retratos de fase [Usar]
3. Aplicar la transformada de Laplace para resolver EDOs con funciones de forzamiento a trozos e impulsivas [Evaluar]

4.7.3. MMS/Ecuaciones en Derivadas Parciales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Clasificación y solución de EDPs: ecuaciones de calor, onda y Laplace; características; métodos de Fourier; y soluciones débiles.

Temas:

Core

- Clasificación de EDPs lineales de segundo orden: elípticas, parabólicas e hiperbólicas
- Series de Fourier y separación de variables para las ecuaciones de calor y onda
- Funciones de Green para las ecuaciones de Laplace y Poisson
- Método de características para EDPs de primer orden e hiperbólicas; formación de choques

- Soluciones débiles, espacios de Sobolev y formulaciones variacionales de problemas elípticos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Clasificar una EDP de segundo orden e identificar la estrategia de solución apropiada [Familiarizarse]
2. Aplicar separación de variables y series de Fourier para resolver problemas de valor inicial y de frontera [Usar]
3. Derivar formulaciones débiles de EDPs elípticas y demostrar la existencia de soluciones usando el teorema de Lax-Milgram [Evaluar]

4.7.4. MMS/Sistemas Dinámicos y Caos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Análisis del plano de fase, estabilidad, bifurcaciones, ciclos límite, atractores extraños y caos en sistemas continuos y discretos.

Temas:

Core

- Retratos de fase de sistemas 2D: equilibrios, estabilidad y ciclos límite
- Teoría de estabilidad de Lyapunov y teorema de linealización (Hartman-Grobman)
- Teoría de bifurcaciones: silla-nodo, pitchfork, Hopf y bifurcaciones globales
- Caos: sistema de Lorenz, sensibilidad a condiciones iniciales y exponentes de Lyapunov
- Mapas en tiempo discreto: puntos fijos, duplicación de período y mapa logístico

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Esbozar retratos de fase de sistemas autónomos 2D y clasificar equilibrios por estabilidad [Familiarizarse]
2. Aplicar el método de Lyapunov y el teorema de Hartman-Grobman para evaluar la estabilidad de equilibrios [Usar]
3. Identificar tipos de bifurcación en sistemas parametrizados y rastrear la evolución de conjuntos invariantes [Evaluar]

4.7.5. MMS/Análisis Asintótico y Métodos de Perturbación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Expansiones asintóticas, teoría de perturbación regular y singular, capas límite y múltiples escalas.

Temas:

Core

- Expansiones asintóticas y símbolos de orden de Landau (O, o, $\sim$)
- Teoría de perturbación regular para ecuaciones algebraicas y EDOs
- Teoría de perturbación singular y expansiones asintóticas acopladas (capas límite)
- Método de múltiples escalas y aproximación WKB
- Método de Laplace y fase estacionaria para evaluación asintótica de integrales

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Construir una expansión de perturbación regular para una EDO con un parámetro pequeño [Familiarizarse]
2. Aplicar expansiones asintóticas acopladas para resolver problemas de perturbación singular con capas límite [Usar]
3. Usar el método de múltiples escalas para analizar oscilaciones de variación lenta y términos seculares [Evaluar]

4.7.6. MMS/Dinámica de Fluidos Matemática

Mecánica del continuo, ecuaciones de Euler y Navier-Stokes, vorticidad, flujo potencial y modelado de turbulencia.

Temas:

Non Core

- Hipótesis del continuo, derivada material y teorema del transporte de Reynolds
- Ecuaciones de Euler para flujo no viscoso: forma conservativa, vorticidad y flujo potencial
- Ecuaciones de Navier-Stokes: deducción, soluciones exactas y análisis dimensional
- Teoría de la capa límite: ecuaciones de Prandtl y solución de Blasius
- Turbulencia: promediado de Reynolds, modelo k-ε y simulación numérica directa

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Derivar las ecuaciones de Navier-Stokes a partir de principios de conservación e identificar el significado físico de cada término [Familiarizarse]
2. Analizar flujos no viscosos usando las ecuaciones de Euler y aplicar el principio de Bernoulli [Usar]
3. Aplicar la teoría de la capa límite para estimar el arrastre sobre una placa plana y comparar con la solución de Blasius [Evaluar]

4.7.7. MMS/Teoría de Control Matemático (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos de espacio de estados, controlabilidad y observabilidad, diseño de realimentación, estabilidad (Lyapunov) y control óptimo.
Temas:
Core

- Representación en espacio de estados de sistemas lineales: funciones de transferencia y realizaciones
- Controlabilidad y observabilidad: condiciones de rango y descomposición de Kalman
- Estabilización por realimentación: asignación de polos, diseño LQR y principio de separación
- Estabilidad de Lyapunov para sistemas no lineales y estabilidad entrada-salida
- Control óptimo: Principio de Máximo de Pontryagin y ecuación de Hamilton-Jacobi-Bellman

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Determinar la controlabilidad y observabilidad de un sistema lineal a partir de sus matrices de espacio de estados [Familiarizarse]
2. Diseñar un controlador por realimentación de estados mediante asignación de polos u optimización LQR [Usar]
3. Aplicar el Principio de Máximo de Pontryagin para resolver un problema básico de control óptimo [Evaluar]

4.7.8. MMS/Ecuaciones Diferenciales Estocásticas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cálculo de Itô, EDEs impulsadas por movimiento browniano, ecuación de Fokker-Planck y simulación numérica de EDEs.
Temas:
Core

- Integral de Itô, variación cuadrática y fórmula de Itô
- Existencia y unicidad de soluciones de EDEs bajo condiciones de Lipschitz
- Ecuación de Fokker-Planck: deducción y conexión con EDEs
- Fórmula de Feynman-Kac y conexiones con EDPs parabólicas
- Esquemas de Euler-Maruyama y Milstein para simulación numérica de EDEs

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar la fórmula de Itô y aplicarla para calcular la diferencial de una función de un proceso estocástico [Familiarizarse]
2. Derivar la ecuación de Fokker-Planck a partir de una EDE dada e interpretarla como un flujo de probabilidad [Usar]
3. Implementar el esquema de Euler-Maruyama para simular trayectorias de una ecuación diferencial estocástica [Evaluar]

4.7.9. MMS/Modelado Basado en Agentes y Sistemas Complejos

Modelado basado en agentes, emergencia, modelos de redes, sistemas adaptativos complejos y metodología de simulación.

Temas:

Non Core

- Modelos basados en agentes (ABMs): principios de diseño, reglas y emergencia de comportamiento a nivel macro
- Modelos de redes complejas: Erdős-Rényi, Barabási-Albert y redes de mundo pequeño
- Autómatas celulares: Juego de la Vida de Conway, métodos de lattice Boltzmann y universalidad
- Dinámica de sistemas: modelos de stock-flujo y bucles de realimentación (Forrester)
- Metodología de simulación: calibración, análisis de sensibilidad y validación de modelos

Aprendizaje esperado (*Learning Outcomes*):

NonCore:

1. Describir los principios del modelado basado en agentes y explicar cómo los patrones a nivel macro emergen de reglas locales [Familiarizarse]
2. Construir un modelo basado en agentes de un sistema social o biológico y analizar su comportamiento emergente [Usar]
3. Validar un modelo de simulación mediante análisis de sensibilidad y comparación con datos empíricos [Evaluar]

4.8. Fundamentos Matemáticos de la Ciencia de Datos (FDS)

Esta área cubre la teoría matemática subyacente a la ciencia de datos moderna y el aprendizaje automático, incluyendo aprendizaje estadístico, optimización, métodos de kernel, reducción de dimensionalidad, análisis topológico de datos, teoría de la información y las matemáticas del aprendizaje profundo.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.8.1 Teoría del Aprendizaje Estadístico (Pág. 211)	1	1
⭐ 4.8.2 Optimización para Aprendizaje Automático (Pág. 212)	1	1
⭐ 4.8.3 Análisis Topológico de Datos (Pág. 212)	1	1
⭐ 4.8.4 Reducción de Dimensionalidad (Pág. 212)	1	1
⭐ 4.8.5 Métodos de Kernel y Espacios de Hilbert con Núcleo Reproductor (Pág. 213)	1	1
⭐ 4.8.6 Teoría de la Información (Pág. 213)	1	1
⭐ 4.8.7 Fundamentos Matemáticos del Aprendizaje Profundo (Pág. 214)	1	1

Cuadro 4.8: Lista de KUs del área de Fundamentos Matemáticos de la Ciencia de Datos.

4.8.1. FDS/Teoría del Aprendizaje Estadístico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Marco PAC-learning, dimensión VC, complejidad de Rademacher, cotas de generalización y equilibrio sesgo-varianza.

Temas:

Core

- Marco PAC-learning: complejidad muestral, clases de hipótesis y capacidad de aprendizaje
- Dimensión VC: definición, ejemplos y el teorema fundamental del aprendizaje
- Complejidad de Rademacher y cotas de convergencia uniforme
- Descomposición sesgo-varianza y el fenómeno de doble descenso
- Regularización: regularización de Tikhonov y minimización del riesgo estructural

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar la definición de PAC-learning y determinar si una clase de hipótesis es PAC-aprendible [Familiarizarse]

2. Calcular la dimensión VC de clases de hipótesis estándar y derivar cotas de complejidad muestral [Usar]
3. Analizar el error de generalización usando complejidad de Rademacher y explicar el fenómeno de doble descenso [Evaluar]

4.8.2. FDS/Optimización para Aprendizaje Automático (Core Tier1: 1 hr, Core Tier2: 1 hr)

Teoría de optimización convexa, SGD y sus variantes, métodos adaptativos y paisajes de optimización no convexa.

Temas:

Core

- Convexidad, convexidad fuerte y dualidad de Fenchel en optimización
- Descenso de gradiente estocástico: tasas de convergencia para objetivos convexos y no convexos
- Métodos de gradiente adaptativo: AdaGrad, RMSProp y Adam
- Optimización no convexa: puntos de silla, mínimos locales y geometría del paisaje de pérdida de redes neuronales
- Métodos de gradiente proximal y descenso por coordenadas para problemas estructurados

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar las garantías de convergencia de SGD para funciones convexas suaves e identificar el papel de la tasa de aprendizaje [Familiarizarse]
2. Aplicar Adam y métodos de gradiente proximal para entrenar modelos de aprendizaje automático regularizados [Usar]
3. Analizar el paisaje de pérdida de una red sobreparametrizada y explicar por qué SGD encuentra buenos mínimos [Evaluar]

4.8.3. FDS/Análisis Topológico de Datos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Homología persistente, complejos de Vietoris-Rips, algoritmo mapper y aplicaciones al análisis de formas de datos.

Temas:

Core

- Complejos simpliciales: complejos de Čech, Vietoris-Rips y witness
- Homología persistente: filtraciones, diagramas de persistencia y códigos de barras
- Teorema de estabilidad para diagramas de persistencia y la distancia de cuello de botella
- Algoritmo mapper para resumir topológicamente datos de alta dimensión

Non Core

- Aplicaciones de TDA a ciencia de materiales, genómica y análisis de redes neuronales (2020-2025)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Construir una filtración de Vietoris-Rips a partir de una nube de puntos y calcular su diagrama de persistencia [Familiarizarse]
2. Aplicar el teorema de estabilidad para acotar la distancia entre diagramas de persistencia de conjuntos de datos similares [Usar]

NonCore:

3. Interpretar características topológicas de un conjunto de datos usando el algoritmo mapper y explicar su significado científico [Evaluar]

4.8.4. FDS/Reducción de Dimensionalidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Reducción de dimensionalidad lineal y no lineal: PCA, proyecciones aleatorias, aprendizaje de variedades y métodos modernos (t-SNE, UMAP).

Temas:

Core

- PCA: descomposición en valores singulares, varianza explicada y aproximación óptima de bajo rango
- Lema de Johnson-Lindenstrauss y proyecciones aleatorias para reducción de dimensionalidad
- Aprendizaje de variedades: Isomap, incrustación lineal local (LLE) e incrustación espectral
- Métodos no lineales modernos: t-SNE y UMAP; teoría, parámetros y limitaciones
- Autoencoders y autoencoders variacionales (VAEs) como reducción de dimensionalidad no lineal

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar el lema de Johnson-Lindenstrauss y justificar por qué las proyecciones aleatorias preservan distancias [Familiarizarse]
2. Aplicar PCA y t-SNE a datos de alta dimensión e interpretar las incrustaciones resultantes en baja dimensión [Usar]
3. Comparar métodos de reducción de dimensionalidad lineales y no lineales con respecto a la preservación de la geometría y el costo computacional [Evaluar]

4.8.5. FDS/Métodos de Kernel y Espacios de Hilbert con Núcleo Reproductor (Core Tier1: 1 hr, Core Tier2: 1 hr)

Kernels definidos positivos, teoría RKHS, máquinas de vectores de soporte, kernel PCA y el truco del kernel.
Temas:
Core

- Kernels definidos positivos: teorema de Mercer y funciones kernel comunes (RBF, polinomial, Matérn)
- Espacios de Hilbert con núcleo reproductor: teorema del representante y regularización
- Máquinas de vectores de soporte: clasificación de margen máximo y formulación dual
- PCA con kernel y regresión con kernel mediante el truco del kernel
- Procesos Gaussianos como distribuciones sobre funciones: inferencia a posteriori y aprendizaje de hiperparámetros

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar el teorema de Mercer y verificar que una función dada es un kernel definido positivo [Familiarizarse]
2. Derivar la formulación dual de SVM y aplicar el truco del kernel para "kernelizar" un algoritmo lineal [Usar]
3. Aplicar regresión por procesos Gaussianos a un problema de predicción y calibrar las estimaciones de incertidumbre [Evaluar]

4.8.6. FDS/Teoría de la Información (Core Tier1: 1 hr, Core Tier2: 1 hr)

Entropía de Shannon, información mutua, capacidad de canal, teoría de tasa-distorsión y aplicaciones al aprendizaje automático.
Temas:
Core

- Entropía de Shannon, entropía conjunta y condicional, y la regla de la cadena
- Información mutua, divergencia KL y la desigualdad de procesamiento de datos
- Capacidad de canal: teorema de codificación de canal ruidoso de Shannon y el canal binario simétrico
- Teoría de tasa-distorsión y el equilibrio entre compresión y fidelidad
- Fundamentos de teoría de la información en ML: MDL, inferencia variacional (ELBO) y el cuello de botella de la información

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular entropía de Shannon e información mutua para distribuciones discretas e interpretar su significado [Familiarizarse]

2. Aplicar el teorema de capacidad de canal de Shannon para determinar la tasa máxima de transmisión confiable [Usar]
3. Analizar un algoritmo de aprendizaje usando el principio del cuello de botella de la información y relacionar compresión con generalización [Evaluar]

4.8.7. FDS/Fundamentos Matemáticos del Aprendizaje Profundo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aproximación universal, kernel tangente neural, sesgo implícito de SGD, sobreparametrización y aprendizaje profundo geométrico.

Temas:

Core

- Teoremas de aproximación universal: compromisos entre anchura y profundidad para redes ReLU
- Kernel tangente neural (NTK) y el límite de anchura infinita de redes neuronales
- Regularización implícita y sesgo implícito de SGD hacia soluciones de norma mínima
- Sobreajuste benigno y doble descenso en modelos sobreparametrizados
- Aprendizaje profundo geométrico: equivarianza, simetría de grupo y redes neuronales de grafos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar el teorema de aproximación universal e identificar sus supuestos y limitaciones [Familiarizarse]
2. Explicar el kernel tangente neural y su papel en la comprensión de la dinámica de entrenamiento de redes anchas [Usar]
3. Analizar el sesgo implícito del descenso de gradiente y relacionarlo con la generalización en regímenes sobreparametrizados [Evaluar]

4.9. Geometría y Topología Computacional (CGT)

Esta área cubre algoritmos geométricos y métodos topológicos para procesar datos espaciales discretos, incluyendo estructuras de Voronoi, envolventes convexas, generación de mallas, topología digital y homología persistente aplicada, con aplicaciones a gráficos por computadora, robótica y computación científica.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.9.1 Diagramas de Voronoi y Triangulaciones de Delaunay (Pág. 214)	1	1
⭐ 4.9.2 Envolventes Convexas y Politopos (Pág. 215)	1	1
⭐ 4.9.3 Topología Digital y Análisis de Imágenes (Pág. 215)	1	1
⭐ 4.9.4 Homología Persistente y Topología Aplicada (Pág. 216)	1	1
⭐ 4.9.5 Generación de Mallas y Algoritmos Geométricos (Pág. 216)	1	1

Cuadro 4.9: Lista de KUs del área de Geometría y Topología Computacional.

4.9.1. CGT/Diagramas de Voronoi y Triangulaciones de Delaunay (Core Tier1: 1 hr, Core Tier2: 1 hr)

Diagramas de Voronoi, triangulaciones de Delaunay, dualidad, algoritmos y aplicaciones en computación espacial.

Temas:

Core

- Diagrama de Voronoi: definición, complejidad combinatoria y propiedades en $\mathbb{R}^d$
- Triangulación de Delaunay: propiedad de la circunferencia vacía y dualidad con Voronoi
- Algoritmos: algoritmo de la línea de barrido de Fortune y construcción incremental aleatorizada
- Diagramas de Voronoi ponderados y diagramas de potencia

- Aplicaciones: búsqueda del vecino más cercano, planificación de trayectorias y generación de mallas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la dualidad entre el diagrama de Voronoi y la triangulación de Delaunay y demostrar la propiedad de la circunferencia vacía [Familiarizarse]
2. Construir diagramas de Voronoi y triangulaciones de Delaunay usando el algoritmo de Fortune y analizar su complejidad [Usar]
3. Aplicar triangulaciones de Delaunay a la generación de mallas y consultas de vecino más cercano en conjuntos de datos espaciales [Evaluar]

4.9.2. CGT/Envolventes Convexas y Politopos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Algoritmos de envolvente convexa en 2D y 3D, politopos convexos, consultas de puntos extremos e intersección de semiespacios.

Temas:
Core

- Algoritmos de envolvente convexa 2D: escaneo de Graham, marcha de Jarvis y algoritmo de Chan
- Envolvente convexa 3D: construcción incremental y algoritmos aleatorizados
- Politopos convexos: retículo de caras, relación de Euler y teorema de la cota superior
- Intersección de semiespacios y dualidad entre envolvente convexa y problemas de semiespacios
- Aplicaciones: programación lineal, detección de colisiones y análisis de formas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comparar la complejidad temporal del escaneo de Graham, la marcha de Jarvis y el algoritmo de Chan e identificar casos óptimos [Familiarizarse]
2. Implementar un algoritmo de envolvente convexa 2D y extenderlo a la construcción incremental 3D [Usar]
3. Aplicar la dualidad entre envolvente convexa e intersección de semiespacios para resolver problemas de programación lineal geométricamente [Evaluar]

4.9.3. CGT/Topología Digital y Análisis de Imágenes (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades topológicas de imágenes digitales, componentes conexas, esqueletización, skeletonización y persistencia topológica en datos de imagen.

Temas:
Core

- Adyacencia digital, conectividad y el teorema de la curva de Jordan para imágenes digitales
- Etiquetado de componentes conexas e invariantes topológicos de imágenes binarias
- Algoritmos de thinning y skeletonización del eje medial
- Característica de Euler de objetos digitales y su cálculo a partir de patrones locales

Non Core

- Métodos topológicos para segmentación de imágenes y reconstrucción de superficies a partir de nubes de puntos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la adyacencia digital y por qué una conectividad inconsistente conduce a paradojas topológicas [Familiarizarse]
2. Aplicar algoritmos de thinning y skeletonización para extraer características estructurales de imágenes binarias [Usar]

NonCore:

3. Calcular invariantes topológicos (característica de Euler, números de Betti) de un objeto digital y usarlos para clasificación de formas [Evaluar]

4.9.4. CGT/Homología Persistente y Topología Aplicada (Core Tier1: 1 hr, Core Tier2: 1 hr)

Homología simplicial, filtraciones, módulos de persistencia, códigos de barras y aplicaciones al análisis de formas y datos.

Temas:

Core

- Homología simplicial: grupos de cadenas, operadores borde y números de Betti
- Módulos de persistencia y el teorema de estructura para descomposición en módulos de intervalo
- Códigos de barras y diagramas de persistencia: pares nacimiento-muerte y vida topológica
- Algoritmos para homología persistente: algoritmo de reducción de matriz de incidencia

Non Core

- Aplicaciones: plegamiento de proteínas, ciencia de materiales y neurociencia (2020-2025)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular grupos de homología simplicial y números de Betti a partir de un complejo simplicial [Familiarizarse]
2. Construir una filtración a partir de una nube de puntos y calcular su diagrama de persistencia mediante reducción de matriz de incidencia [Usar]

NonCore:

3. Interpretar códigos de barras de datos reales y explicar las características topológicas que revelan en un contexto científico [Evaluar]

4.9.5. CGT/Generación de Mallas y Algoritmos Geométricos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Generación de mallas triangulares y tetraédricas, refinamiento de Delaunay, métricas de calidad y algoritmos para subdivisión espacial.

Temas:

Core

- Métricas de calidad de malla: relación de aspecto, ángulo mínimo y relación circunradio-arista más corta
- Refinamiento de Delaunay: algoritmo de Ruppert y algoritmo de Shewchuk para generación de mallas de calidad
- Generación de mallas tetraédricas: tetraedralización de Delaunay restringida y puntos de Steiner
- Mallado dominante por cuadriláteros y hexaedros: métodos de campo cruzado y mallas poliédricas

Non Core

- Refinamiento adaptativo de malla guiado por estimadores de error en simulaciones de elementos finitos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar la estrategia de refinamiento de Delaunay y el papel de los puntos de Steiner para lograr garantías de calidad [Familiarizarse]
2. Generar una triangulación de Delaunay restringida de calidad para un dominio planar dado usando el algoritmo de Ruppert [Usar]

NonCore:

3. Diseñar una estrategia de refinamiento adaptativo de malla para un problema de EDP y justificar los criterios de refinamiento basados en estimaciones de error a posteriori [Evaluar]

4.10. Información y Computación Cuántica (QIC)

Esta área cubre la teoría matemática de la información y computación cuántica, incluyendo el formalismo de álgebra lineal de la mecánica cuántica, algoritmos cuánticos, teoría del entrelazamiento,

corrección de errores y métodos de redes de tensores para simulación cuántica.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.10.1 Fundamentos de Mecánica Cuántica para Computación (Pág. 217)	1	1
⭐ 4.10.2 Circuitos y Puertas Cuánticas (Pág. 217)	1	1
⭐ 4.10.3 Algoritmos Cuánticos (Pág. 218)	1	1
⭐ 4.10.4 Entrelazamiento y Desigualdades de Bell (Pág. 218)	1	1
⭐ 4.10.5 Corrección de Errores Cuánticos (Pág. 218)	1	1
4.10.6 Redes de Tensores y Simulación Cuántica (Pág. 219)	Electivo	

Cuadro 4.10: Lista de KUs del área de Información y Computación Cuántica.

4.10.1. QIC/Fundamentos de Mecánica Cuántica para Computación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Postulados de la mecánica cuántica formulados para computación: espacios de estados, observables, medición y evolución unitaria.

Temas:

Core

- Formalismo de espacio de Hilbert: vectores de estado, notación de Dirac y productos internos
- Observables como operadores hermíticos: valores propios, estados propios y teorema espectral
- Postulado de medición: regla de Born, mediciones proyectivas y colapso del estado
- Evolución unitaria: ecuación de Schrödinger y operador de evolución temporal
- Matrices densidad: estados mezcla, traza parcial y representación de la esfera de Bloch

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar los postulados de la mecánica cuántica y representar estados de un solo qubit en la esfera de Bloch [Familiarizarse]
2. Calcular probabilidades de medición y estados post-medición usando la regla de Born y operadores de proyección [Usar]
3. Distinguir estados puros de estados mezcla usando matrices densidad y la traza parcial sobre un subsistema [Evaluar]

4.10.2. QIC/Circuitos y Puertas Cuánticas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Qubits, puertas cuánticas, circuitos cuánticos, universalidad y el modelo de circuitos de la computación cuántica.

Temas:

Core

- Puertas de un solo qubit: Pauli, Hadamard, fase y rotación
- Puertas de múltiples qubits: CNOT, Toffoli y puertas controladas-U
- Circuitos cuánticos: diagramas de circuito, profundidad y reversibilidad
- Universalidad: teorema de Solovay-Kitaev y conjuntos de puertas universales
- Paralelismo cuántico, interferencia y el papel de la superposición en la computación

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Construir circuitos cuánticos usando conjuntos de puertas estándar y representar su acción sobre estados base [Familiarizarse]
2. Analizar la acción de un circuito cuántico sobre un estado de entrada mediante multiplicación de matrices y notación de Dirac [Usar]
3. Justificar la universalidad de un conjunto de puertas usando el teorema de Solovay-Kitaev y aproximar un unitario objetivo [Evaluar]

4.10.3. QIC/Algoritmos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Transformada de Fourier cuántica, estimación de fase, algoritmo de factorización de Shor, búsqueda de Grover y HHL para sistemas lineales.

Temas:

Core

- Transformada de Fourier cuántica: construcción de circuitos y análisis de complejidad
- Algoritmo de estimación de fase cuántica y su papel como subrutina
- Algoritmo de factorización de Shor: determinación del período y reducción a factorización
- Algoritmo de búsqueda de Grover: aceleración cuadrática y amplificación de amplitud

Non Core

- Algoritmo HHL para sistemas lineales y aceleración cuántica para aprendizaje automático

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir la transformada de Fourier cuántica y explicar cómo la estimación de fase extrae información de valores propios [Familiarizarse]
2. Recorrer paso a paso los algoritmos de Shor y Grover e identificar la fuente de la aceleración cuántica [Usar]

NonCore:

3. Analizar los requisitos de recursos del algoritmo HHL e identificar las condiciones bajo las cuales logra una aceleración exponencial [Evaluar]

4.10.4. QIC/Entrelazamiento y Desigualdades de Bell (Core Tier1: 1 hr, Core Tier2: 1 hr)

Entrelazamiento cuántico: estados de Bell, desigualdades de Bell, desigualdad CHSH, no-clonación y medidas de entrelazamiento.

Temas:

Core

- Estados de Bell, estados entrelazados vs. separables y descomposición de Schmidt
- Desigualdades de Bell y desigualdad CHSH: teorías de variables ocultas locales vs. mecánica cuántica
- Teorema de no-clonación y teorema de no-comunicación
- Medidas de entrelazamiento: entropía de entrelazamiento, concurrencia y negatividad
- Protocolos de teleportación cuántica y codificación superdensa

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Verificar que los estados de Bell están máximamente entrelazados y calcular su descomposición de Schmidt [Familiarizarse]
2. Derivar la desigualdad CHSH y mostrar que la mecánica cuántica la viola para el estado singlete [Usar]
3. Cuantificar el entrelazamiento de un estado bipartito usando entropía de entrelazamiento y concurrencia [Evaluar]

4.10.5. QIC/Corrección de Errores Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Decoherencia, modelos de error cuántico, códigos estabilizadores, código superficial y computación cuántica tolerante a fallos.

Temas:

Core

- Decoherencia y ruido cuántico: canales de bit-flip, phase-flip y depolarizante
- Códigos correctores de errores cuánticos: código de Shor y código de Steane de 7 qubits
- Formalismo de estabilizadores: grupo de Pauli, códigos estabilizadores y medición de síndrome
- Código superficial: teorema del umbral, qubits lógicos y realizaciones experimentales recientes (2023-2025)
- Computación cuántica tolerante a fallos: puertas transversales y teorema del umbral

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Explicar la construcción del código de Shor y describir cómo corrige errores arbitrarios de un solo qubit [Familiarizarse]
2. Aplicar el formalismo de estabilizadores para codificar un qubit lógico y determinar el síndrome para un error dado [Usar]
3. Analizar el umbral del código superficial y explicar por qué es el principal candidato para el hardware cuántico tolerante a fallos [Evaluar]

4.10.6. QIC/Redes de Tensores y Simulación Cuántica

Representaciones de redes de tensores de estados cuánticos (MPS, PEPS, MERA), algoritmos de contracción y aplicaciones a la simulación de muchos cuerpos cuánticos.
Temas:
Non Core

- Contracciones de tensores: notación de índices, orden de contracción y complejidad computacional
- Estados de producto matricial (MPS) y el grupo de renormalización de matriz densidad (DMRG)
- Estados de pares entrelazados proyectados (PEPS) para sistemas bidimensionales y de dimensiones superiores
- Ansatz de renormalización de entrelazamiento multiescala (MERA) y conexiones holográficas
- Redes de tensores en aprendizaje automático: descomposiciones de tensores y algoritmos inspirados en la cuántica

Aprendizaje esperado (*Learning Outcomes*):
NonCore:

1. Describir la representación MPS de un estado cuántico de muchos cuerpos y explicar cómo la dimensión de enlace controla la calidad de la aproximación [Familiarizarse]
2. Aplicar DMRG para calcular el estado fundamental de una cadena de espines cuántica 1D y analizar la convergencia con la dimensión de enlace [Usar]
3. Evaluar el costo computacional de la contracción de redes de tensores y comparar la simulación clásica con redes de tensores frente al hardware cuántico [Evaluar]

4.11. Biología Matemática y Ciencias de la Vida (MBL)

Esta área cubre la aplicación de métodos matemáticos a sistemas biológicos, abarcando dinámica de poblaciones, epidemiología, biología de sistemas, bioinformática y neurociencia computacional, con énfasis en modelos basados en datos experimentales modernos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.11.1 Dinámica de Poblaciones (Pág. 219)	1	1
⭐ 4.11.2 Epidemiología Matemática (Pág. 220)	1	1
⭐ 4.11.3 Biología de Sistemas y Dinámica de Redes (Pág. 220)	1	1
⭐ 4.11.4 Algoritmos de Bioinformática (Pág. 221)	1	1
⭐ 4.11.5 Neurociencia Matemática (Pág. 221)	1	1

Cuadro 4.11: Lista de KUs del área de Biología Matemática y Ciencias de la Vida.

4.11.1. MBL/Dinámica de Poblaciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos matemáticos de poblaciones simples e interactuantes: crecimiento logístico, sistemas de Lotka-Volterra, ecología espacial y dinámica evolutiva.
Temas:
Core

- Modelos de crecimiento exponencial y logístico; capacidad de carga y efecto Allee

- Modelos depredador-presa y de competición de Lotka-Volterra; equilibrios y análisis de estabilidad
- Modelos estructurados por edad: matrices de Leslie y ecuación de Euler-Lotka
- Modelos de reacción-difusión y ondas viajeras en ecología espacial (ecuación de Fisher-KPP)
- Dinámica evolutiva: ecuaciones de replicador, teoría de juegos y ecuación de Price

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar la estabilidad de los equilibrios en sistemas de Lotka-Volterra mediante linealización y retratos de fase [Familiarizarse]
2. Construir modelos de población estructurados por edad usando matrices de Leslie y calcular tasas de crecimiento a largo plazo [Usar]
3. Aplicar la teoría de reacción-difusión para modelar la propagación espacial de una especie invasora usando la ecuación de Fisher-KPP [Evaluar]

4.11.2. MBL/Epidemiología Matemática (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos compartimentales (SIR, SEIR), el número básico de reproducción, inmunidad colectiva, epidemiología en redes y predicción en tiempo real de epidemias.
Temas:
Core

- Modelos compartimentales SIR y SIS: deducción, equilibrios y el umbral epidémico
- Número básico de reproducción $\mathcal{R}_0$: definición, cálculo y su papel en la inmunidad colectiva
- Extensiones SEIR y SEIRS: latencia, inmunidad decreciente y epidemias estructuradas por edad
- Epidemiología en redes: estructuras de contacto heterogéneas y eventos de superpropagación
- Predicción epidemiológica en tiempo real: asimilación de datos, cuantificación de incertidumbre y lecciones del COVID-19 (2020-2025)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Derivar la condición de umbral epidémico a partir del modelo SIR e interpretar $\mathcal{R}_0$ en términos de inmunidad colectiva [Familiarizarse]
2. Extender el modelo SIR básico para incorporar vacunación, inmunidad decreciente o estructura por edad [Usar]
3. Calibrar un modelo SEIR con datos epidémicos y cuantificar la incertidumbre de la predicción usando métodos de conjunto [Evaluar]

4.11.3. MBL/Biología de Sistemas y Dinámica de Redes (Core Tier1: 1 hr, Core Tier2: 1 hr)

Redes de regulación génica, osciladores bioquímicos, transducción de señales, topología de redes biológicas y modelado multi-escala.
Temas:
Core

- Redes de regulación génica: modelos booleanos, funciones de Hill y motivos de realimentación
- Osciladores bioquímicos: el represilador, oscilador de Goodwin y bifurcación de Hopf en biología
- Transducción de señales: ultrasensibilidad, biestabilidad y la cascada MAPK
- Topología de redes biológicas: redes libres de escala, motivos de red y robustez

Non Core

- Modelado multi-escala: de la escala molecular a la tisular; modelos metabólicos basados en restricciones (FBA)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Identificar motivos de realimentación en redes de regulación génica y predecir su papel en la adaptación y la biestabilidad [Familiarizarse]

2. Analizar un modelo de oscilador bioquímico usando teoría de bifurcaciones y determinar condiciones para oscilaciones de ciclo límite [Usar]

NonCore:

3. Construir un modelo de análisis de balance de flujo de una red metabólica y predecir tasas de crecimiento bajo restricciones de nutrientes [Evaluar]

4.11.4. MBL/Algoritmos de Bioinformática (Core Tier1: 1 hr, Core Tier2: 1 hr)

Alineamiento de secuencias, modelos ocultos de Markov, filogenética, ensamblaje de genomas y aprendizaje automático en genómica.

Temas:

Core

- Alineamiento de secuencias: programación dinámica (Needleman-Wunsch, Smith-Waterman) y heurísticas BLAST
- Modelos ocultos de Markov para análisis de secuencias: algoritmos de Viterbi y Baum-Welch
- Inferencia filogenética: métodos de distancia (UPGMA, neighbor-joining) y máxima verosimilitud
- Ensamblaje de genomas: enfoque de grafo de Bruijn, solapamiento-consenso y ensamblaje con lecturas largas

Non Core

- Aprendizaje automático en genómica: aprendizaje profundo para expresión génica, detección de variantes y estructura de proteínas (AlphaFold, 2021-2025)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar programación dinámica para realizar alineamiento de secuencias global y local e interpretar las puntuaciones de alineamiento [Familiarizarse]
2. Implementar un HMM para la búsqueda de genes y aplicar el algoritmo de Viterbi para decodificar la secuencia de estados ocultos más probable [Usar]

NonCore:

3. Evaluar enfoques de aprendizaje profundo para la predicción de estructura de proteínas y explicar las innovaciones clave de AlphaFold [Evaluar]

4.11.5. MBL/Neurociencia Matemática (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos neuronales biofísicos, codificación neural, dinámica de redes, reducción de dimensionalidad en datos neuronales y psiquiatría computacional.

Temas:

Core

- Modelo de Hodgkin-Huxley: dinámica de canales iónicos, generación de potencial de acción y análisis de bifurcaciones
- Modelos de integración y disparo: IF con fuga, IF exponencial y adaptación de frecuencia de disparo
- Codificación neural: codificación por tasa, temporización de disparos, códigos poblacionales y el cuello de botella de la información neural
- Dinámica de poblaciones neuronales: atractores anulares, variedades de baja dimensión y modelos de variables latentes (2020-2025)

Non Core

- Psiquiatría computacional: hipótesis del cerebro bayesiano, codificación predictiva y aprendizaje por refuerzo en la toma de decisiones

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Derivar el potencial de acción a partir de las ecuaciones de Hodgkin-Huxley e identificar el papel de cada corriente iónica [Familiarizarse]
2. Aplicar reducción de dimensionalidad a registros de poblaciones neuronales e identificar variedades de actividad de baja dimensión [Usar]

NonCore:

3. Formular un modelo de codificación predictiva de la percepción y relacionarlo con la inferencia bayesiana jerárquica [Evaluar]

4.12. Matemáticas Financieras y Actuariales (FAM)

Esta área cubre los fundamentos matemáticos de las finanzas y la ciencia actuarial, desde la teoría del interés y el arbitraje hasta modelos estocásticos para derivados, medición de riesgos, tablas de mortalidad actuariales y métodos computacionales en finanzas cuantitativas.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 4.12.1 Teoría del Interés y Valor del Dinero en el Tiempo (Pág. 222)	1	1
⭐ 4.12.2 Valoración Libre de Arbitraje y Modelos de Mercado (Pág. 222)	1	1
⭐ 4.12.3 Modelo de Black-Scholes y Valoración de Opciones (Pág. 223)	1	1
⭐ 4.12.4 Medidas de Riesgo y Teoría de Portafolios (Pág. 223)	1	1
⭐ 4.12.5 Modelos Actuariales y Tablas de Mortalidad (Pág. 224)	1	1
⭐ 4.12.6 Finanzas Computacionales y Métodos de Monte Carlo (Pág. 224)	1	1

Cuadro 4.12: Lista de KUs del área de Matemáticas Financieras y Actuariales.

4.12.1. FAM/Teoría del Interés y Valor del Dinero en el Tiempo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Valor del dinero en el tiempo, interés compuesto, anualidades, curvas de rendimiento, valoración de bonos y estrategias de inmunización.

Temas:

Core

- Interés compuesto: funciones de acumulación, tasas efectivas y nominales, y fuerza de interés
- Anualidades vencidas y anticipadas: valores actuales y acumulados, perpetuidades
- Valoración de bonos: relación precio-rendimiento, duración, convexidad y planes de amortización
- Curvas de rendimiento: tasas spot, tasas forward y métodos de bootstrap para la estructura de plazos
- Inmunización de carteras: condiciones de Redington y emparejamiento de flujos de efectivo

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el valor actual y el valor acumulado de anualidades bajo varios supuestos de tasa de interés [Familiarizarse]
2. Valorar bonos usando la relación precio-rendimiento y calcular la duración y convexidad para la sensibilidad a la tasa de interés [Usar]
3. Construir una cartera que satisfaga las condiciones de inmunización de Redington frente a desplazamientos paralelos de la curva de rendimiento [Evaluar]

4.12.2. FAM/Valoración Libre de Arbitraje y Modelos de Mercado (Core Tier1: 1 hr, Core Tier2: 1 hr)

Teoremas fundamentales de valoración de activos, medidas neutrales al riesgo, mercados completos y modelos de mercado discretos y continuos.

Temas:

Core

- Arbitraje: definición, ley del precio único y argumentos de replicación estática
- Medidas de probabilidad neutrales al riesgo y el primer teorema fundamental de la valoración de activos
- Mercados completos y el segundo teorema fundamental de la valoración de activos
- Modelo de árbol binomial: replicación, valoración neutral al riesgo y convergencia a Black-Scholes

- Teorema de representación de martingalas y cobertura en mercados continuos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enunciar el primer y segundo teoremas fundamentales de la valoración de activos y explicar su interpretación financiera [Familiarizarse]
2. Valorar derivados europeos y americanos usando el modelo de árbol binomial bajo valoración neutral al riesgo [Usar]
3. Verificar la completitud del mercado y construir una cartera replicante para un derivado usando el teorema de representación de martingalas [Evaluar]

4.12.3. FAM/Modelo de Black-Scholes y Valoración de Opciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelo de Black-Scholes: deducción, EDP, griegas, volatilidad implícita y extensiones a modelos de volatilidad estocástica y con saltos.

Temas:
Core

- EDP de Black-Scholes: deducción mediante cobertura delta y valoración neutral al riesgo
- Fórmula cerrada de Black-Scholes para opciones de compra y venta europeas; paridad put-call
- Griegas de opciones: delta, gamma, vega, theta y rho; estrategias de cobertura dinámica
- Superficie de volatilidad implícita, sonrisa de volatilidad y modelo de volatilidad local de Dupire

Non Core

- Modelos de volatilidad estocástica: modelo de Heston y swaps de varianza (métodos de calibración 2020-2025)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Derivar la EDP de Black-Scholes a partir del argumento de cobertura delta e identificar sus condiciones de contorno [Familiarizarse]
2. Calcular precios de opciones y las griegas usando la fórmula de Black-Scholes e interpretar su significado para la cobertura [Usar]

NonCore:

3. Calibrar un modelo de volatilidad estocástica a precios de opciones de mercado y reproducir la sonrisa de volatilidad implícita [Evaluar]

4.12.4. FAM/Medidas de Riesgo y Teoría de Portafolios (Core Tier1: 1 hr, Core Tier2: 1 hr)

Optimización media-varianza, medidas de riesgo coherentes, Valor en Riesgo, CVaR, cópulas y teoría del valor extremo.

Temas:
Core

- Teoría de portafolios media-varianza: frontera eficiente, modelo CAPM y ratio de Sharpe
- Valor en Riesgo (VaR): definición, simulación histórica y estimación paramétrica
- Medidas de riesgo coherentes: axiomas, Valor en Riesgo Condicional (CVaR) y Déficit Esperado
- Cópulas: teorema de Sklar, cópulas Gaussianas y t-, y dependencia de colas
- Teoría del valor extremo: distribución Generalizada del Valor Extremo y método de Peaks-over-Threshold

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Construir la frontera eficiente de media-varianza e identificar el portafolio tangente bajo el CAPM [Familiarizarse]
2. Calcular VaR y CVaR para un portafolio usando simulación histórica y métodos paramétricos [Usar]

3. Aplicar modelos de cópulas para capturar la dependencia de colas en un análisis de riesgo multivariante [Evaluar]

4.12.5. FAM/Modelos Actuariales y Tablas de Mortalidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos de supervivencia, tablas de mortalidad, primas netas, reservas de pólizas, modelos de pérdida y teoría de la credibilidad.

Temas:

Core

- Funciones de supervivencia, tasas de riesgo y fuerza de mortalidad; leyes de Gompertz y Makeham
- Tablas de mortalidad: construcción, tablas selectas y ultimativas, y métodos de interpolación
- Valoración de seguros de vida y anualidades: primas netas, primas brutas y principio de equivalencia
- Reservas de pólizas: métodos prospectivo y retrospectivo, ecuación diferencial de Thiele
- Modelos de pérdida: marco frecuencia-severidad, distribuciones agregadas y teoría de la credibilidad

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular cantidades de tablas de mortalidad (tasas de mortalidad, esperanza de vida) a partir de datos brutos de supervivencia [Familiarizarse]
2. Valorar productos de seguros de vida y anualidades usando valoración por prima neta bajo tablas de mortalidad estándar [Usar]
3. Calcular reservas de pólizas de forma prospectiva y retrospectiva y verificar su equivalencia bajo interés constante [Evaluar]

4.12.6. FAM/Finanzas Computacionales y Métodos de Monte Carlo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Simulación de Monte Carlo para valoración de derivados, reducción de varianza, métodos de diferencias finitas para la EDP de Black-Scholes y aprendizaje automático en finanzas cuantitativas.

Temas:

Core

- Simulación de Monte Carlo para valoración de derivados europeos y dependientes de la trayectoria
- Técnicas de reducción de varianza: variables antitéticas, variables de control y muestreo por importancia
- Métodos de diferencias finitas para la EDP de Black-Scholes: esquemas explícito, implícito y de Crank-Nicolson
- Valoración de opciones americanas: Monte Carlo de mínimos cuadrados (Longstaff-Schwartz) y parada óptima

Non Core

- Aprendizaje automático en finanzas cuantitativas: cobertura profunda, aprendizaje por refuerzo para ejecución y modelos de factores (2020-2025)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Implementar un valuador por Monte Carlo para una opción europea y estimar su error estándar [Familiarizarse]
2. Aplicar técnicas de reducción de varianza para acelerar la valoración por Monte Carlo de derivados dependientes de la trayectoria [Usar]

NonCore:

3. Evaluar enfoques de cobertura profunda y aprendizaje por refuerzo para la valoración de derivados y compararlos con la cobertura clásica de Black-Scholes [Evaluar]

Capítulo 5

Cuerpo del conocimiento de Física

Las 12 áreas de conocimiento en Física son:

5.1 Mecánica Clásica (CME) (Pág. 226)
- 5.1.1 Cinemática y Dinámica de Partículas (Pág. 227)
- 5.1.2 Trabajo, Energía y Fuerzas Conservativas (Pág. 227)
- 5.1.3 Sistemas de Partículas y Momento Lineal (Pág. 228)
- 5.1.4 Movimiento Rotacional y Momento Angular (Pág. 229)
- 5.1.5 Dinámica del Cuerpo Rígido en 3D (Pág. 229)
- 5.1.6 Marcos de Referencia no Inerciales y Fuerzas Ficticias (Pág. 230)
- 5.1.7 Mecánica Lagrangiana y Hamiltoniana (Pág. 230)
- 5.1.8 Movimiento de Fuerzas Centrales y Mecánica Orbital (Pág. 231)
- 5.1.9 Teoría de Pequeñas Oscilaciones y Osciladores Acoplados (Pág. 232)
- 5.1.10 Fundamentos de la Mecánica de Medios Continuos (Pág. 232)
- 5.1.11 Elasticidad (Pág. 233)
- 5.1.12 Fluidos: Hidrostática, Hidrodinámica y Viscosidad (Pág. 233)
- 5.1.13 Vibraciones y Ondas (Pág. 234)
- 5.1.14 Fenómeno de Transporte (Pág. 234)

5.2 Electromagnetismo (ELM) (Pág. 235)
- 5.2.1 Electrostática: Campo Eléctrico y Ley de Gauss (Pág. 235)
- 5.2.2 Potencial Eléctrico y Ecuaciones de Poisson/Laplace (Pág. 236)
- 5.2.3 Energía Electrostática y Dieléctricos (Pág. 236)
- 5.2.4 Magnetostática: Leyes de Biot-Savart y Ampère (Pág. 237)
- 5.2.5 Propiedades Magnéticas de la Materia (Pág. 237)
- 5.2.6 Electrodinámica: Ley de Faraday e Inducción (Pág. 238)
- 5.2.7 Ecuaciones de Maxwell y Transformaciones de Norma (Pág. 238)
- 5.2.8 Ondas Electromagnéticas y Propagación en Medios (Pág. 239)
- 5.2.9 Teoría de la Radiación y Radiación Dipolar (Pág. 239)
- 5.2.10 Relatividad Especial en Electromagnetismo (Pág. 240)
- 5.2.11 Corriente Eléctrica, Circuitos y Leyes de Kirchhoff (Pág. 240)

5.3 Termodinámica y Mecánica Estadística (TSM) (Pág. 240)
- 5.3.1 Leyes de la Termodinámica y Funciones de Estado (Pág. 241)
- 5.3.2 Teoría Cinética de los Gases (Pág. 241)
- 5.3.3 Máquinas Térmicas, Entropía y la Segunda Ley (Pág. 242)
- 5.3.4 Potenciales Termodinámicos y Relaciones de Maxwell (Pág. 242)
- 5.3.5 Transiciones de Fase y Fenómenos Críticos (Pág. 242)
- 5.3.6 Colectivos Microcanónico, Canónico y Gran Canónico (Pág. 243)
- 5.3.7 Estadística Cuántica: Fermi-Dirac y Bose-Einstein (Pág. 243)
- 5.3.8 Radiación de Cuerpo Negro y Fonones (Pág. 244)
- 5.3.9 Mecánica Estadística de No-Equilibrio (Pág. 244)
- 5.3.10 Aplicaciones a Sistemas Biológicos y Complejos (Pág. 244)

5.4 Mecánica Cuántica (QME) (Pág. 245)
- 5.4.1 Orígenes de la Teoría Cuántica y Dualidad Onda-Partícula (Pág. 245)
- 5.4.2 La Ecuación de Schrödinger y Funciones de Onda (Pág. 245)
- 5.4.3 Formalismo: Espacio de Hilbert y Notación de Dirac (Pág. 246)
- 5.4.4 Potenciales Unidimensionales (Pozos, Barreras, Tunelamiento) (Pág. 246)
- 5.4.5 El Oscilador Armónico Cuántico (Pág. 247)
- 5.4.6 Momento Angular y Física del Espín (Pág. 247)
- 5.4.7 El átomo de Hidrógeno y Potenciales Centrales (Pág. 247)
- 5.4.8 Teoría de Perturbaciones Independiente y Dependiente del Tiempo (Pág. 248)
- 5.4.9 Partículas Idénticas y Simetría de Intercambio (Pág. 248)
- 5.4.10 Teoría de la Dispersión y Secciones Eficaces (Pág. 248)

5.5 Relatividad y Cosmología (RCO) (Pág. 249)
- 5.5.1 Relatividad Especial: Transformaciones de Lorentz y Cuadrivectores (Pág. 249)
- 5.5.2 Dinámica Relativista y Energía-Momento (Pág. 250)
- 5.5.3 Fundamentos de la Relatividad General: Principio de Equivalencia (Pág. 250)
- 5.5.4 Curvatura del Espacio-Tiempo y Ecuaciones de Campo de Einstein (Pág. 250)
- 5.5.5 Geometría de Schwarzschild y Agujeros Negros (Pág. 251)
- 5.5.6 Ondas Gravitacionales (Pág. 251)
- 5.5.7 Teoría del Big Bang y Ecuaciones de Friedmann (Pág. 251)
- 5.5.8 Fondo Cósmico de Microondas (CMB) (Pág. 252)
- 5.5.9 Materia Oscura y Energía Oscura (Pág. 252)
- 5.5.10 Inflación del Universo Temprano y Nucleosíntesis (Pág. 252)

5.6 Física de la Materia Condensada (SSP) (Pág. 253)
- 5.6.1 Estructuras Cristalinas y Redes Recíprocas (Pág. 253)
- 5.6.2 Difracción de Rayos X y Dinámica de Red (Pág. 253)
- 5.6.3 Modelo de Electrones Libres y Teoría de Bandas de Sólidos (Pág. 254)
- 5.6.4 Semiconductores, Aislantes y Metales (Pág. 254)
- 5.6.5 Propiedades Magnéticas: Para-, Dia- y Ferromagnetismo (Pág. 254)
- 5.6.6 Superconductividad: Teoría BCS y Alta Tc (Pág. 255)
- 5.6.7 Física de la Materia Blanda y Polímeros (Pág. 255)
- 5.6.8 Nanoestructuras y Sistemas de Baja Dimensionalidad (Pág. 255)
- 5.6.9 Espintrónica y Aislantes Topológicos (Pág. 256)

- 5.6.10 Sistemas de Electrones Fuertemente Correlacionados (Pág. 256)

5.7 Física Nuclear y de Partículas (NPP) (Pág. 256)
- 5.7.1 Estructura Nuclear, Energía de Enlace y Modelo de Capas (Pág. 257)
- 5.7.2 Radiactividad y Desintegraciones Nucleares (Alfa, Beta, Gamma) (Pág. 257)
- 5.7.3 Reacciones Nucleares, Fisión y Fusión (Pág. 258)
- 5.7.4 Aceleradores y Detectores de Partículas (Pág. 258)
- 5.7.5 El Modelo Estándar de la Física de Partículas (Pág. 258)
- 5.7.6 Quarks, Leptones y Bosones de Norma (Pág. 259)
- 5.7.7 Simetrías y Leyes de Conservación (CPT) (Pág. 259)
- 5.7.8 Interacciones Fundamentales y Diagramas de Feynman (Pág. 259)
- 5.7.9 Mecanismo de Higgs y más allá del Modelo Estándar (Pág. 260)
- 5.7.10 Física de Neutrinos y Oscilación (Pág. 260)

5.8 Información y Computación Cuántica (QIC) (Pág. 260)
- 5.8.1 Qubits, Superposición y Entrelazamiento (Pág. 261)
- 5.8.2 Comuertas y Circuitos Lógicos Cuánticos (Pág. 261)
- 5.8.3 Algoritmos Cuánticos (Shor, Grover) (Pág. 262)
- 5.8.4 Teletransportación y Criptografía Cuántica (Pág. 262)
- 5.8.5 Decoherencia y Corrección de Errores Cuánticos (Pág. 262)
- 5.8.6 Realizaciones Físicas de Computadoras Cuánticas (Pág. 263)
- 5.8.7 Sensores Cuánticos y Metrología (Pág. 263)
- 5.8.8 Redes Cuánticas y Comunicación (Pág. 263)

5.9 Óptica y Fotónica (OPH) (Pág. 264)
- 5.9.1 Óptica Geométrica e Instrumentos Ópticos (Pág. 264)
- 5.9.2 Óptica Ondulatoria: Interferencia y Difracción (Pág. 265)
- 5.9.3 Polarización y Óptica de Cristales (Pág. 265)
- 5.9.4 Física de Láseres: Principios y Aplicaciones (Pág. 265)
- 5.9.5 Óptica No Lineal y Conversión de Frecuencia (Pág. 266)
- 5.9.6 Fibra Óptica y Cristales Fotónicos (Pág. 266)
- 5.9.7 Óptica Ultrarrápida y Espectroscopía (Pág. 266)
- 5.9.8 Nano-óptica y Plasmónica (Pág. 267)

5.10 Física Computacional y Basada en Datos (CDP) (Pág. 267)
- 5.10.1 Métodos Numéricos para Ecuaciones Diferenciales en Física (Pág. 268)
- 5.10.2 Métodos y Simulaciones de Monte Carlo (Pág. 268)
- 5.10.3 Computación de Alto Rendimiento (HPC) para Simulaciones a Gran Escala (Pág. 268)
- 5.10.4 Aprendizaje Automático para el Descubrimiento Físico (Pág. 269)
- 5.10.5 Análisis de Datos y Cuantificación de la Incertidumbre (Pág. 269)
- 5.10.6 Modelado de Sistemas Adaptativos Complejos (Pág. 269)
- 5.10.7 Redes Neuronales para Sistemas Cuánticos (Pág. 270)
- 5.10.8 Redes Neuronales Informadas por la Física (PINNs) (Pág. 270)

5.11 Biofísica y Física Médica (BMP) (Pág. 271)
- 5.11.1 Física de las Macromoléculas Biológicas (Pág. 271)
- 5.11.2 Biomecánica y Fuerzas Celulares (Pág. 271)
- 5.11.3 Física Neuronal y Procesamiento de Información en el Cerebro (Pág. 272)
- 5.11.4 Física de Imágenes (MRI, CT, PET, Ultrasonido) (Pág. 272)
- 5.11.5 Radioterapia y Dosimetría (Pág. 272)
- 5.11.6 Mecánica Estadística de Redes Biológicas (Pág. 273)
- 5.11.7 Microfluídica y Lab-on-a-Chip (Pág. 273)

5.12 Astrofísica y Ciencias del Espacio (ASS) (Pág. 273)
- 5.12.1 Estructura y Evolución Estelar (Pág. 274)
- 5.12.2 Ciencia Planetaria y Exoplanetas (Pág. 274)
- 5.12.3 Dinámica y Estructura Galáctica (Pág. 274)
- 5.12.4 Astrofísica de Altas Energías (Rayos X, Rayos Gamma) (Pág. 275)
- 5.12.5 Técnicas Observacionales y Telescopios (Pág. 275)
- 5.12.6 Física Solar y Clima Espacial (Pág. 275)
- 5.12.7 Astronomía de Multimensajeros (Ondas Gravitacionales + Neutrinos) (Pág. 276)

5.1. Mecánica Clásica (CME)

La Mecánica Clásica proporciona los principios fundamentales que gobiernan el movimiento de objetos macroscópicos, desde partículas hasta cuerpos rígidos y medios continuos. Engloba dinámica newtoniana, leyes de conservación, formalismos Lagrangiano y Hamiltoniano, sistemas oscilatorios y mecánica celeste.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.1.1 Cinemática y Dinámica de Partículas (Pág. 227)	1	1
⭐ 5.1.2 Trabajo, Energía y Fuerzas Conservativas (Pág. 227)	1	1
⭐ 5.1.3 Sistemas de Partículas y Momento Lineal (Pág. 228)	1	1
⭐ 5.1.4 Movimiento Rotacional y Momento Angular (Pág. 229)	1	1
⭐ 5.1.5 Dinámica del Cuerpo Rígido en 3D (Pág. 229)	1	1
⭐ 5.1.6 Marcos de Referencia no Inerciales y Fuerzas Ficticias (Pág. 230)	1	1
⭐ 5.1.7 Mecánica Lagrangiana y Hamiltoniana (Pág. 230)	1	1
⭐ 5.1.8 Movimiento de Fuerzas Centrales y Mecánica Orbital (Pág. 231)	1	1
⭐ 5.1.9 Teoría de Pequeñas Oscilaciones y Osciladores Acoplados (Pág. 232)	1	1
5.1.10 Fundamentos de la Mecánica de Medios Continuos (Pág. 232)	Electivo	
⭐ 5.1.11 Elasticidad (Pág. 233)	1	1
⭐ 5.1.12 Fluidos: Hidrostática, Hidrodinámica y Viscosidad (Pág. 233)	1	1
⭐ 5.1.13 Vibraciones y Ondas (Pág. 234)	1	1
⭐ 5.1.14 Fenómeno de Transporte (Pág. 234)	1	1

Cuadro 5.1: Lista de KUs del área de Mecánica Clásica.

5.1.1. CME/Cinemática y Dinámica de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Descripción matemática del movimiento (cinemática) y estudio de las fuerzas que causan o modifican el movimiento (dinámica) para partículas puntuales en varios sistemas coordenados y campos de fuerza.

Temas:

Core

- Descripción vectorial de posición, velocidad y aceleración en coordenadas cartesianas, polares y esféricas
- Tres leyes de Newton del movimiento y marcos de referencia inerciales
- Formulación y solución de ecuaciones de movimiento para fuerzas constantes y variables
- Análisis del movimiento de proyectiles con y sin resistencia del aire ($F_{arrastre} = -bv$ o $-cv^2$)
- Fuerzas de fricción estática y cinética, incluyendo el ángulo de reposo
- Movimiento en medios resistivos incluyendo modelos de arrastre lineal y cuadrático
- Movimiento bajo restricciones como un péndulo o cuentas en alambres
- Introducción a osciladores no lineales y análisis del espacio de fase
- Integración numérica de ecuaciones de movimiento usando métodos de Euler y Runge-Kutta

Electivo

- Movimiento de partículas cargadas en campos electromagnéticos bajo la fuerza de Lorentz: $F = q(E + v \times B)$

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Formular las ecuaciones de movimiento para una partícula en una, dos y tres dimensiones dada una ley de fuerza [Usar]
2. Resolver analíticamente la trayectoria de un proyectil bajo gravedad constante despreciando la resistencia del aire [Usar]
3. Calcular la velocidad terminal de un objeto que cae a través de un fluido con arrastre cuadrático [Evaluar]
4. Determinar puntos de equilibrio y estabilidad para una partícula en un potencial unidimensional [Usar]
5. Identificar todas las fuerzas que actúan sobre una partícula en un escenario mecánico dado y dibujar un diagrama de cuerpo libre correcto [Familiarizarse]
6. Implementar un algoritmo numérico simple como el método de Euler para computar el movimiento de un oscilador armónico amortiguado [Usar]
7. Construir e interpretar diagramas de espacio de fase (x vs. $\dot{x}$) para sistemas oscilatorios simples [Evaluar]

Elective:

8. Analizar el movimiento de una partícula cargada en campos eléctricos y magnéticos uniformes [Usar]

5.1.2. CME/Trabajo, Energía y Fuerzas Conservativas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Conceptos de energía en mecánica: teorema del trabajo y la energía, energía cinética y potencial, conservación de la energía mecánica y propiedades de los campos de fuerzas conservativas.

Temas:

Core

- Definición de trabajo como una integral de línea $W = \int \vec{F} \cdot d\vec{r}$
- Teorema del trabajo y la energía que relaciona el trabajo total con el cambio en la energía cinética: $W_{total} = \Delta K$
- Energía cinética para partículas individuales y sistemas de partículas
- Fuerzas conservativas e independencia de la trayectoria del trabajo
- Función de energía potencial $U(\vec{r})$ y la relación de fuerza $\vec{F} = -\nabla U$
- Conservación de la energía mecánica total $E = K + U$
- Diagramas de energía e interpretación de puntos de retorno y estabilidad del equilibrio

- Trabajo realizado por fuerzas no conservativas y mecanismos de disipación de energía
- Superficies de energía potencial en Dos y Tres dimensiones

Elective

- Consideraciones energéticas para fuerzas dependientes de la velocidad, como las fuerzas magnéticas y de arrastre

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el trabajo realizado por una fuerza a lo largo de una trayectoria especificada utilizando integrales de línea [Usar]
2. Determinar si un campo de fuerzas dado es conservativo y hallar su función de energía potencial [Usar]
3. Aplicar la conservación de la energía mecánica para resolver velocidades y posiciones en el movimiento unidimensional [Usar]
4. Analizar un diagrama de energía para identificar puntos de retorno, regiones permitidas y estabilidad del equilibrio [Evaluar]
5. Calcular la potencia instantánea entregada por una fuerza: $P = \vec{F} \cdot \vec{v}$ [Usar]
6. Estimar la energía disipada por fuerzas no conservativas en un proceso mecánico [Evaluar]
7. Localizar y clasificar puntos críticos como máximos, mínimos y puntos de silla en una superficie de potencial bidimensional [Usar]

Elective:

8. Explicar por qué las fuerzas magnéticas no realizan trabajo sobre partículas cargadas a pesar de cambiar la dirección de su velocidad [Familiarizarse]

5.1.3. CME/Sistemas de Partículas y Momento Lineal (Core Tier1: 1 hr, Core Tier2: 1 hr)

Dinámica de sistemas de múltiples partículas: movimiento del centro de masa, conservación del momento lineal, fuerzas internas y externas, y sistemas de masa variable.

Temas:

Core

- Definición de centro de masa: $\vec{R}_{CM} = \frac{1}{M}\sum m_i \vec{r}_i$
- Momento lineal de una partícula y de un sistema: $\vec{p} = m\vec{v}$ y $\vec{P} = M\vec{V}_{CM}$
- Segunda ley de Newton para sistemas: $\vec{F}_{ext} = \frac{d\vec{P}}{dt}$
- Conservación del momento lineal para sistemas aislados
- Distinción entre fuerzas internas y externas en un sistema
- Colisiones de dos cuerpos, incluyendo casos elásticos, inelásticos y completamente inelásticos
- Coeficiente de restitución y su significado físico en las colisiones
- Sistemas con masa variable, como la ecuación del cohete: $M\frac{dV}{dt} = v_{ex}\frac{dM}{dt}$
- Sistemas de muchas partículas y su dinámica del centro de masa

Elective

- Distribuciones continuas de masa y el cálculo de su centro de masa mediante integración

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Localizar el centro de masa para un conjunto discreto de partículas en Una, Dos y Tres dimensiones [Usar]
2. Determinar el momento lineal total de un sistema a partir del movimiento de su centro de masa [Usar]
3. Aplicar la conservación del momento lineal para resolver problemas de colisión en Una y Dos dimensiones [Usar]
4. Clasificar colisiones como elásticas, inelásticas o completamente inelásticas basándose en la conservación de la energía cinética [Familiarizarse]
5. Calcular el coeficiente de restitución a partir de las velocidades antes y después de la colisión [Usar]
6. Resolver la ecuación del cohete para determinar la velocidad en función de la masa expulsada [Usar]

7. Analizar cómo las fuerzas internas afectan a las partículas individuales pero no al momento total de un sistema [Evaluar]

Elective:

8. Integrar para hallar el centro de masa de cuerpos continuos simétricos como varillas, discos y esferas [Usar]

5.1.4. CME/Movimiento Rotacional y Momento Angular (Core Tier1: 1 hr, Core Tier2: 1 hr)

Cinemática y dinámica rotacional, torque, momento de inercia, energía rotacional y conservación del momento angular para partículas y sistemas.

Temas:

Core

- Cinemática rotacional incluyendo desplazamiento angular, velocidad y aceleración (θ, ω, α)
- Definición de torque: $\vec{\tau} = \vec{r} \times \vec{F}$ y el cálculo de su magnitud
- Momento angular de una partícula: $\vec{L} = \vec{r} \times \vec{p}$
- Forma rotacional de la segunda ley de Newton: $\vec{\tau}_{net} = \frac{d\vec{L}}{dt}$
- Momento de inercia $I = \sum m_i r_i^2$ y el teorema de los ejes paralelos
- Energía cinética rotacional: $K_{rot} = \frac{1}{2} I\omega^2$
- Conservación del momento angular para sistemas aislados
- Movimiento giroscópico y el fenómeno de la precesión
- Dinámica de rodadura sin deslizamiento: $v_{CM} = R\omega$

Electivo

- El efecto Coriolis como manifestación de la dinámica rotacional en marcos no inerciales

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Relacionar las magnitudes cinemáticas lineales y angulares para el movimiento circular [Usar]
2. Calcular el torque respecto a un eje dado para una fuerza aplicada en un punto específico [Usar]
3. Determinar el momento de inercia para geometrías simples como una varilla, un aro, un disco o una esfera [Usar]
4. Aplicar la conservación del momento angular para analizar cambios en la velocidad de rotación cuando cambia el momento de inercia [Evaluar]
5. Calcular la energía cinética rotacional de un cuerpo rígido en rotación [Usar]
6. Analizar el movimiento de objetos que ruedan sin deslizar por un plano inclinado [Evaluar]
7. Predecir la dirección de la precesión para una peonza o un giroscopio en rotación [Evaluar]

Elective:

8. Explicar el efecto Coriolis en términos de marcos de referencia rotatorios y su efecto sobre los proyectiles [Familiarizarse]

5.1.5. CME/Dinámica del Cuerpo Rígido en 3D (Core Tier1: 1 hr, Core Tier2: 1 hr)

Movimiento tridimensional de cuerpos rígidos: tensor de inercia, ejes principales, ecuaciones de Euler y movimiento rotacional complejo.

Temas:

Core

- Tensor de inercia I_{ij} y su representación matricial
- Ejes principales y la diagonalización del tensor de inercia
- Ecuaciones de Euler para la rotación del cuerpo rígido: $I_1\dot{\omega}_1 = (I_2 - I_3)\omega_2\omega_3 + \tau_1$
- Rotación libre de peonzas asimétricas y análisis de estabilidad de los ejes de rotación
- Casos en los que el vector de momento angular no es paralelo al vector de velocidad angular
- Movimiento libre de torque y polhodias en la dinámica del cuerpo rígido
- ángulos de Euler (ϕ, θ, ψ) y la descripción de la orientación en el espacio

Electivo

- Rotación forzada y el fenómeno de la nutación
- Colisiones entre cuerpos rígidos en Tres dimensiones

- Simulación numérica de la dinámica compleja de cuerpos rígidos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Construir el tensor de inercia para un sistema de masas discretas [Usar]
2. Diagonalizar un tensor de inercia dado para hallar los momentos principales y los ejes principales [Usar]
3. Aplicar las ecuaciones de Euler para analizar la rotación libre de torque de peonzas simétricas [Usar]
4. Interpretar por qué los vectores de momento angular y velocidad angular no son paralelos para cuerpos asimétricos [Evaluar]
5. Calcular los ángulos de Euler para describir la orientación de un cuerpo rígido en el espacio [Usar]

Elective:

6. Simular numéricamente el movimiento tridimensional de un cuerpo rígido bajo torques dados [Usar]
7. Analizar la nutación en una peonza simétrica pesada [Evaluar]

5.1.6. CME/Marcos de Referencia no Inerciales y Fuerzas Ficticias (Core Tier1: 1 hr, Core Tier2: 1 hr)

Física en marcos de referencia acelerados: fuerzas centrífuga, de Coriolis y de Euler, con aplicaciones a la rotación de la Tierra.

Temas:

Core

- Transformaciones entre marcos de referencia inerciales y acelerados
- Fuerza centrífuga definida como $F_{cent} = -m\vec{\omega} \times (\vec{\omega} \times \vec{r}')$
- Fuerza de Coriolis definida como $F_{Cor} = -2m\vec{\omega} \times \vec{v}'$
- Fuerza de Euler para rotación no constante: $F_{Euler} = -m\frac{d\vec{\omega}}{dt} \times \vec{r}'$
- Gravedad efectiva en la Tierra en rotación: $g_{eff} = g - \vec{\omega} \times (\vec{\omega} \times \vec{R})$
- Péndulo de Foucault como demostración de la rotación de la Tierra
- Efecto Coriolis en los patrones climáticos globales y las corrientes oceánicas

Electivo

- Precesión de Larmor en campos magnéticos vista como un efecto de marco rotatorio
- Fuerzas de marea como efectos gravitatorios diferenciales en marcos co-rotatorios
- Conexión con el principio de equivalencia de la Relatividad General

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Transformar la segunda ley de Newton a un marco de referencia uniformemente rotatorio [Usar]
2. Calcular las fuerzas centrífuga y de Coriolis para posiciones y velocidades dadas en un marco rotatorio [Usar]
3. Predecir la dirección de desviación de los proyectiles debido a la rotación de la Tierra mediante el efecto Coriolis [Evaluar]
4. Explicar la precesión de un péndulo de Foucault en términos de la rotación de la Tierra [Familiarizarse]
5. Calcular la aceleración gravitatoria efectiva en diferentes latitudes de la Tierra [Usar]

Elective:

6. Analizar la precesión de Larmor utilizando el concepto de marcos de referencia rotatorios [Evaluar]
7. Comparar las fuerzas de marea con las fuerzas centrífugas en sistemas co-rotatorios [Evaluar]

5.1.7. CME/Mecánica Lagrangiana y Hamiltoniana (Core Tier1: 1 hr, Core Tier2: 1 hr)

Formulaciones avanzadas de la mecánica: principio de mínima acción, formulación lagrangiana, mecánica hamiltoniana y transformaciones canónicas.

Temas:
Core
- Principio de mínima acción y principio de acción estacionaria de Hamilton
- Definición de Lagrangiano: $L = T - V$ para sistemas conservativos
- Ecuaciones de movimiento de Euler-Lagrange: $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) = \frac{\partial L}{\partial q_i}$
- Coordenadas generalizadas y el tratamiento de restricciones holonómicas
- Coordenadas cíclicas y las leyes de conservación asociadas
- Transformada de Legendre del Lagrangiano al Hamiltoniano: $H = \sum p_i \dot{q}_i - L$
- Ecuaciones de Hamilton: $\dot{q}_i = \frac{\partial H}{\partial p_i}$ y $\dot{p}_i = -\frac{\partial H}{\partial q_i}$

Electivo
- Paréntesis de Poisson: $\{f, g\} = \sum \left(\frac{\partial f}{\partial q_i}\frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial g}{\partial q_i}\right)$
- Transformaciones canónicas y sus funciones generatrices
- Variables de acción-ángulo para sistemas hamiltonianos periódicos

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Construir el Lagrangiano para sistemas mecánicos simples en coordenadas generalizadas adecuadas [Usar]
2. Deducir las ecuaciones de movimiento a partir de las ecuaciones de Euler-Lagrange [Usar]
3. Identificar coordenadas cíclicas y sus magnitudes conservadas asociadas en un Lagrangiano dado [Evaluar]
4. Formular el Hamiltoniano a partir de un sistema lagrangiano dado [Usar]
5. Resolver las ecuaciones de Hamilton para sistemas simples como el oscilador armónico [Usar]

Elective:
6. Calcular los paréntesis de Poisson para pares de variables dinámicas [Usar]
7. Aplicar transformaciones canónicas para simplificar la descripción de sistemas hamiltonianos [Usar]

5.1.8. CME/Movimiento de Fuerzas Centrales y Mecánica Orbital (Core Tier1: 1 hr, Core Tier2: 1 hr)

Movimiento bajo fuerzas centrales: leyes de Kepler, dinámica orbital, potencial gravitatorio y el problema de los dos cuerpos.

Temas:
Core
- Fuerzas centrales $\vec{F} = F(r)\hat{r}$ y la conservación del momento angular
- Masa reducida y la reducción del problema de los dos cuerpos a un problema de un solo cuerpo
- Potencial efectivo: $U_{eff}(r) = U(r) + \frac{L^2}{2\mu r^2}$ y puntos de retorno
- Las Tres leyes de Kepler del movimiento planetario
- Ecuación de la órbita: $\frac{1}{r} = \frac{GM\mu^2}{L^2}(1 + e\cos\theta)$ para leyes del inverso del cuadrado
- Elementos orbitales incluyendo el semieje mayor, la excentricidad y la inclinación
- Clasificación energética de las órbitas en estados ligados ($E < 0$) y no ligados ($E \geq 0$)

Electivo
- Teoría de la dispersión de Rutherford y el concepto de sección eficaz
- Precesión de órbitas en potenciales gravitatorios no Newtonianos
- Problema restringido de los Tres cuerpos y los cinco puntos de Lagrange

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Reducir un problema de fuerza central de dos cuerpos a un problema equivalente de Un solo cuerpo utilizando la masa reducida [Usar]
2. Analizar el potencial efectivo para determinar los puntos de retorno y los tipos de órbita permitidos [Evaluar]
3. Deducir la Tercera ley de Kepler $T^2 \propto a^3$ a partir de los principios de la gravitación newtoniana [Usar]
4. Calcular parámetros orbitales como el periodo y la energía a partir de las condiciones iniciales [Usar]

5. Clasificar una órbita como circular, elíptica, parabólica o hiperbólica según la energía total [Evaluar]

Elective:

6. Calcular ángulos de dispersión en experimentos de dispersión de Rutherford [Usar]
7. Localizar los cinco puntos de Lagrange en el problema circular restringido de los Tres cuerpos [Usar]

5.1.9. CME/Teoría de Pequeñas Oscilaciones y Osciladores Acoplados (Core Tier1: 1 hr, Core Tier2: 1 hr)

Pequeñas oscilaciones alrededor del equilibrio: modos normales, autofrecuencias, amortiguamiento, forzamiento y sistemas acoplados.

Temas:

Core

- Análisis de estabilidad de los puntos de equilibrio utilizando las segundas derivadas de la energía potencial
- Aproximación de pequeñas oscilaciones mediante expansión de Taylor alrededor de un equilibrio estable
- Osciladores acoplados y la formulación de las ecuaciones de movimiento en forma matricial
- Modos normales y autofrecuencias derivadas del problema de autovalores
- Coordenadas normales que diagonalizan el sistema mecánico
- Osciladores amortiguados: regímenes subamortiguado, críticamente amortiguado y sobreamortiguado
- Oscilaciones forzadas y la física de los fenómenos de resonancia

Electivo

- Sistemas continuos y la ecuación de onda como límite de muchos osciladores acoplados
- Oscilaciones no lineales y el estudio de los efectos anarmónicos
- Resonancia paramétrica e introducción a la ecuación de Mathieu

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Linealizar las ecuaciones de movimiento alrededor de un punto de equilibrio estable [Usar]
2. Hallar las frecuencias de los modos normales y los autovectores para sistemas de Dos osciladores acoplados [Usar]
3. Resolver la ecuación del oscilador armónico amortiguado para los Tres regímenes de amortiguamiento [Usar]
4. Analizar la amplitud y la respuesta de fase de un oscilador armónico forzado cerca de la resonancia [Evaluar]
5. Transformar a coordenadas normales para desacoplar las ecuaciones de movimiento de un sistema [Usar]

Elective:

6. Deducir la ecuación de onda a partir del límite continuo de muchos osciladores acoplados [Usar]
7. Investigar las correcciones anarmónicas al potencial del oscilador armónico simple [Evaluar]

5.1.10. CME/Fundamentos de la Mecánica de Medios Continuos

Introducción a los medios continuos: tensores de tensión y deformación, elasticidad, mecánica de fluidos y leyes de conservación.

Temas:

Electivo

- Hipótesis del continuo y definición de campos de densidad
- Tensor de tensión σ_{ij} y el concepto de fuerza por unidad de área
- Tensor de deformación ϵ_{ij} como medida de la deformación local
- Ley de Hooke para materiales isotrópicos: $\sigma_{ij} = \lambda\epsilon_{kk}\delta_{ij} + 2\mu\epsilon_{ij}$
- Ecuación de Navier-Stokes para fluidos viscosos: $\rho\left(\frac{\partial\vec{v}}{\partial t} + \vec{v}\cdot\nabla\vec{v}\right) = -\nabla p + \mu\nabla^2\vec{v} + \vec{f}$
- Principio de Bernoulli para flujo de fluidos no viscosos e incompresibles
- Conservación de la masa y la ecuación de continuidad: $\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) = 0$

- Conservación del momento expresada en forma de mecánica de medios continuos
- Propagación de ondas en medios elásticos incluyendo ondas P y ondas S
- Viscosidad y reología de fluidos no Newtonianos

Aprendizaje esperado (*Learning Outcomes*):

Elective:

1. Interpretar el significado físico de las componentes del tensor de tensión [Familiarizarse]
2. Calcular las componentes del tensor de deformación para deformaciones simples [Usar]
3. Aplicar la ley de Hooke para relacionar tensión y deformación en materiales elásticos isotrópicos [Usar]
4. Simplificar la ecuación de Navier-Stokes para casos especiales como flujo estacionario o no viscoso [Usar]
5. Aplicar el principio de Bernoulli para calcular variaciones de presión en el flujo de fluidos [Usar]
6. Deducir la ecuación de continuidad a partir del principio de conservación de la masa [Usar]
7. Analizar diferentes modos de propagación de ondas en sólidos elásticos [Evaluar]

5.1.11. CME/Elasticidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades elásticas de sólidos bajo carga axial, de corte y volumétrica: esfuerzo, deformación unitaria y módulos elásticos escalares.

Temas:

Core

- Definición de esfuerzo (fuerza/área) y deformación unitaria ($\Delta L/L$)
- Módulo de Young y ley de Hooke para deformación longitudinal: $\sigma = E\epsilon$
- Módulo de corte (rigidez) y deformación por cizalladura
- Módulo de volumen (compresibilidad) y deformación volumétrica
- Límite elástico, región plástica y punto de ruptura en el diagrama esfuerzo-deformación

Electivo

- Energía elástica almacenada en un material deformado

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar la ley de Hooke escalar para calcular la deformación de una barra bajo carga axial [Usar]
2. Calcular el módulo de Young, de corte o de volumen a partir de datos experimentales esfuerzo-deformación [Usar]
3. Identificar la región elástica, el límite elástico y el punto de fractura en una curva esfuerzo-deformación [Familiarizarse]

Elective:

4. Calcular la energía potencial elástica almacenada en un resorte o barra deformada [Usar]

5.1.12. CME/Fluidos: Hidrostática, Hidrodinámica y Viscosidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estática y dinámica de fluidos incompresibles: presión, empuje, ecuaciones de continuidad y de Bernoulli, e introducción al flujo viscoso.

Temas:

Core

- Densidad, presión y su variación con la profundidad en un fluido en reposo: $p = p_0 + \rho gh$
- Principio de Pascal y sus aplicaciones (prensa hidráulica)
- Principio de Arquímedes, empuje y flotación
- Medición de presión con manómetros y barómetros
- Ecuación de continuidad para flujo incompresible: $A_1 v_1 = A_2 v_2$
- Ecuación de Bernoulli y conservación de energía en flujo ideal
- Aplicaciones: tubo de Venturi, teorema de Torricelli, tubo de Pitot

Electivo

- Viscosidad y ley de Poiseuille para flujo laminar en tuberías
- Número de Reynolds y transición a flujo turbulento
- Ley de Stokes para la fuerza de arrastre sobre una esfera en un fluido viscoso

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la presión en un punto de un fluido en reposo dado la profundidad y densidad [Usar]
2. Aplicar el principio de Arquímedes para determinar si un objeto flota, se hunde o permanece en equilibrio [Usar]
3. Aplicar las ecuaciones de continuidad y de Bernoulli para resolver problemas de flujo en tuberías de sección variable [Usar]
4. Interpretar lecturas de manómetros y barómetros para determinar presiones absolutas y manométricas [Evaluar]

Elective:

5. Aplicar la ley de Poiseuille para calcular el caudal de un fluido viscoso en un tubo cilíndrico [Usar]
6. Estimar el número de Reynolds de un flujo y predecir si es laminar o turbulento [Evaluar]

5.1.13. CME/Vibraciones y Ondas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Movimiento armónico simple, oscilaciones amortiguadas y forzadas, y propagación, superposición y resonancia de ondas mecánicas.

Temas:
Core

- Movimiento armónico simple (MAS): ecuación de movimiento, posición, velocidad y aceleración
- Sistemas oscilantes: resorte-masa y péndulo simple/físico
- Energía en el movimiento armónico simple
- Oscilaciones amortiguadas y forzadas; introducción a la resonancia
- Ondas transversales y longitudinales; velocidad de propagación
- Ecuación de onda y velocidad de una onda en una cuerda tensa: $v = \sqrt{T/\mu}$
- Principio de superposición y ondas estacionarias
- Ondas sonoras, intensidad y nivel de intensidad sonora (decibeles)

Electivo

- Efecto Doppler para ondas sonoras
- Pulsaciones (beats) e interferencia de ondas de frecuencias cercanas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Deducir la ecuación de movimiento y el período de un sistema resorte-masa y de un péndulo simple [Usar]
2. Calcular la energía cinética, potencial y total de un oscilador armónico simple en cualquier instante [Usar]
3. Analizar la respuesta en amplitud de un oscilador forzado cerca de la frecuencia de resonancia [Evaluar]
4. Calcular la velocidad de propagación de una onda en una cuerda a partir de la tensión y la densidad lineal [Usar]
5. Determinar las frecuencias y longitudes de onda de los modos normales de una onda estacionaria en una cuerda fija [Usar]
6. Calcular el nivel de intensidad sonora en decibeles a partir de la intensidad de una onda sonora [Usar]

Elective:

7. Aplicar la fórmula del efecto Doppler para fuente y/o observador en movimiento [Usar]

5.1.14. CME/Fenómeno de Transporte (Core Tier1: 1 hr, Core Tier2: 1 hr)

Introducción a los fenómenos de transporte en medios continuos: conducción de calor y difusión de masa como procesos de flujo impulsados por gradientes.

Temas:
Core

- Conducción de calor y la ley de Fourier: $\vec{q} = -k\nabla T$

- Conductividad térmica de materiales y resistencia térmica en paredes compuestas
- Difusión molecular y la primera ley de Fick: $\vec{J} = -D\nabla C$

Electivo

- Analogía entre el transporte de calor, masa y momento (números de Prandtl y Schmidt)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar la ley de Fourier para calcular el flujo de calor a través de una pared plana o compuesta [Usar]
2. Aplicar la primera ley de Fick para calcular el flujo difusivo de una especie en un gradiente de concentración [Usar]

Elective:

3. Identificar analogías matemáticas entre los procesos de transporte de calor, masa y momento [Familiarizarse]

5.2. Electromagnetismo (ELM)

El electromagnetismo estudia el comportamiento de las cargas eléctricas en reposo y en movimiento, las propiedades de los campos eléctricos y magnéticos, y su interacción según lo descrito por las ecuaciones de Maxwell. Abarca la electrostática, magnetostática, electrodinámica y la propagación de ondas electromagnéticas en el vacío y la materia.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.2.1 Electrostática: Campo Eléctrico y Ley de Gauss (Pág. 235)	1	1
⭐ 5.2.2 Potencial Eléctrico y Ecuaciones de Poisson/Laplace (Pág. 236)	1	1
⭐ 5.2.3 Energía Electrostática y Dieléctricos (Pág. 236)	1	1
⭐ 5.2.4 Magnetostática: Leyes de Biot-Savart y Ampère (Pág. 237)	1	1
⭐ 5.2.5 Propiedades Magnéticas de la Materia (Pág. 237)	1	1
⭐ 5.2.6 Electrodinámica: Ley de Faraday e Inducción (Pág. 238)	1	1
⭐ 5.2.7 Ecuaciones de Maxwell y Transformaciones de Norma (Pág. 238)	1	1
⭐ 5.2.8 Ondas Electromagnéticas y Propagación en Medios (Pág. 239)	1	1
⭐ 5.2.9 Teoría de la Radiación y Radiación Dipolar (Pág. 239)	1	1
⭐ 5.2.10 Relatividad Especial en Electromagnetismo (Pág. 240)	1	1
⭐ 5.2.11 Corriente Eléctrica, Circuitos y Leyes de Kirchhoff (Pág. 240)	1	1

Cuadro 5.2: Lista de KUs del área de Electromagnetismo.

5.2.1. ELM/Electrostática: Campo Eléctrico y Ley de Gauss (Core Tier1: 1 hr, Core Tier2: 1 hr)

Análisis de cargas estacionarias, la naturaleza vectorial del campo eléctrico y la aplicación de la Ley de Gauss a sistemas simétricos.

Temas:

Core

- Ley de Coulomb para cargas puntuales y el principio de superposición lineal
- El vector de campo eléctrico $\vec{E}$ y el cálculo para distribuciones continuas de carga
- La Ley de Gauss en forma integral y su relación con la carga encerrada
- Forma diferencial de la Ley de Gauss: $\nabla \cdot \vec{E} = \rho/\epsilon_0$
- Definición de flujo eléctrico a través de superficies abiertas y cerradas
- Aplicaciones de la Ley de Gauss a esferas y cortezas
- Aplicaciones de la Ley de Gauss a líneas infinitas y cilindros
- Aplicaciones de la Ley de Gauss a láminas infinitas y placas paralelas
- El Teorema de la Divergencia aplicado a vectores de campo electrostático
- Cargas puntuales representadas por la función delta de Dirac en 3D

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la fuerza electrostática neta sobre una carga a partir de una distribución discreta de cargas [Usar]
2. Aplicar la Ley de Gauss para hallar el campo eléctrico en sistemas con simetría esférica o cilíndrica [Usar]
3. Evaluar el flujo eléctrico a través de una superficie dada para un campo eléctrico no uniforme [Evaluar]
4. Determinar la densidad de carga volumétrica a partir de una expresión dada del campo eléctrico usando la forma diferencial de la Ley de Gauss [Usar]
5. Usar la función delta de Dirac para expresar formalmente la densidad de carga de una carga puntual en el origen [Usar]

5.2.2. ELM/Potencial Eléctrico y Ecuaciones de Poisson/Laplace (Core Tier1: 1 hr, Core Tier2: 1 hr)

La formulación del potencial escalar de la electrostática y la resolución de problemas de valores en la frontera.

Temas:

Core

- Definición del potencial escalar V y la relación $\vec{E} = -\nabla V$
- La naturaleza conservativa del campo electrostático: $\nabla \times \vec{E} = 0$
- Cálculo del potencial a partir de distribuciones de carga continuas de línea, superficie y volumen
- Propiedades de las superficies equipotenciales y su relación con las líneas de campo
- Derivación de la ecuación de Poisson $\nabla^2 V = -\rho/\epsilon_0$
- Ecuación de Laplace $\nabla^2 V = 0$ en regiones libres de carga
- El primer y segundo teorema de unicidad para soluciones electrostáticas
- El método de las imágenes para conductores planos y esféricos conectados a tierra
- Solución de la ecuación de Laplace mediante separación de variables en coordenadas cartesianas y esféricas
- Expansión del potencial en términos de momentos monopolares, dipolares y cuadrupolares

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía potencial de una configuración de cargas puntuales [Usar]
2. Derivar el vector de campo eléctrico a partir de una función de potencial escalar dada [Usar]
3. Resolver la ecuación de Laplace para una tubería rectangular con condiciones de frontera dadas [Evaluar]
4. Aplicar el método de las imágenes para encontrar la fuerza entre una carga puntual y un plano conductor infinito [Usar]
5. Analizar el término dominante de la expansión multipolar para una distribución de carga a grandes distancias [Evaluar]

5.2.3. ELM/Energía Electrostática y Dieléctricos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estudio del almacenamiento de energía en campos, capacitores y el comportamiento macroscópico de materiales dieléctricos.

Temas:

Core

- Trabajo requerido para ensamblar una distribución de cargas
- Densidad de energía en el campo eléctrico: $u = \frac{1}{2}\epsilon_0 E^2$
- Definición de capacitancia $C = Q/V$ y cálculo para geometrías estándar
- Polarización atómica y el vector de polarización $\vec{P}$
- Cargas ligadas superficiales y volumétricas en medios polarizados
- El vector de desplazamiento eléctrico $\vec{D}$ y la Ley de Gauss en la materia
- Dieléctricos lineales, susceptibilidad eléctrica χ_e y permitividad relativa ϵ_r
- Condiciones de frontera en la interfaz para los vectores $\vec{E}$ y $\vec{D}$

- Energía en sistemas dieléctricos: $W = \frac{1}{2} \int \vec{D} \cdot \vec{E} d\tau$

Electivo

- Descripción microscópica de los dieléctricos y la relación de Clausius-Mossotti

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la capacitancia de un capacitor de placas paralelas lleno de un dieléctrico lineal [Usar]
2. Evaluar la distribución de carga ligada para una esfera dieléctrica con polarización uniforme [Evaluar]
3. Resolver el campo eléctrico en ambos lados de una interfaz dieléctrica usando condiciones de frontera [Usar]
4. Analizar la fuerza ejercida sobre una placa dieléctrica al ser introducida en un capacitor [Evaluar]

5.2.4. ELM/Magnetostática: Leyes de Biot-Savart y Ampère (Core Tier1: 1 hr, Core Tier2: 1 hr)

La física de las corrientes eléctricas constantes, los campos magnéticos y las fuerzas sobre cargas en movimiento.

Temas:

Core

- Ley de la fuerza de Lorentz para cargas puntuales: $\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$
- Fuerza magnética sobre cables y lazos conductores de corriente
- Ley de Biot-Savart para el campo magnético de un elemento de corriente
- Ley de Ampère en forma integral para corrientes de alta simetría
- Forma diferencial de la Ley de Ampère: $\nabla \times \vec{B} = \mu_0 \vec{J}$
- La naturaleza solenoidal de $\vec{B}$: $\nabla \cdot \vec{B} = 0$
- El potencial vectorial magnético $\vec{A}$ y su relación $\vec{B} = \nabla \times \vec{A}$
- El calibre de Coulomb $\nabla \cdot \vec{A} = 0$ en magnetostática
- Momento dipolar magnético $\vec{m}$ de un lazo de corriente
- Expansión multipolar del potencial vectorial y el término dipolar

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el campo magnético de un cable recto infinito usando la Ley de Ampère [Usar]
2. Calcular el campo magnético en el eje de un lazo de corriente circular usando la Ley de Biot-Savart [Usar]
3. Verificar que un campo magnético dado satisface la condición solenoidal $\nabla \cdot \vec{B} = 0$ [Evaluar]
4. Determinar el potencial vectorial magnético para un campo magnético uniforme [Usar]
5. Analizar el torque y la fuerza sobre un dipolo magnético en un campo magnético no uniforme [Evaluar]

5.2.5. ELM/Propiedades Magnéticas de la Materia (Core Tier1: 1 hr, Core Tier2: 1 hr)

Magnetización de materiales, corrientes ligadas y clasificación de medios magnéticos.

Temas:

Core

- El vector de magnetización $\vec{M}$ como momento dipolar magnético por unidad de volumen
- Corrientes ligadas superficiales $\vec{K}_b$ y corrientes volumétricas $\vec{J}_b$
- La intensidad de campo magnético $\vec{H}$ y la Ley de Ampère en la materia
- Principios físicos del diamagnetismo y paramagnetismo
- Ferromagnetismo, dominios magnéticos y el ciclo de histéresis
- Condiciones de frontera en la interfaz para los vectores $\vec{B}$ y $\vec{H}$
- Medios magnéticos lineales y permeabilidad magnética μ

Electivo

- El potencial escalar magnético en regiones libres de corriente
- Introducción al diamagnetismo perfecto y el efecto Meissner

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Hallar la distribución de corriente ligada para un cilindro largo con magnetización longitudinal uniforme [Usar]
2. Clasificar materiales como diamagnéticos, paramagnéticos o ferromagnéticos según su susceptibilidad [Evaluar]
3. Resolver los campos $\vec{B}$ y $\vec{H}$ en una esfera magnética lineal colocada en un campo uniforme [Usar]
4. Analizar la pérdida de energía por ciclo en un material ferromagnético a partir de su curva de histéresis [Evaluar]

5.2.6. ELM/Electrodinámica: Ley de Faraday e Inducción (Core Tier1: 1 hr, Core Tier2: 1 hr)

Campos magnéticos dependientes del tiempo y la generación de fuerzas electromotrices.
Temas:
Core

- Ley de inducción de Faraday y Ley de Lenz para la dirección
- La regla del flujo para la fuerza electromotriz (FEM) en circuitos móviles
- La naturaleza no conservativa del campo eléctrico inducido: $\nabla \times \vec{E} = -\partial \vec{B}/\partial t$
- Inductancia mutua M entre dos circuitos
- Autoinductancia L y la fuerza contraelectromotriz en inductores
- Energía almacenada en el campo magnético: $W = \frac{1}{2\mu_0} \int B^2 d\tau$
- Formación de corrientes de Foucault y frenado magnético
- Formalismo de la matriz de inductancia para sistemas de múltiples circuitos
- La fórmula de Neumann para calcular la inductancia mutua

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la FEM inducida en un lazo giratorio dentro de un campo magnético constante [Usar]
2. Aplicar la Ley de Lenz para determinar la dirección de la corriente inducida en un escenario físico dado [Evaluar]
3. Calcular la energía magnética total almacenada en un solenoide largo [Usar]
4. Evaluar la inductancia mutua entre un cable largo y un lazo rectangular adyacente [Evaluar]

5.2.7. ELM/Ecuaciones de Maxwell y Transformaciones de Norma (Core Tier1: 1 hr, Core Tier2: 1 hr)

La unificación de la electricidad y el magnetismo y la estructura matemática de las ecuaciones de campo.
Temas:
Core

- Corrección de Maxwell a la Ley de Ampère y la corriente de desplazamiento $\vec{J}_d$
- El conjunto completo de las cuatro ecuaciones de Maxwell en el vacío y la materia
- Conservación de la carga y la ecuación de continuidad: $\nabla \cdot \vec{J} + \partial \rho/\partial t = 0$
- Formulación de las ecuaciones de Maxwell en términos de V y $\vec{A}$
- Concepto de invarianza de norma en electrodinámica
- La condición de calibre de Lorentz: $\nabla \cdot \vec{A} + \mu_0 \epsilon_0 \partial V/\partial t = 0$
- El teorema de Poynting y la conservación de la energía en los campos
- El vector de Poynting $\vec{S} = \vec{E} \times \vec{H}$ como densidad de flujo de energía
- Conservación del momento y el tensor de esfuerzos de Maxwell $\vec{T}$

Electivo

- Potenciales retardados y ecuaciones de Jefimenko

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la corriente de desplazamiento en el espacio de un capacitor en proceso de carga [Usar]

2. Enunciar las cuatro ecuaciones de Maxwell tanto en forma integral como diferencial [Familiarizarse]
3. Aplicar una transformación de norma a un conjunto de potenciales dado manteniendo los campos invariantes [Usar]
4. Analizar el flujo de energía y la disipación de potencia en una resistencia usando el vector de Poynting [Evaluar]
5. Calcular la presión de radiación ejercida por una onda plana sobre una superficie perfectamente reflectante [Usar]

5.2.8. ELM/Ondas Electromagnéticas y Propagación en Medios (Core Tier1: 1 hr, Core Tier2: 1 hr)

La propagación de perturbaciones electromagnéticas en el vacío, dieléctricos y conductores.

Temas:

Core

- Derivación de la ecuación de onda electromagnética en el vacío
- Soluciones de ondas planas monocromáticas y sus propiedades
- Estados de polarización: lineal, circular y elíptica
- Densidad de energía, intensidad y momento de las ondas planas
- Reflexión y refracción en incidencia normal sobre una interfaz
- Ecuaciones de Fresnel y ángulo de Brewster para incidencia oblicua
- Propagación de ondas en medios conductores y la profundidad de piel δ
- Dependencia de la permitividad con la frecuencia y la física de la dispersión
- Velocidad de fase frente a velocidad de grupo en medios dispersivos

Electivo

- Modos TE y TM en guías de onda rectangulares

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Derivar la relación de dispersión y la velocidad de la luz a partir de las ecuaciones de Maxwell [Usar]
2. Determinar el estado de polarización de una onda a partir de sus componentes vectoriales [Evaluar]
3. Aplicar las ecuaciones de Fresnel para encontrar el coeficiente de reflexión para las polarizaciones TE y TM [Usar]
4. Analizar cómo varía la profundidad de piel con la frecuencia para un conductor metálico dado [Evaluar]
5. Calcular el ángulo de Brewster para una interfaz entre aire y vidrio [Usar]

5.2.9. ELM/Teoría de la Radiación y Radiación Dipolar (Core Tier1: 1 hr, Core Tier2: 1 hr)

Generación de radiación electromagnética por cargas aceleradas y dipolos oscilantes.

Temas:

Core

- Radiación de un dipolo eléctrico oscilante
- Radiación de un dipolo magnético oscilante
- Concepto de resistencia de radiación en antenas
- La fórmula de Larmor para la potencia total radiada por una carga puntual acelerada

Electivo

- Los potenciales de Liénard-Wiechert para una carga puntual en movimiento
- Introducción a Bremsstrahlung y radiación sincrotrón

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la potencia total radiada por un dipolo eléctrico oscilante [Usar]
2. Analizar la distribución angular de la radiación de un dipolo eléctrico [Evaluar]

3. Aplicar la fórmula de Larmor para determinar la pérdida de potencia de un electrón en una órbita circular [Usar]

5.2.10. ELM/Relatividad Especial en Electromagnetismo (Core Tier1: 1 hr, Core Tier2: 1 hr)

La covarianza relativista de las ecuaciones de Maxwell y la transformación de campos.

Temas:

Core

- La densidad de cuadricorriente J^μ
- El tensor de campo electromagnético $F^{\mu\nu}$ y sus propiedades
- Transformación de Lorentz de los campos $\vec{E}$ y $\vec{B}$
- Invariantes relativistas del campo electromagnético

Electivo

- El tensor de campo dual y su papel en las ecuaciones de Maxwell

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Transformar los campos eléctricos y magnéticos de un sistema inercial a otro [Usar]
2. Verificar la covarianza de las ecuaciones de Maxwell usando la notación de cuadrivectores [Evaluar]
3. Explicar cómo un campo puramente eléctrico en un sistema puede aparecer como un campo magnético en otro [Familiarizarse]

5.2.11. ELM/Corriente Eléctrica, Circuitos y Leyes de Kirchhoff (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estudio del flujo de carga eléctrica, la resistencia, la potencia disipada y el análisis de circuitos de corriente continua mediante las leyes de Kirchhoff.

Temas:

Core

- Definición de corriente eléctrica, densidad de corriente y la ecuación de continuidad
- Ley de Ohm, resistividad y dependencia de la resistencia con la temperatura
- Potencia disipada en un resistor: efecto Joule y $P = I^2R$
- Ley de Kirchhoff de corrientes (KCL): conservación de la carga en un nodo
- Ley de Kirchhoff de voltajes (KVL): conservación de energía en una malla
- Análisis de circuitos en serie y paralelo: resistencias equivalentes
- Fuentes de fuerza electromotriz (FEM) y resistencia interna de baterías
- Circuitos RC: carga y descarga, constante de tiempo $\tau = RC$

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar las leyes de Kirchhoff para resolver circuitos de corriente continua con múltiples mallas y nodos [Usar]
2. Calcular la potencia disipada y la energía consumida en los elementos de un circuito resistivo [Evaluar]
3. Analizar el comportamiento transitorio de un circuito RC e interpretar su constante de tiempo [Usar]

5.3. Termodinámica y Mecánica Estadística (TSM)

La Termodinámica y la Mecánica Estadística cierran la brecha entre las propiedades microscópicas de las partículas individuales y los observables macroscópicos de la materia en conjunto. Esta área cubre las leyes de transferencia de energía, la entropía, los estados de equilibrio y los colectivos estadísticos utilizados para describir sistemas de muchos cuerpos en regímenes tanto clásicos como cuánticos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.3.1 Leyes de la Termodinámica y Funciones de Estado (Pág. 241)	1	1
⭐ 5.3.2 Teoría Cinética de los Gases (Pág. 241)	1	1
⭐ 5.3.3 Máquinas Térmicas, Entropía y la Segunda Ley (Pág. 242)	1	1
⭐ 5.3.4 Potenciales Termodinámicos y Relaciones de Maxwell (Pág. 242)	1	1
⭐ 5.3.5 Transiciones de Fase y Fenómenos Críticos (Pág. 242)	1	1
⭐ 5.3.6 Colectivos Microcanónico, Canónico y Gran Canónico (Pág. 243)	1	1
⭐ 5.3.7 Estadística Cuántica: Fermi-Dirac y Bose-Einstein (Pág. 243)	1	1
⭐ 5.3.8 Radiación de Cuerpo Negro y Fonones (Pág. 244)	1	1
⭐ 5.3.9 Mecánica Estadística de No-Equilibrio (Pág. 244)	1	1
⭐ 5.3.10 Aplicaciones a Sistemas Biológicos y Complejos (Pág. 244)	1	1

Cuadro 5.3: Lista de KUs del área de Termodinámica y Mecánica Estadística.

5.3.1. TSM/Leyes de la Termodinámica y Funciones de Estado (Core Tier1: 1 hr, Core Tier2: 1 hr)

Principios fundamentales que rigen la energía, el calor y el trabajo, y la definición matemática de las variables de estado.

Temas:

Core

- Ley Cero de la Termodinámica y la definición de temperatura
- Primera Ley de la Termodinámica: $dU = dQ - dW$ y la conservación de la energía interna
- Distinción entre funciones de estado y variables dependientes de la trayectoria
- Propiedades matemáticas de las diferenciales exactas e inexactas en termodinámica
- Energía interna de los gases ideales y su dependencia de la temperatura
- Definición de entalpía $H = U + PV$ y su significado físico
- Capacidades caloríficas a volumen constante C_V y a presión constante C_P
- Procesos adiabáticos y la relación $PV^\gamma = \text{constante}$
- Expansión de gases reales y el coeficiente de Joule-Thomson

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el trabajo realizado por un gas durante expansiones isotérmicas y adiabáticas [Usar]
2. Evaluar el cambio en la energía interna para un proceso cíclico usando la Primera Ley [Evaluar]
3. Aplicar la relación de Mayer para encontrar la diferencia entre C_P y C_V para un gas ideal [Usar]

5.3.2. TSM/Teoría Cinética de los Gases (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelo microscópico de los gases basado en el movimiento molecular y las colisiones.

Temas:

Core

- Derivación de la presión macroscópica a partir de la transferencia de momento molecular
- Distribución de velocidades de Maxwell-Boltzmann y su dependencia de la temperatura
- Teorema de equipartición y grados de libertad para gases monoatómicos y diatómicos
- Velocidad cuadrática media, media y más probable de las moléculas
- Camino libre medio λ y su relación con la sección eficaz molecular y la densidad
- Fenómenos de transporte: viscosidad, conductividad térmica y difusión en gases
- Efusión molecular y ley de Graham de la efusión
- Ecuación de estado de Van der Waals para gases reales: $(P + a/v^2)(v - b) = RT$

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la velocidad cuadrática media de las moléculas de nitrógeno a temperatura ambiente [Usar]
2. Interpretar el desplazamiento del pico de la distribución de Maxwell-Boltzmann al aumentar la temperatura [Evaluar]

3. Calcular el camino libre medio de las partículas en una cámara de alto vacío [Usar]

5.3.3. TSM/Máquinas Térmicas, Entropía y la Segunda Ley (Core Tier1: 1 hr, Core Tier2: 1 hr)

Análisis de la eficiencia térmica y la definición estadística/termodinámica de la entropía.

Temas:

Core

- Enunciados de Kelvin-Planck y Clausius de la Segunda Ley
- El ciclo de Carnot y la eficiencia teórica máxima $\eta = 1 - T_c/T_h$
- Definición de la escala de temperatura termodinámica absoluta
- Desigualdad de Clausius y la definición de entropía $dS \geq dQ/T$
- Principio del aumento de la entropía en sistemas aislados
- Análisis de ciclos termodinámicos usando diagramas Temperatura-Entropía $(T - S)$
- Fórmula de la entropía de Boltzmann: $S = k_B \ln \Omega$
- Entropía de mezcla y la paradoja de Gibbs
- Tercera Ley de la Termodinámica y la inalcanzabilidad del cero absoluto

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la eficiencia máxima de una máquina térmica que opera entre dos temperaturas dadas [Usar]
2. Evaluar el cambio de entropía para un proceso irreversible como la mezcla de dos líquidos [Evaluar]
3. Calcular la entropía de un sistema de diez espines que no interactúan [Usar]

5.3.4. TSM/Potenciales Termodinámicos y Relaciones de Maxwell (Core Tier1: 1 hr, Core Tier2: 1 hr)

Las transformaciones de Legendre y las relaciones de simetría entre variables termodinámicas.

Temas:

Core

- Base matemática de las transformaciones de Legendre para potenciales termodinámicos
- Energía libre de Gibbs $G = H - TS$ y su papel en el equilibrio químico
- Energía libre de Helmholtz $A = U - TS$ y su relación con el trabajo máximo
- Derivación de las cuatro relaciones primarias de Maxwell a partir de diferenciales exactas
- La ecuación de Gibbs-Duhem y sus restricciones sobre las variables intensivas
- Uso de cuadros termodinámicos para recordar relaciones entre potenciales
- Definición de potencial químico μ y equilibrio de intercambio de partículas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar una relación de Maxwell para simplificar el cálculo de cambios de energía interna [Usar]
2. Determinar el potencial termodinámico apropiado a minimizar para un sistema a P y T constantes [Evaluar]
3. Derivar una relación de Maxwell a partir del diferencial de la energía libre de Helmholtz [Usar]

5.3.5. TSM/Transiciones de Fase y Fenómenos Críticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estudio de los cambios de estado de la materia y el comportamiento cerca de los puntos críticos.

Temas:

Core

- Diagramas de fase $P - T$ y $P - V$ para sistemas de un solo componente
- La ecuación de Clapeyron para las pendientes de los límites de fase
- Ecuación de Clausius-Clapeyron para transiciones líquido-vapor
- Clasificación de Ehrenfest de las transiciones de fase de primer y segundo orden
- Calor latente de fusión, vaporización y sublimación

- Temperatura crítica, presión y opalescencia

Electivo

- Introducción a la teoría de Landau de las transiciones de fase y parámetros de orden

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Estimar el cambio en el punto de ebullición del agua a altas altitudes usando la ecuación de Clausius-Clapeyron [Usar]
2. Clasificar una transición dada como de primer o segundo orden basada en las discontinuidades de la capacidad calorífica [Evaluar]
3. Determinar la presión y temperatura críticas a partir de la ecuación de van der Waals [Usar]

5.3.6. TSM/Colectivos Microcanónico, Canónico y Gran Canónico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Marcos estadísticos para calcular propiedades macroscópicas a partir de probabilidades de microestados.

Temas:
Core

- Espacio de fase, celdas y el conteo de microestados
- El Colectivo Microcanónico (N, V, E): Postulado de igual probabilidad a priori
- El Colectivo Canónico (N, V, T) y el factor de Boltzmann $e^{-\beta E}$
- La función de partición Z y su relación con la energía libre de Helmholtz $A = -k_B T \ln Z$
- El Colectivo Gran Canónico (μ, V, T) y el factor de Gibbs
- Fluctuaciones de energía y número de partículas en los colectivos
- Fórmula de la entropía de Gibbs y su relación con la teoría de la información

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la función de partición para un sistema de dos niveles y derivar su energía media [Usar]
2. Evaluar la presión y la entropía de un sistema partiendo de su función de partición [Evaluar]
3. Calcular la gran función de partición para un sistema de partículas que no interactúan [Usar]

5.3.7. TSM/Estadística Cuántica: Fermi-Dirac y Bose-Einstein (Core Tier1: 1 hr, Core Tier2: 1 hr)

Comportamiento estadístico de partículas idénticas gobernadas por el principio de exclusión de Pauli o la simetría de Bose.

Temas:
Core

- Partículas distinguibles vs. indistinguibles y simetría de la función de onda
- Distribución de Fermi-Dirac para fermiones y el nivel de Fermi ϵ_F
- Distribución de Bose-Einstein para bosones
- Densidad de estados $g(\epsilon)$ en el espacio de momento en TresD
- Propiedades del gas de Fermi degenerado y presión de degeneración electrónica
- Fenómeno de la Condensación de Bose-Einstein (BEC) a bajas temperaturas
- El límite de Maxwell-Boltzmann de las estadísticas cuánticas a altas temperaturas

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la energía de Fermi para una densidad dada de electrones libres en un metal [Usar]
2. Analizar las condiciones de temperatura y densidad requeridas para lograr la condensación de Bose-Einstein [Evaluar]
3. Determinar la presión de un gas de Fermi completamente degenerado a $T = 0$ [Usar]

5.3.8. TSM/Radiación de Cuerpo Negro y Fonones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aplicación de las estadísticas cuánticas a los gases de fotones y vibraciones de red.

Temas:

Core

- Derivación de la ley de Planck para la densidad de energía espectral de la radiación de cuerpo negro
- Ley de Stefan-Boltzmann para la potencia total radiada: $P \propto T^4$
- Ley de desplazamiento de Wien para la longitud de onda máxima de emisión
- Modelo de Einstein de calor específico basado en osciladores independientes
- Modelo de Debye de los sólidos y la ley T^3 para la capacidad calorífica a baja temperatura
- Los fonones como vibraciones de red cuantizadas y su densidad de estados

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la temperatura superficial de una estrella dada su longitud de onda de emisión máxima [Usar]
2. Comparar los modelos de Einstein y Debye al explicar el calor específico experimental de los sólidos [Evaluar]
3. Estimar la frecuencia de Debye a partir de la velocidad del sonido en un sólido [Usar]

5.3.9. TSM/Mecánica Estadística de No-Equilibrio (Core Tier1: 1 hr, Core Tier2: 1 hr)

Estudio de sistemas que se aproximan al equilibrio y coeficientes de transporte.

Temas:

Core

- Teoría de Einstein del movimiento browniano y el coeficiente de difusión
- Ecuación de Langevin para dinámica estocástica
- Introducción a la ecuación de Fokker-Planck para la evolución de la densidad de probabilidad
- El Teorema de Fluctuación-Disipación y su significado

Electivo

- La ecuación de transporte de Boltzmann y la aproximación del tiempo de relajación

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la constante de difusión para una partícula esférica en un fluido viscoso [Usar]
2. Analizar el efecto de las fuerzas aleatorias en el desplazamiento cuadrático medio de una partícula [Evaluar]
3. Aplicar la relación de Einstein para obtener la movilidad a partir del coeficiente de difusión [Usar]

5.3.10. TSM/Aplicaciones a Sistemas Biológicos y Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelado termodinámico de macromoléculas, membranas y sistemas ricos en información.

Temas:

Core

- Mecánica estadística de cadenas poliméricas: Modelos de unión libre y tipo gusano
- Termodinámica del plegamiento de proteínas y desnaturalización del ADN
- Mecánica estadística de canales iónicos y potenciales de membrana

Electivo

- Entropía de Shannon y su conexión con la entropía física
- Introducción a la autoorganización en sistemas alejados del equilibrio

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Modelar la curva fuerza-extensión de una molécula de ADN usando estadística de polímeros [Usar]
2. Evaluar la estabilidad del plegamiento de una proteína basada en el paisaje de energía libre [Evaluar]
3. Estimar la tensión superficial de una bicapa lipídica a partir de sus módulos elásticos [Usar]

5.4. Mecánica Cuántica (QME)

La Mecánica Cuántica proporciona el marco fundamental para comprender el comportamiento de la materia y la energía a la escala de átomos y partículas subatómicas. Sustituye el determinismo clásico por funciones de onda probabilísticas, operadores en espacios de Hilbert y los principios de superposición y entrelazamiento.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.4.1 Orígenes de la Teoría Cuántica y Dualidad Onda-Partícula (Pág. 245)	1	1
⭐ 5.4.2 La Ecuación de Schrödinger y Funciones de Onda (Pág. 245)	1	1
⭐ 5.4.3 Formalismo: Espacio de Hilbert y Notación de Dirac (Pág. 246)	1	1
⭐ 5.4.4 Potenciales Unidimensionales (Pozos, Barreras, Tunelamiento) (Pág. 246)	1	1
⭐ 5.4.5 El Oscilador Armónico Cuántico (Pág. 247)	1	1
⭐ 5.4.6 Momento Angular y Física del Espín (Pág. 247)	1	1
⭐ 5.4.7 El átomo de Hidrógeno y Potenciales Centrales (Pág. 247)	1	1
⭐ 5.4.8 Teoría de Perturbaciones Independiente y Dependiente del Tiempo (Pág. 248)	1	1
⭐ 5.4.9 Partículas Idénticas y Simetría de Intercambio (Pág. 248)	1	1
⭐ 5.4.10 Teoría de la Dispersión y Secciones Eficaces (Pág. 248)	1	1

Cuadro 5.4: Lista de KUs del área de Mecánica Cuántica.

5.4.1. QME/Orígenes de la Teoría Cuántica y Dualidad Onda-Partícula (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fundamentos históricos y la transición de la física clásica a la cuántica.

Temas:

Core

- Radiación de cuerpo negro y la cuantización de la energía de Planck
- El efecto fotoeléctrico y la hipótesis del fotón de Einstein
- Dispersión Compton y el comportamiento corpuscular de la luz
- El modelo de Bohr y la cuantización de las órbitas atómicas
- Hipótesis de De Broglie de las ondas de materia y la relación $\lambda = h/p$
- Experimento de Davisson-Germer y la difracción de electrones
- Principio de Incertidumbre de Heisenberg: $\Delta x \Delta p \geq \hbar/2$
- Principio de complementariedad de Bohr y la naturaleza de la medición

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía y el momento de un fotón dada su longitud de onda [Usar]
2. Evaluar la incertidumbre mínima en la posición para una partícula con una dispersión de momento conocida [Evaluar]
3. Explicar los resultados del experimento de la doble rendija tanto para la luz como para los electrones [Familiarizarse]

5.4.2. QME/La Ecuación de Schrödinger y Funciones de Onda (Core Tier1: 1 hr, Core Tier2: 1 hr)

La ecuación fundamental de movimiento y la interpretación estadística de la función de onda.

Temas:

Core

- La ecuación de Schrödinger dependiente del tiempo: $i\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi$
- La ecuación de Schrödinger independiente del tiempo y estados estacionarios de energía
- Interpretación estadística de Born de la función de onda y densidad de probabilidad
- La condición de normalización $\int |\Psi|^2 dx = 1$
- Cálculo de valores de expectación para posición y momento
- Corriente de probabilidad J y la ecuación de continuidad
- Teorema de Ehrenfest que relaciona los promedios cuánticos con las leyes clásicas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Normalizar una función de onda matemática dada para una partícula en el espacio [Usar]
2. Calcular el valor de expectación de la energía cinética para un paquete de ondas gaussiano [Usar]
3. Verificar que la probabilidad total se conserva en el tiempo usando la ecuación de Schrödinger [Evaluar]

5.4.3. QME/Formalismo: Espacio de Hilbert y Notación de Dirac (Core Tier1: 1 hr, Core Tier2: 1 hr)

La estructura matemática rigurosa de la mecánica cuántica usando álgebra lineal.

Temas:

Core

- Vectores de estado como elementos de un espacio de Hilbert complejo
- Notación de Dirac: Bras, kets y productos internos
- Operadores hermíticos y su papel como observables físicos
- Ecuaciones de autovalores y la ortonormalidad de los autoestados
- La relación de completitud y operadores de proyección
- Conmutadores y el principio de incertidumbre generalizado
- Representación matricial de operadores en una base discreta
- Transformaciones unitarias y cambio de base en el espacio de Hilbert

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar la notación de Dirac para calcular el producto interno y las amplitudes de probabilidad [Usar]
2. Verificar si un operador dado es hermítico examinando sus elementos de matriz [Evaluar]
3. Resolver el problema de autovalores para una matriz hamiltoniana de Dos-por-Dos [Usar]

5.4.4. QME/Potenciales Unidimensionales (Pozos, Barreras, Tunelamiento) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Modelos solubles en UnaD que demuestran efectos cuánticos clave como estados ligados y tunelamiento.

Temas:

Core

- El pozo cuadrado infinito: niveles de energía y funciones de onda
- El pozo cuadrado finito y la existencia de estados ligados
- Dispersión en un escalón de potencial: coeficientes de reflexión y transmisión
- Tunelamiento cuántico a través de una barrera de potencial rectangular
- Estados ligados y dispersión en un potencial de delta de Dirac
- La solución de partícula libre y la dispersión de paquetes de ondas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía del estado fundamental para un electrón en un pozo cuadrado infinito [Usar]
2. Evaluar la probabilidad de transmisión para una partícula que tunela a través de una barrera delgada [Evaluar]
3. Esbozar el comportamiento cualitativo de las funciones de onda para varios perfiles de potencial [Usar]

5.4.5. QME/El Oscilador Armónico Cuántico (Core Tier1: 1 hr, Core Tier2: 1 hr)

El modelo más importante de la física, resuelto mediante métodos algebraicos y analíticos.

Temas:

Core

- El hamiltoniano del oscilador armónico y su límite clásico
- Método algebraico: operadores de escala (creación y aniquilación)
- Niveles de energía cuantizados: $E_n = (n + 1/2)\hbar\omega$
- Significado físico de la energía del punto cero
- Método analítico: solución por serie de potencias y polinomios de Hermite

Electivo

- Introducción a los estados coherentes y su evolución temporal

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar operadores de escala para encontrar el primer estado excitado a partir del estado fundamental [Usar]
2. Analizar la probabilidad de encontrar una partícula osciladora en la región clásicamente prohibida [Evaluar]
3. Calcular la energía de punto cero de un oscilador armónico con una constante de resorte dada [Usar]

5.4.6. QME/Momento Angular y Física del Espín (Core Tier1: 1 hr, Core Tier2: 1 hr)

Simetría rotacional y el momento angular intrínseco de las partículas.

Temas:

Core

- Operadores de momento angular orbital y relaciones de conmutación
- Autofunciones del momento angular: armónicos esféricos Y_l^m
- Sistemas de espín un medio y el experimento de Stern-Gerlach
- álgebra de las matrices de espín de Pauli
- Adición de momento angular y coeficientes de Clebsch-Gordan
- Precesión del espín en un campo magnético uniforme

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el valor de expectación de S_z para un estado de espinor dado [Usar]
2. Evaluar los posibles valores del momento angular total para dos espines que interactúan [Evaluar]
3. Aplicar el álgebra de matrices de Pauli para calcular las relaciones de conmutación de los operadores de espín [Usar]

5.4.7. QME/El átomo de Hidrógeno y Potenciales Centrales (Core Tier1: 1 hr, Core Tier2: 1 hr)

La solución de la mecánica cuántica para el átomo más simple y el movimiento en simetría esférica.

Temas:

Core

- Reducción de la ecuación de Schrödinger en 3D a la ecuación radial
- Término centrífugo y el potencial efectivo
- Espectro de energía del átomo de hidrógeno y la constante de Rydberg
- Números cuánticos principal, azimutal y magnético
- Estructura de las funciones de onda radiales y polinomios de Laguerre
- Estructura de capas atómicas y las formas de los orbitales s, p, d

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar el radio más probable para un electrón en el estado fundamental del hidrógeno [Usar]

2. Analizar la degeneración de los niveles de energía en el átomo de hidrógeno despreciando el espín [Evaluar]
3. Calcular la constante de Rydberg a partir de la energía del estado fundamental del hidrógeno [Usar]

5.4.8. QME/Teoría de Perturbaciones Independiente y Dependiente del Tiempo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos de aproximación para sistemas que no pueden resolverse exactamente.

Temas:

Core

- Correcciones de energía y de estado de primer orden para sistemas no degenerados
- Correcciones de energía de segundo orden y su significado
- Teoría de perturbaciones degenerada y la eliminación de degeneraciones
- Estructura fina del hidrógeno: correcciones relativistas y de espín-órbita
- El efecto Zeeman: átomos en campos magnéticos externos
- Teoría de perturbaciones dependiente del tiempo y amplitudes de transición
- Regla de Oro de Fermi para tasas de transición

Electivo

- El principio variacional para estimar energías del estado fundamental
- La aproximación WKB para potenciales que varían lentamente

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el desplazamiento de energía de primer orden para un oscilador con una pequeña perturbación anharmónica [Usar]
2. Evaluar la tasa de transición entre dos estados bajo una perturbación armónica usando la Regla de Oro de Fermi [Evaluar]
3. Aplicar el método variacional para estimar la energía del estado fundamental de un pozo de potencial delta [Usar]

5.4.9. QME/Partículas Idénticas y Simetría de Intercambio (Core Tier1: 1 hr, Core Tier2: 1 hr)

La física de los sistemas que contienen múltiples partículas indistinguibles.

Temas:

Core

- El operador de intercambio y la simetría/antisimetría de los estados
- Distinción entre bosones y fermiones
- El principio de exclusión de Pauli y determinantes de Slater
- El átomo de helio y el papel de la energía de intercambio
- Estados de espín singlete y triplete para sistemas de dos electrones

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Enunciar el principio de exclusión de Pauli y su impacto en la tabla periódica [Familiarizarse]
2. Construir un determinante de Slater para un sistema de tres fermiones que no interactúan [Usar]
3. Determinar el signo de la energía de intercambio para dos electrones en un estado triplete [Evaluar]

5.4.10. QME/Teoría de la Dispersión y Secciones Eficaces (Core Tier1: 1 hr, Core Tier2: 1 hr)

Formalismo para describir colisiones e interacciones entre partículas.

Temas:

Core

- Definición de las secciones eficaces diferencial y total
- La amplitud de dispersión $f(\theta)$ y su relación con la sección eficaz

- La primera aproximación de Born para potenciales débiles
- Análisis de ondas parciales y desplazamientos de fase

Electivo

- El teorema óptico que relaciona la dispersión hacia adelante con la sección eficaz total

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar la aproximación de Born para encontrar la amplitud de dispersión para un potencial de Yukawa [Usar]
2. Analizar el efecto de un potencial de esfera dura en el desplazamiento de fase de dispersión a baja energía [Evaluar]
3. Calcular la sección eficaz total a partir de la expansión en ondas parciales [Usar]

5.5. Relatividad y Cosmología (RCO)

La Relatividad y la Cosmología exploran la naturaleza fundamental del espacio, el tiempo y la gravedad, desde el comportamiento local de partículas a altas velocidades hasta la evolución a gran escala del universo. Esta área cubre la transición de la relatividad especial a la general y los modelos matemáticos que describen la historia y composición del cosmos.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.5.1 Relatividad Especial: Transformaciones de Lorentz y Cuadrivectores (Pág. 249)	1	1
⭐ 5.5.2 Dinámica Relativista y Energía-Momento (Pág. 250)	1	1
⭐ 5.5.3 Fundamentos de la Relatividad General: Principio de Equivalencia (Pág. 250)	1	1
⭐ 5.5.4 Curvatura del Espacio-Tiempo y Ecuaciones de Campo de Einstein (Pág. 250)	1	1
⭐ 5.5.5 Geometría de Schwarzschild y Agujeros Negros (Pág. 251)	1	1
⭐ 5.5.6 Ondas Gravitacionales (Pág. 251)	1	1
⭐ 5.5.7 Teoría del Big Bang y Ecuaciones de Friedmann (Pág. 251)	1	1
⭐ 5.5.8 Fondo Cósmico de Microondas (CMB) (Pág. 252)	1	1
⭐ 5.5.9 Materia Oscura y Energía Oscura (Pág. 252)	1	1
⭐ 5.5.10 Inflación del Universo Temprano y Nucleosíntesis (Pág. 252)	1	1

Cuadro 5.5: Lista de KUs del área de Relatividad y Cosmología.

5.5.1. RCO/Relatividad Especial: Transformaciones de Lorentz y Cuadrivectores (Core Tier1: 1 hr, Core Tier2: 1 hr)

El marco matemático del espacio-tiempo plano y la unificación del espacio y el tiempo.

Temas:

Core

- Ecuaciones de transformación de Lorentz para impulsos de coordenadas y velocidad
- El intervalo invariante ds^2 y la métrica de Minkowski
- Dilatación del tiempo, contracción de la longitud y tiempo propio τ
- Conos de luz, líneas de universo y estructura causal del espacio-tiempo
- Definición general de cuadrivectores y el producto escalar
- Introducción a los tensores y notación de índices en el espacio de Minkowski

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el tiempo transcurrido para un gemelo que viaja a velocidades relativistas [Usar]
2. Verificar la invarianza del intervalo espacio-temporal bajo un impulso de Lorentz [Evaluar]
3. Calcular el factor de Lorentz para una partícula que se mueve a 0.99c [Usar]

5.5.2. RCO/Dinámica Relativista y Energía-Momento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Leyes de conservación y física de partículas dentro del régimen relativista.

Temas:

Core

- Definición del cuadrivector energía-momento P^μ
- Energía de masa en reposo y la relación $E = \gamma mc^2$
- La relación de capa de masa: $E^2 - (pc)^2 = (mc^2)^2$
- Dinámica de colisiones de partículas y energías umbral de creación
- Fuerza de Minkowski y la versión relativista de la Segunda Ley de Newton

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el momento de una partícula dada su energía total y masa en reposo [Usar]
2. Analizar un proceso de desintegración de partículas usando la conservación del cuadrimomento [Evaluar]
3. Calcular la energía umbral para la producción de piones en colisiones protón-protón [Usar]

5.5.3. RCO/Fundamentos de la Relatividad General: Principio de Equivalencia (Core Tier1: 1 hr, Core Tier2: 1 hr)

El cambio conceptual de la gravedad como fuerza a la gravedad como geometría.

Temas:

Core

- El Principio de Equivalencia Débil y la universalidad de la caída libre
- Principio de Equivalencia de Einstein: física local en marcos de caída libre
- Derivación del corrimiento al rojo gravitacional a partir del principio de equivalencia
- El Principio de Mach y su influencia en el desarrollo de la RG

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar la equivalencia entre un campo gravitatorio y un marco acelerado [Familiarizarse]
2. Calcular el corrimiento gravitacional al rojo de la luz que escapa de una estrella masiva [Usar]
3. Analizar las consecuencias del principio de equivalencia para la deflexión de la luz por la gravedad [Evaluar]

5.5.4. RCO/Curvatura del Espacio-Tiempo y Ecuaciones de Campo de Einstein (Core Tier1: 1 hr, Core Tier2: 1 hr)

El núcleo matemático de la RG: tensores, curvatura y las ecuaciones de campo.

Temas:

Core

- El tensor métrico $g_{\mu\nu}$ como descripción fundamental de la geometría
- Símbolos de Christoffel y la derivada covariante
- El tensor de curvatura de Riemann y las fuerzas de marea
- Tensor de Ricci, escalar de Ricci y el tensor de Einstein $G_{\mu\nu}$
- Las Ecuaciones de Campo de Einstein (EFE): $G_{\mu\nu} = \kappa T_{\mu\nu}$
- El tensor de energía-momento para fluidos perfectos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular los símbolos de Christoffel no nulos para una métrica simple dada [Usar]
2. Interpretar la relación entre la densidad de materia-energía y la curvatura del espacio-tiempo [Evaluar]
3. Calcular el escalar de Ricci para la métrica de Schwarzschild en una coordenada radial dada [Usar]

5.5.5. RCO/Geometría de Schwarzschild y Agujeros Negros (Core Tier1: 1 hr, Core Tier2: 1 hr)

Soluciones de vacío de las EFE y la física de la gravedad extrema.

Temas:

Core

- La métrica de Schwarzschild para una masa estática con simetría esférica
- El radio de Schwarzschild R_s y el horizonte de sucesos
- Órbitas de luz y partículas en la geometría de Schwarzschild
- Dilatación del tiempo gravitacional cerca de un cuerpo masivo
- Singularidades de coordenadas frente a singularidades físicas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el radio de Schwarzschild para el Sol y la Tierra [Usar]
2. Analizar la deflexión de la luz por un objeto masivo en el límite de campo débil [Evaluar]
3. Calcular el período orbital en la última órbita circular estable (ISCO) de un agujero negro [Usar]

5.5.6. RCO/Ondas Gravitacionales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Ondulaciones propagadas en el tejido del espacio-tiempo.

Temas:

Core

- Gravedad linealizada y la ecuación de onda para perturbaciones del espacio-tiempo
- Modos de polarización Plus (+) y Cruz (×)
- La fórmula del cuadrupolo para la potencia de la radiación gravitacional
- Principios de detección mediante interferometría láser (LIGO/VIRGO)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Estimar la amplitud adimensional de las ondas gravitacionales de un púlsar binario [Usar]
2. Calcular la sensibilidad de deformación necesaria para detectar una señal de ondas gravitacionales dada [Usar]
3. Analizar la evolución en frecuencia de la coalescencia de un sistema de agujeros negros binarios durante los últimos segundos [Evaluar]

5.5.7. RCO/Teoría del Big Bang y Ecuaciones de Friedmann (Core Tier1: 1 hr, Core Tier2: 1 hr)

El modelo cosmológico estándar y la expansión del universo.

Temas:

Core

- El Principio Cosmológico: Isotropía y Homogeneidad
- La métrica de Friedmann-Lemaître-Robertson-Walker (FLRW)
- La Ley de Hubble y la constante de Hubble H_0
- Las ecuaciones de Friedmann para el factor de escala $a(t)$
- Densidad crítica y los parámetros de densidad $\Omega_m, \Omega_\Lambda, \Omega_k$
- Relación entre el corrimiento al rojo z y el factor de escala a

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar la edad aproximada de un universo de flujo de Hubble dado H_0 [Usar]
2. Analizar el destino del universo basado en el parámetro de densidad total [Evaluar]
3. Calcular la densidad crítica del universo para una constante de Hubble dada [Usar]

5.5.8. RCO/Fondo Cósmico de Microondas (CMB) (Core Tier1: 1 hr, Core Tier2: 1 hr)

La radiación relicta del universo temprano y caliente.

Temas:

Core

- época de recombinación y desacoplamiento de fotones
- Espectro de cuerpo negro del CMB y temperatura actual ($T \approx 2{,}73$ K)
- Anisotropías de temperatura y semillas de la estructura cósmica
- El espectro de potencia angular y los picos acústicos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir el significado de la superficie de última dispersión [Familiarizarse]
2. Estimar la densidad numérica de fotones del fondo cósmico de microondas en la actualidad [Usar]
3. Analizar cómo la posición del primer pico acústico restringe la geometría del universo [Evaluar]

5.5.9. RCO/Materia Oscura y Energía Oscura (Core Tier1: 1 hr, Core Tier2: 1 hr)

Los misteriosos componentes que dominan la densidad de energía del universo.

Temas:

Core

- Evidencia de materia oscura a partir de las curvas de rotación galáctica
- La constante cosmológica Λ como energía oscura
- Lentes gravitacionales como herramienta para mapear la masa de materia oscura
- Evidencia de expansión acelerada a partir de supernovas de tipo Ia
- El modelo estándar de cosmología ΛCDM

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Contrastar los roles de la materia oscura y la energía oscura en la evolución cósmica [Familiarizarse]
2. Evaluar la evidencia de materia oscura a partir del lensado gravitacional y las curvas de rotación [Evaluar]
3. Calcular Ω_m y Ω_Λ a partir de un conjunto dado de módulos de distancia de supernovas [Usar]

5.5.10. RCO/Inflación del Universo Temprano y Nucleosíntesis (Core Tier1: 1 hr, Core Tier2: 1 hr)

Procesos de alta energía en los primeros minutos y fracciones de segundo.

Temas:

Core

- Teoría inflacionaria: solución a los problemas del horizonte y de planitud
- Nucleosíntesis del Big Bang: origen de los elementos ligeros (H, He, Li)
- Bariogénesis y la asimetría materia-antimateria

Electivo

- Ruptura de simetría y transiciones de fase en el universo temprano

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Evaluar la abundancia predicha de helio-4 a partir de la teoría BBN [Evaluar]
2. Calcular el tamaño de una región causalmente conectada en la recombinación y compararlo con la homogeneidad observada del fondo cósmico de microondas [Usar]
3. Explicar por qué el problema de la planitud motiva una época inflacionaria [Familiarizarse]

5.6. Física de la Materia Condensada (SSP)

La Física de la Materia Condensada investiga las propiedades macroscópicas y microscópicas de la materia en sus fases sólida y líquida. Explora cómo las interacciones colectivas entre vastos números de átomos dan lugar a fenómenos emergentes como la superconductividad, el magnetismo y las fases topológicas.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.6.1 Estructuras Cristalinas y Redes Recíprocas (Pág. 253)	1	1
⭐ 5.6.2 Difracción de Rayos X y Dinámica de Red (Pág. 253)	1	1
⭐ 5.6.3 Modelo de Electrones Libres y Teoría de Bandas de Sólidos (Pág. 254)	1	1
⭐ 5.6.4 Semiconductores, Aislantes y Metales (Pág. 254)	1	1
⭐ 5.6.5 Propiedades Magnéticas: Para-, Dia- y Ferromagnetismo (Pág. 254)	1	1
⭐ 5.6.6 Superconductividad: Teoría BCS y Alta Tc (Pág. 255)	1	1
⭐ 5.6.7 Física de la Materia Blanda y Polímeros (Pág. 255)	1	1
⭐ 5.6.8 Nanoestructuras y Sistemas de Baja Dimensionalidad (Pág. 255)	1	1
⭐ 5.6.9 Espintrónica y Aislantes Topológicos (Pág. 256)	1	1
⭐ 5.6.10 Sistemas de Electrones Fuertemente Correlacionados (Pág. 256)	1	1

Cuadro 5.6: Lista de KUs del área de Física de la Materia Condensada.

5.6.1. SSP/Estructuras Cristalinas y Redes Recíprocas (Core Tier1: 1 hr, Core Tier2: 1 hr)

La descripción matemática de los arreglos atómicos periódicos en el espacio real y de momentos.

Temas:

Core

- Las 14 redes de Bravais y simetrías de grupos puntuales
- Celdas unitarias primitivas, celdas de Wigner-Seitz y la base de átomos
- Índices de Miller para planos y direcciones cristalinas
- Construcción de la red recíproca: $\vec{b}_i = 2\pi \frac{\vec{a}_j \times \vec{a}_k}{\vec{a}_i \cdot (\vec{a}_j \times \vec{a}_k)}$
- La primera zona de Brillouin y su significado físico en el espacio k
- Fracciones de empaquetamiento y números de coordinación para SC, BCC, FCC y HCP

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar los vectores de la red recíproca para un cristal cúbico centrado en las caras (FCC) [Usar]
2. Calcular el factor de empaquetamiento atómico para una estructura de diamante [Evaluar]
3. Identificar las operaciones de simetría rotacional y traslacional en un mosaico cristalino 2D [Usar]

5.6.2. SSP/Difracción de Rayos X y Dinámica de Red (Core Tier1: 1 hr, Core Tier2: 1 hr)

Sondeo de estructuras cristalinas y cuantización de vibraciones colectivas (fonones).

Temas:

Core

- Ley de Bragg de difracción: $2d \sin\theta = n\lambda$
- Condición de Laue para interferencia constructiva: $\Delta\vec{k} = \vec{G}$
- El factor de estructura geométrico S_G y ausencias sistemáticas en la difracción
- Relaciones de dispersión de fonones para cadenas monoatómicas y diatómicas
- Distinción física entre ramas de fonones acústicas y ópticas
- Dispersión inelástica de neutrones como herramienta para medir espectros de fonones

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el ángulo de difracción para el plano (111) del silicio usando rayos X de longitud de onda 1.54 Å [Usar]
2. Analizar la aparición de una brecha de frecuencia en una curva de dispersión de red diatómica [Evaluar]
3. Calcular el factor de estructura para un cristal BCC e identificar reflexiones prohibidas [Usar]

5.6.3. SSP/Modelo de Electrones Libres y Teoría de Bandas de Sólidos (Core Tier1: 1 hr, Core Tier2: 1 hr)

El comportamiento de los electrones en potenciales periódicos y el origen de las bandas de energía.

Temas:

Core

- El modelo clásico de Drude: conductividad, velocidad de deriva y efecto Hall
- El modelo de Sommerfeld: energía de Fermi, superficie de Fermi y densidad de estados
- Teorema de Bloch: funciones de onda en un potencial periódico $\psi(\vec{r}) = u(\vec{r})e^{i\vec{k}\cdot\vec{r}}$
- El modelo de Kronig-Penney y la emergencia de brechas de energía
- La aproximación del electrón casi libre y perturbación en los bordes de zona
- La aproximación de enlace fuerte para orbitales atómicos localizados

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía de Fermi y la temperatura de Fermi para el cobre [Usar]
2. Evaluar la masa efectiva de un electrón a partir de una relación de dispersión $E(k)$ dada [Evaluar]
3. Calcular la densidad de estados para un gas de electrones libres en 3D [Usar]

5.6.4. SSP/Semiconductores, Aislantes y Metales (Core Tier1: 1 hr, Core Tier2: 1 hr)

Clasificación de sólidos basada en el transporte electrónico y el dopaje.

Temas:

Core

- Semiconductores intrínsecos: concentración de portadores y ley de acción de masas
- Semiconductores extrínsecos: dopaje tipo n y tipo p, niveles de donantes/aceptores
- Física de la unión p-n: región de agotamiento y potencial de contacto
- Dependencia de la movilidad de portadores y conductividad con la temperatura
- Propiedades ópticas: semiconductores de brecha directa vs. indirecta

Electivo

- Principios de operación de MOSFETs y celdas solares

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar la concentración de portadores del silicio a 300K dada su brecha de energía [Usar]
2. Analizar cómo se desplaza el nivel de Fermi al aumentar la concentración de donantes [Evaluar]
3. Calcular el potencial de contacto de una unión p-n con concentraciones de dopaje dadas [Usar]

5.6.5. SSP/Propiedades Magnéticas: Para-, Dia- y Ferromagnetismo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Orígenes de los momentos magnéticos y el ordenamiento colectivo en sólidos.

Temas:

Core

- Diamagnetismo de Langevin y ley de Curie para el paramagnetismo
- Campo molecular de Weiss e interacción de intercambio de Heisenberg
- La ley de Curie-Weiss y la transición de fase paramagnética-ferromagnética
- Antiferromagnetismo (temperatura de Néel) y ferrimagnetismo
- Energía de pared de dominio y el ciclo de histéresis magnética
- Ondas de espín cuantizadas (magnones) y su excitación térmica

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la susceptibilidad magnética de una sal paramagnética a una temperatura específica [Usar]
2. Comparar las curvas de susceptibilidad de materiales ferromagnéticos y antiferromagnéticos [Evaluar]
3. Estimar la temperatura de Curie a partir de la constante de campo molecular en un ferromagneto [Usar]

5.6.6. SSP/Superconductividad: Teoría BCS y Alta Tc (Core Tier1: 1 hr, Core Tier2: 1 hr)

Resistencia cero y diamagnetismo perfecto en fluidos cuánticos.
Temas:
Core

- El efecto Meissner-Ochsenfeld y la expulsión del flujo magnético
- Superconductores Tipo I y Tipo II: campos críticos y vórtices
- Ecuaciones de London y longitud de penetración magnética λ_L
- Teoría BCS: pares de Cooper, atracción mediada por fonones y brecha de energía
- Efectos Josephson DC y AC en uniones superconductoras

Electivo

- Perspectiva de los cupratos de alta Tc y superconductividad no convencional

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Describir la dependencia del campo magnético crítico con la temperatura [Familiarizarse]
2. Calcular la profundidad de penetración de London para un superconductor de plomo [Usar]
3. Estimar la brecha de energía superconductora a partir de la temperatura crítica usando la teoría BCS [Evaluar]

5.6.7. SSP/Física de la Materia Blanda y Polímeros (Core Tier1: 1 hr, Core Tier2: 1 hr)

Complejidad en sistemas donde la energía térmica es comparable a la energía de interacción.
Temas:
Core

- Cristales líquidos: fases nemática, esméctica y colestérica
- Modelos de cadena ideal: caminos aleatorios y teoría de Flory
- Viscoelasticidad y transición vítrea en sólidos amorfos

Electivo

- Estabilidad coloidal, teoría DLVO y autoensamblaje

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar el origen entrópico de la elasticidad en polímeros similares al caucho [Evaluar]
2. Calcular la distancia extremo a extremo cuadrática media para una cadena libremente articulada [Usar]
3. Explicar el concepto de temperatura de transición vítrea en polímeros amorfos [Familiarizarse]

5.6.8. SSP/Nanoestructuras y Sistemas de Baja Dimensionalidad (Core Tier1: 1 hr, Core Tier2: 1 hr)

Efectos de confinamiento cuántico en materiales 0D, 1D y 2D.
Temas:
Core

- Densidad de estados en pozos cuánticos (2D), hilos (1D) y puntos (0D)
- Estructura electrónica del grafeno y propiedades de los nanotubos de carbono
- Tunelamiento de un solo electrón y bloqueo de Coulomb

- Efecto Hall cuántico entero y niveles de Landau

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular el desplazamiento del primer nivel de energía en un pozo cuántico de GaAs de 10nm [Usar]
2. Analizar la densidad de estados para un gas de electrones 2D frente a un nanoalambres 1D [Evaluar]
3. Calcular el espaciado de niveles de Landau en un gas bidimensional de electrones bajo un campo magnético [Usar]

5.6.9. SSP/Espintrónica y Aislantes Topológicos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Explotación del espín del electrón y la topología para la próxima generación de electrónica.

Temas:
Core

- Magnetorresistencia gigante (GMR) y sensores de válvula de espín
- Acoplamiento espín-órbita y el efecto Spin Hall

Electivo

- Aislantes topológicos: correspondencia bulto-borde y estados protegidos
- Estados ligados de Majorana en superconductores topológicos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Analizar el papel de la dispersión dependiente del espín en dispositivos GMR [Evaluar]
2. Estimar el ángulo Hall de espín en un metal pesado a partir de datos experimentales [Usar]
3. Describir el concepto de estados de borde topológicamente protegidos [Familiarizarse]

5.6.10. SSP/Sistemas de Electrones Fuertemente Correlacionados (Core Tier1: 1 hr, Core Tier2: 1 hr)

Física más allá de la aproximación de una sola partícula: efectos de muchos cuerpos.

Temas:
Core

- Aislantes de Mott y el colapso de la teoría de bandas
- El efecto Kondo y la dispersión por impurezas magnéticas

Electivo

- El modelo de Hubbard para electrones correlacionados en una red
- Sistemas de fermiones pesados y renormalización de masa

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Evaluar cuándo la repulsión electrón-electrón domina sobre la energía cinética en el modelo de Hubbard [Evaluar]
2. Estimar la temperatura de Kondo a partir de la constante de acoplamiento de intercambio [Usar]
3. Explicar la transición metal-aislante de Mott en términos de la correlación electrónica [Familiarizarse]

5.7. Física Nuclear y de Partículas (NPP)

La Física Nuclear y de Partículas explora los constituyentes fundamentales de la materia y las fuerzas que gobiernan sus interacciones. Este campo abarca desde la estructura y estabilidad del núcleo atómico hasta las interacciones de alta energía de quarks y leptones descritas por el Modelo Estándar.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.7.1 Estructura Nuclear, Energía de Enlace y Modelo de Capas (Pág. 257)	1	1
⭐ 5.7.2 Radiactividad y Desintegraciones Nucleares (Alfa, Beta, Gamma) (Pág. 257)	1	1
⭐ 5.7.3 Reacciones Nucleares, Fisión y Fusión (Pág. 258)	1	1
⭐ 5.7.4 Aceleradores y Detectores de Partículas (Pág. 258)	1	1
⭐ 5.7.5 El Modelo Estándar de la Física de Partículas (Pág. 258)	1	1
⭐ 5.7.6 Quarks, Leptones y Bosones de Norma (Pág. 259)	1	1
⭐ 5.7.7 Simetrías y Leyes de Conservación (CPT) (Pág. 259)	1	1
⭐ 5.7.8 Interacciones Fundamentales y Diagramas de Feynman (Pág. 259)	1	1
⭐ 5.7.9 Mecanismo de Higgs y más allá del Modelo Estándar (Pág. 260)	1	1
⭐ 5.7.10 Física de Neutrinos y Oscilación (Pág. 260)	1	1

Cuadro 5.7: Lista de KUs del área de Física Nuclear y de Partículas.

5.7.1. NPP/Estructura Nuclear, Energía de Enlace y Modelo de Capas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades estáticas de los núcleos y los modelos que describen la organización de los nucleones.

Temas:

Core

- Radios nucleares, distribución de carga y la ley de escala $R = R_0 A^{1/3}$
- El Modelo de la Gota Líquida y la Fórmula Semiempírica de Masa (SEMF)
- Curva de energía de enlace por nucleón y tendencias de estabilidad nuclear
- El Modelo de Capas Nuclear: Evidencia de los números mágicos
- Acoplamiento espín-órbita fuerte y reordenamiento de los niveles de energía nucleares
- Momentos dipolares magnéticos y cuadripolares eléctricos del núcleo
- Modelo colectivo: Estados excitados rotacionales y vibracionales en núcleos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía de enlace total de un núcleo de Carbono-12 usando la SEMF [Usar]
2. Predecir el espín y paridad del estado fundamental de núcleos con A impar usando el Modelo de Capas [Evaluar]
3. Determinar la energía de separación de neutrones para un isótopo específico a partir de datos de masa [Usar]

5.7.2. NPP/Radiactividad y Desintegraciones Nucleares (Alfa, Beta, Gamma) (Core Tier1: 1 hr, Core Tier2: 1 hr)

La física de los núcleos inestables y los tres modos principales de desintegración radiactiva.

Temas:

Core

- La ley de desintegración exponencial, vida media y conceptos de vida promedio
- Desintegración Alfa: Teoría de Gamow del tunelamiento cuántico a través de la barrera de Coulomb
- Desintegración Beta (β^-, β^+) y Captura Electrónica: El papel del neutrino
- Teoría de Fermi de la desintegración beta y el gráfico de Kurie
- Desintegración Gamma: Transiciones multipolares y conversión interna
- Reglas de selección para transiciones nucleares basadas en el momento angular y la paridad
- Cadenas de desintegración radiactiva y equilibrio secular/transitorio

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la actividad de una muestra radiactiva después de cinco vidas medias [Usar]
2. Analizar por qué el espectro de partículas beta es continuo mientras que el de alfa es discreto [Evaluar]

3. Evaluar si una transición E1 o M1 está permitida entre dos estados nucleares [Usar]

5.7.3. NPP/Reacciones Nucleares, Fisión y Fusión (Core Tier1: 1 hr, Core Tier2: 1 hr)

Energética y mecanismos de transmutaciones nucleares y liberación de energía.

Temas:

Core

- Valor Q de la reacción y energía umbral para reacciones endotérmicas
- Secciones eficaces nucleares y fórmula de resonancia de Breit-Wigner
- Mecanismos de fisión inducida y el principio de reacción en cadena
- Fusión nuclear: La cadena p-p y el ciclo CNO en la nucleosíntesis estelar
- Moderación de neutrones, fórmula de los cuatro factores y criticidad del reactor
- Clasificación de reacciones: Reacciones directas frente a formación de núcleo compuesto

Electivo

- Fusión por confinamiento inercial y magnético: Criterio de Lawson

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la energía cinética mínima requerida para una reacción (α, n) [Usar]
2. Analizar la distribución de masa de los fragmentos de fisión en el U-235 [Evaluar]
3. Calcular la energía liberada en un solo evento de fusión Deuterio-Tritio [Usar]

5.7.4. NPP/Aceleradores y Detectores de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Herramientas experimentales para la física de alta energía y medición de radiación.

Temas:

Core

- Principios de Aceleradores Lineales (Linacs) y cavidades de RF
- Ciclotrones, Sincrotrones y el principio de estabilidad de fase
- Cámaras de ionización, contadores proporcionales y tubos Geiger-Müller
- Contadores de centelleo y tubos fotomultiplicadores (PMTs)
- Detectores de tiras de silicio y de germanio de alta pureza (HPGe)
- Calorímetros electromagnéticos/hadrónicos y detectores de radiación Cherenkov
- Física de colisionadores: Energía en el centro de masas y luminosidad integrada

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la frecuencia de ciclotrón para un protón en un campo de 1.5 Tesla [Usar]
2. Seleccionar el tipo de detector apropiado para espectroscopía de rayos gamma de alta resolución [Evaluar]
3. Calcular la luminosidad integrada necesaria para producir un número dado de eventos [Usar]

5.7.5. NPP/El Modelo Estándar de la Física de Partículas (Core Tier1: 1 hr, Core Tier2: 1 hr)

El marco unificado que describe el electromagnetismo y las interacciones débil y fuerte.

Temas:

Core

- Las cuatro fuerzas fundamentales: rango, intensidad y partículas mediadoras
- Clasificación de partículas: Fermiones (materia) y Bosones (portadores de fuerza)
- Estructura conceptual del Lagrangiano del Modelo Estándar
- Invarianza de norma local como generadora de interacciones
- Las tres generaciones de quarks y leptones
- Unificación electrodébil y la escala de la Teoría de la Gran Unificación (GUT)
- Libertad asintótica y la evolución de las constantes de acoplamiento

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Enumerar todas las partículas del Modelo Estándar y sus propiedades básicas (espín, carga) [Familiarizarse]
2. Contrastar las propiedades del Gluón y el Fotón [Evaluar]
3. Explicar por qué la constante de acoplamiento fuerte disminuye a altas energías [Usar]

5.7.6. NPP/Quarks, Leptones y Bosones de Norma (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades específicas de los bloques fundamentales del universo.
Temas:
Core

- Sabores de quarks (u, d, s, c, b, t) y carga de color (R, G, B)
- Clasificación de hadrones: Bariones (qqq) y Mesones (q-antiquark)
- Familias de leptones y la conservación del número leptónico (L_e, L_μ, L_τ)
- Los bosones W^+, W^- y Z^0: Mediadores de la fuerza débil
- Gluones y el principio de confinamiento de color
- Extrañeza y el descubrimiento de kaones e hiperones
- Charmonium y Bottomonium: Espectroscopía de quarks pesados

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Determinar el contenido de quarks para un protón, neutrón y pion [Usar]
2. Calcular la carga eléctrica total y extrañeza de un barión Ω^- [Evaluar]
3. Asignar los números cuánticos de sabor a un estado hadrónico dado [Usar]

5.7.7. NPP/Simetrías y Leyes de Conservación (CPT) (Core Tier1: 1 hr, Core Tier2: 1 hr)

La conexión profunda entre las simetrías y los invariantes físicos.
Temas:
Core

- Teorema de Noether: Simetrías espacio-temporales y conservación de E, p, L
- Conservación del número bariónico (B) y el número leptónico (L)
- Simetría de Isospín en la interacción fuerte (simetría p/n)
- Violación de la Paridad (P) en interacciones débiles (experimento de Wu)
- Conjugación de Carga (C) y la simetría C
- Violación CP en el sistema de kaones neutros
- El Teorema CPT y sus implicaciones para la paridad de masa partícula-antipartícula

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Comprobar si una reacción de partículas propuesta viola la conservación de B, L o Q [Usar]
2. Evaluar la paridad de un sistema de dos piones en un estado orbital específico [Evaluar]
3. Predecir posibles desintegraciones con violación de CP en el sistema de mesones B [Usar]

5.7.8. NPP/Interacciones Fundamentales y Diagramas de Feynman (Core Tier1: 1 hr, Core Tier2: 1 hr)

Visualización y cálculo de interacciones de partículas mediante diagramas de intercambio.
Temas:
Core

- Reglas básicas para dibujar diagramas de Feynman: Vértices y propagadores
- Vértices de Electrodinámica Cuántica (QED) y acoplamiento electrón-fotón
- Vértices débiles: Corriente cargada ($W^\pm$) y corriente neutra (Z)
- Vértices de Cromodinámica Cuántica (QCD) y auto-interacción de gluones
- Relación entre diagramas de Feynman y elementos de la matriz de transición

- Partículas fuera de la capa de masa y significado físico de los propagadores
- Introducción a los diagramas de bucle y conceptos de renormalización

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Dibujar un diagrama de Feynman para la desintegración de un muón en un electrón y neutrinos [Usar]
2. Identificar si un diagrama representa una interacción EM, débil o fuerte basada en los vértices [Evaluar]
3. Estimar el elemento de matriz para la dispersión electrón-electrón en QED [Usar]

5.7.9. NPP/Mecanismo de Higgs y más allá del Modelo Estándar (Core Tier1: 1 hr, Core Tier2: 1 hr)

Origen de la masa y limitaciones de la teoría actual.

Temas:

Core

- Ruptura espontánea de la simetría y bosones de Goldstone
- El potencial del campo de Higgs (sombrero mexicano) y valor esperado del vacío
- Generación de masa a través del acoplamiento de Yukawa con el campo de Higgs
- El problema de la jerarquía y el ajuste fino en el Modelo Estándar

Electivo

- Introducción a la Supersimetría (SUSY) y supercompañeros
- Teorías de Gran Unificación (GUTs) y candidatos WIMP para la materia oscura

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Explicar cómo el mecanismo de Higgs proporciona masa a los bosones W y Z [Familiarizarse]
2. Calcular el valor de expectativa del vacío del campo de Higgs a partir de la constante de Fermi [Usar]
3. Evaluar la magnitud del problema de jerarquía en el Modelo Estándar [Evaluar]

5.7.10. NPP/Física de Neutrinos y Oscilación (Core Tier1: 1 hr, Core Tier2: 1 hr)

La naturaleza elusiva de los neutrinos y la evidencia de una masa no nula.

Temas:

Core

- Fuentes de neutrinos: Neutrinos solares, atmosféricos y de reactor
- El experimento de Reines-Cowan y escalas de la sección eficaz de los neutrinos
- Mezcla de sabores de neutrinos y la matriz PMNS
- Fórmula de oscilación de neutrinos: Aproximación de dos sabores
- El problema de los neutrinos solares y el efecto MSW

Electivo

- Neutrinos de Dirac frente a Majorana y desintegración beta doble sin neutrinos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la longitud de oscilación para un neutrino de energía 1 GeV [Usar]
2. Evaluar las implicaciones de las oscilaciones de neutrinos para la suposición de masa nula del Modelo Estándar [Evaluar]
3. Calcular la probabilidad de supervivencia del neutrino electrónico en función de la distancia recorrida [Usar]

5.8. Información y Computación Cuántica (QIC)

La Información y Computación Cuántica aprovecha los principios de la mecánica cuántica (superposición, entrelazamiento e interferencia) para procesar y transmitir información de formas que superan los límites clásicos. Esta área cubre los fundamentos teóricos, el desarrollo de algoritmos y las arquitecturas físicas del hardware cuántico.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.8.1 Qubits, Superposición y Entrelazamiento (Pág. 261)	1	1
⭐ 5.8.2 Comuertas y Circuitos Lógicos Cuánticos (Pág. 261)	1	1
⭐ 5.8.3 Algoritmos Cuánticos (Shor, Grover) (Pág. 262)	1	1
⭐ 5.8.4 Teletransportación y Criptografía Cuántica (Pág. 262)	1	1
⭐ 5.8.5 Decoherencia y Corrección de Errores Cuánticos (Pág. 262)	1	1
⭐ 5.8.6 Realizaciones Físicas de Computadoras Cuánticas (Pág. 263)	1	1
⭐ 5.8.7 Sensores Cuánticos y Metrología (Pág. 263)	1	1
⭐ 5.8.8 Redes Cuánticas y Comunicación (Pág. 263)	1	1

Cuadro 5.8: Lista de KUs del área de Información y Computación Cuántica.

5.8.1. QIC/Qubits, Superposición y Entrelazamiento (Core Tier1: 1 hr, Core Tier2: 1 hr)

Las unidades fundamentales de la información cuántica y sus correlaciones no clásicas.

Temas:

Core

- La Esfera de Bloch: Visualización de qubits en estados puros
- Principio de superposición lineal y estados de la base computacional $|0\rangle, |1\rangle$
- Los cuatro estados de Bell y pares máximamente entrelazados
- Formalismo de la matriz de densidad para estados puros y mixtos
- Descomposición de Schmidt y el concepto de purificación de estados
- Traza parcial de subsistemas y operadores de densidad reducida

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Representar un estado cuántico dado como coordenadas en la Esfera de Bloch [Usar]
2. Utilizar la traza parcial para determinar si un estado de dos qubits está entrelazado o es separable [Evaluar]
3. Calcular la entropía de Von Neumann para cuantificar el grado de entrelazamiento en un sistema [Usar]

5.8.2. QIC/Comuertas y Circuitos Lógicos Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Transformaciones unitarias y la construcción de circuitos cuánticos universales.

Temas:

Core

- Comuertas de un solo qubit: Pauli X, Y, Z, Hadamard (H) y fases (S, T)
- Comuertas de dos qubits: Operaciones CNOT, CZ y SWAP
- Definición de conjuntos de comuertas universales para la computación cuántica
- Comuertas de tres qubits: Comuertas Toffoli (CCNOT) y Fredkin (CSWAP)
- El principio de computación reversible en circuitos cuánticos
- Descomposición de matrices unitarias arbitrarias en comuertas elementales

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Construir un circuito cuántico usando comuertas H y CNOT para generar un estado de Bell [Usar]
2. Evaluar el vector de estado resultante tras aplicar una secuencia de comuertas de Pauli [Evaluar]
3. Sintetizar una comuerta Toffoli usando solo comuertas CNOT de 2 qubits y comuertas de 1 qubit [Usar]

5.8.3. QIC/Algoritmos Cuánticos (Shor, Grover) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aprovechamiento de la aceleración cuántica para factorización, búsqueda y simulación.

Temas:

Core

- La Transformada de Fourier Cuántica (QFT) y su implementación en circuitos
- El algoritmo de Deutsch-Jozsa y el concepto de aceleración basada en oráculos
- Algoritmo de Shor: Búsqueda de periodos y la amenaza al criptosistema RSA
- Algoritmo de Grover: Amplificación de amplitud y el operador de difusión
- El protocolo de Estimación de Fase Cuántica (QPE)

Electivo

- Variational Quantum Eigensolver (VQE) y QAOA para dispositivos NISQ

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Aplicar la QFT a un registro de 3 qubits y determinar el estado de salida [Usar]
2. Analizar la aceleración $O(\sqrt{N})$ del algoritmo de Grover en comparación con la búsqueda lineal clásica [Evaluar]
3. Seguir los pasos del algoritmo de Estimación de Fase para hallar el autovalor de una comuerta T [Usar]

5.8.4. QIC/Teletransportación y Criptografía Cuántica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Protocolos para la transferencia segura de información y reconstrucción de estados.

Temas:

Core

- El protocolo de Teletransportación Cuántica: Medición y corrección clásica
- El Teorema de No-Clonación y sus implicaciones para la seguridad cuántica
- El protocolo BB84 de Distribución de Claves Cuánticas (QKD)
- Protocolo Ekert91: QKD basado en entrelazamiento y la desigualdad de Bell
- Codificación superdensa: Envío de dos bits clásicos mediante un solo qubit
- Preparación remota de estados e intercambio de entrelazamiento (entanglement swapping)

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Diagramar la secuencia de comuertas y mediciones requeridas para teletransportar un qubit [Usar]
2. Identificar la presencia de espionaje en un protocolo BB84 analizando las tasas de error en la clave filtrada [Evaluar]
3. Ejecutar el mapeo de bits a comuertas para una transmisión de codificación superdensa [Usar]

5.8.5. QIC/Decoherencia y Corrección de Errores Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Protección de la información cuántica frente al ruido y las interacciones ambientales.

Temas:

Core

- Tipos de ruido cuántico: Canales de bit-flip, phase-flip y de despolarización
- Escalas de tiempo de relajación (T_1) y desfasamiento (T_2)
- El código de bit-flip de 3 qubits y votación por mayoría
- El código de 9 qubits de Shor: Corrección de errores arbitrarios en un solo qubit
- El formalismo estabilizador para códigos de corrección de errores cuánticos

Electivo

- Introducción a los códigos de superficie y protección topológica

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Evaluar el impacto del desfasamiento en los elementos fuera de la diagonal de una matriz de densidad [Evaluar]
2. Calcular el síndrome de error para un código de 3 qubits usando mediciones de estabilizadores [Usar]
3. Analizar el teorema del umbral para la computación cuántica tolerante a fallos [Evaluar]

5.8.6. QIC/Realizaciones Físicas de Computadoras Cuánticas (Core Tier1: 1 hr, Core Tier2: 1 hr)

La ingeniería de sistemas físicos para satisfacer los criterios de DiVincenzo.

Temas:

Core

- Criterios de DiVincenzo para una computadora cuántica escalable
- Qubits superconductores: Cajas de pares de Cooper y qubits Transmon
- Computación cuántica de iones atrapados: Trampas de Paul y comuertas de Cirac-Zoller
- Computación cuántica óptica lineal (LOQC) y estados comprimidos
- Arreglos de átomos neutros e interacciones de bloqueo de Rydberg

Electivo

- Puntos cuánticos semiconductores y qubits basados en espín

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Comparar los tiempos de coherencia y velocidades de comuerta de qubits superconductores frente a trampas de iones [Familiarizarse]
2. Analizar el Hamiltoniano de un qubit Transmon y su anarmonicidad [Evaluar]
3. Diseñar una configuración de cadena de iones 1D para una operación de comuerta específica [Usar]

5.8.7. QIC/Sensores Cuánticos y Metrología (Core Tier1: 1 hr, Core Tier2: 1 hr)

Uso de estados cuánticos para lograr mediciones de precisión más allá del Límite Cuántico Estándar.

Temas:

Core

- El Límite Cuántico Estándar y el Límite de Heisenberg
- Luz comprimida (squeezed light) y su aplicación en la detección de ondas gravitacionales (LIGO)
- Centros de vacante de nitrógeno (NV) en diamante para la detección de campos magnéticos
- Interferometría de Ramsey y la física de los relojes atómicos
- Información de Fisher cuántica y precisión en la estimación de fase

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la escala de precisión de medición para N partículas entrelazadas frente a N partículas independientes [Usar]
2. Evaluar la sensibilidad de un sensor de centro NV ante un campo magnético de escala nano-Tesla [Evaluar]
3. Modelar una secuencia de pulsos de Ramsey para medir un desplazamiento de frecuencia [Usar]

5.8.8. QIC/Redes Cuánticas y Comunicación (Core Tier1: 1 hr, Core Tier2: 1 hr)

Escalado de sistemas cuánticos mediante distribución de entrelazamiento y repetidores cuánticos.

Temas:

Core

- Repetidores cuánticos: Purificación de entrelazamiento y mediciones del estado de Bell
- Intercambio de entrelazamiento para comunicación cuántica de larga distancia
- Memoria cuántica: Almacenamiento de estados cuánticos en interfaces luz-materia
- Comunicación cuántica en espacio libre y satelital (misión Micius)

Electivo

- Arquitecturas para una internet cuántica global

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar una cadena de repetidores cuánticos para conectar dos nodos separados por 1000 km [Usar]
2. Evaluar la fidelidad de un estado cuántico tras su recuperación de un medio de almacenamiento [Evaluar]
3. Calcular la probabilidad de éxito en la distribución de entrelazamiento sobre un enlace de fibra con pérdidas [Usar]

5.9. Óptica y Fotónica (OPH)

La Óptica y la Fotónica estudian el comportamiento y las propiedades de la luz y su interacción con la materia. Este campo abarca las leyes clásicas de reflexión y refracción, la naturaleza ondulatoria de la luz manifestada en interferencias y difracción, y los avances modernos en láseres, fibra óptica y nanofotónica.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.9.1 Óptica Geométrica e Instrumentos Ópticos (Pág. 264)	1	1
⭐ 5.9.2 Óptica Ondulatoria: Interferencia y Difracción (Pág. 265)	1	1
⭐ 5.9.3 Polarización y Óptica de Cristales (Pág. 265)	1	1
⭐ 5.9.4 Física de Láseres: Principios y Aplicaciones (Pág. 265)	1	1
⭐ 5.9.5 Óptica No Lineal y Conversión de Frecuencia (Pág. 266)	1	1
⭐ 5.9.6 Fibra Óptica y Cristales Fotónicos (Pág. 266)	1	1
⭐ 5.9.7 Óptica Ultrarrápida y Espectroscopía (Pág. 266)	1	1
⭐ 5.9.8 Nano-óptica y Plasmónica (Pág. 267)	1	1

Cuadro 5.9: Lista de KUs del área de Óptica y Fotónica.

5.9.1. OPH/Óptica Geométrica e Instrumentos Ópticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

La aproximación de rayos de la luz y el diseño de sistemas de formación de imágenes.
Temas:
Core

- Principio de Fermat y la derivación de las leyes de reflexión/refracción
- La fórmula del fabricante de lentes y ecuaciones de espejos esféricos
- Lentes gruesas, puntos cardinales y matrices de sistema
- Teoría de Abbe de formación de imágenes y límites de resolución
- Microscopios compuestos, telescopios keplerianos/galileanos y cámaras
- Trazado de rayos paraxiales usando el formalismo de matrices ABCD
- Las cinco aberraciones de Seidel: Esférica, Coma, Astigmatismo, Curvatura, Distorsión

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Diseñar un sistema de telescopio con un aumento angular específico usando dos lentes delgadas [Usar]
2. Evaluar la posición y naturaleza de una imagen formada por una combinación de espejo y lente [Evaluar]
3. Calcular la matriz ABCD equivalente para un tren óptico complejo de tres componentes [Usar]

5.9.2. OPH/Óptica Ondulatoria: Interferencia y Difracción (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fenómenos de naturaleza ondulatoria de la luz y la resolución de sistemas ópticos.

Temas:

Core

- Principio de superposición y representación fasorial de las ondas
- Experimento de la doble rendija de Young e intensidad de las franjas de interferencia
- Interferómetro de Michelson: Medición de desplazamientos pequeños
- Difracción de Fraunhofer por rendijas simples, dobles y múltiples (N)
- Difracción por aberturas circulares y el disco de Airy
- El criterio de Rayleigh para el límite de resolución
- Difracción de Fresnel y placas de zona de Fresnel
- Interferencia de haces múltiples e interferómetro de Fabry-Perot

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el espaciado de franjas en una pantalla para un montaje de doble rendija con luz de 532 nm [Usar]
2. Determinar la separación mínima resoluble de dos estrellas usando un telescopio de 10 pulgadas [Evaluar]
3. Calcular la fineza y el rango espectral libre de una cavidad Fabry-Perot [Usar]

5.9.3. OPH/Polarización y Óptica de Cristales (Core Tier1: 1 hr, Core Tier2: 1 hr)

La naturaleza vectorial de la luz y su interacción con medios anisotrópicos.

Temas:

Core

- Estados de polarización lineal, circular y elíptica
- Ley de Malus para la intensidad de la luz polarizada
- Polarización por reflexión y ángulo de Brewster
- Birrefringencia en cristales uniaxiales: Rayos ordinarios y extraordinarios
- Placas de cuarto de onda y media onda: Retardo de fase
- Vectores y matrices de Jones para el análisis de estados de polarización
- Actividad óptica y rotación de Faraday

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Determinar la intensidad final de luz no polarizada que pasa por tres polarizadores con desfases de 45 grados [Usar]
2. Analizar el estado de polarización de salida de luz linealmente polarizada tras pasar por una placa de cuarto de onda [Evaluar]
3. Usar el cálculo de Jones para hallar el estado resultante tras un polarizador y un retardador [Usar]

5.9.4. OPH/Física de Láseres: Principios y Aplicaciones (Core Tier1: 1 hr, Core Tier2: 1 hr)

Generación de luz coherente mediante emisión estimulada.

Temas:

Core

- Coeficientes A y B de Einstein para absorción y emisión
- Condiciones para la acción láser: Inversión de población y bombeo óptico
- Resonadores ópticos: Modos longitudinales y transversales (TEM)
- Esquemas de niveles de energía: Láseres de 3 niveles frente a 4 niveles
- Propagación de haces gaussianos: Cintura del haz, rango de Rayleigh y divergencia
- Operación pulsada: Técnicas de Q-switching y Mode-locking
- Características de los láseres de He-Ne, Nd:YAG y de diodo

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la ganancia umbral requerida para la acción láser en una cavidad con pérdidas de espejo conocidas [Usar]
2. Explicar por qué un sistema de 2 niveles no puede lograr una inversión de población en estado estacionario [Evaluar]
3. Determinar el tamaño del punto enfocado de un haz láser gaussiano usando una lente dada [Usar]

5.9.5. OPH/Óptica No Lineal y Conversión de Frecuencia (Core Tier1: 1 hr, Core Tier2: 1 hr)

Interacciones luz-materia donde la polarización depende de forma no lineal del campo eléctrico.
Temas:
Core

- Tensores de susceptibilidad no lineal ($\chi^{(2)}, \chi^{(3)}$)
- Generación de segundo armónico (SHG) y duplicación de frecuencia
- Condición de ajuste de fase (phase matching): Conservación del momento en procesos no lineales
- Efectos electro-ópticos: Efecto Pockels y efecto Kerr
- Oscilación paramétrica óptica (OPO) y conversión descendente (down-conversion)

Electivo

- Autoenfoque, automodulación de fase y solitones ópticos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la eficiencia de conversión de SHG para una longitud específica de un cristal no lineal [Usar]
2. Evaluar la orientación requerida del cristal para lograr el ajuste de fase tipo I [Evaluar]
3. Modelar el desplazamiento de fase inducido por el efecto Kerr en una fibra óptica [Usar]

5.9.6. OPH/Fibra Óptica y Cristales Fotónicos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Guía de luz en guías de onda y estructuras dieléctricas periódicas.
Temas:
Core

- Reflexión interna total y estructuras de fibra de índice escalonado
- Apertura Numérica (NA) y el cono de aceptación de una fibra
- Fibras monomodo frente a multimodo y el número V
- Degradación de señal: Atenuación y dispersión cromática/modal
- Cristales fotónicos y la emergencia de brechas de banda fotónicas (bandgaps)
- Fibras microestructuradas y de cristal fotónico (PCFs)

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la apertura numérica de una fibra con índice de núcleo 1.48 y revestimiento 1.46 [Usar]
2. Determinar la longitud de onda máxima para operación monomodo en una fibra dada [Evaluar]
3. Evaluar el ensanchamiento del pulso en un enlace de 50 km debido a la dispersión cromática [Usar]

5.9.7. OPH/Óptica Ultrarrápida y Espectroscopía (Core Tier1: 1 hr, Core Tier2: 1 hr)

Generación y medición de pulsos de luz de femtosegundos y attosegundos.
Temas:
Core

- Generación de pulsos de femtosegundos mediante láseres de Ti:Zafiro
- El producto tiempo-ancho de banda y pulsos limitados por transformada

- Dispersión de velocidad de grupo (GVD) y chirrido (chirping) de pulsos
- Caracterización de pulsos usando autocorrelación de intensidad
- Técnicas de espectroscopía de bomba-sonda (pump-probe) resueltas en el tiempo

Electivo

- Generación de armónicos de alto orden y ciencia de attosegundos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el ancho de pulso ensanchado tras pasar por un bloque de vidrio con una GVD conocida [Usar]
2. Interpretar una traza de autocorrelación experimental para estimar la duración del pulso [Evaluar]
3. Modelar el decaimiento de una población en estado excitado usando datos de bomba-sonda [Usar]

5.9.8. OPH/Nano-óptica y Plasmónica (Core Tier1: 1 hr, Core Tier2: 1 hr)

Interacción de la luz con estructuras de sub-longitud de onda y nanoestructuras metálicas.

Temas:

Core

- Ruptura del límite de difracción: Óptica de campo cercano frente a campo lejano
- Plasmones de superficie polaritones (SPPs) en interfaces metal-dieléctrico
- Resonancia de plasmón de superficie localizado (LSPR) en nanopartículas metálicas
- Microscopía óptica de barrido de campo cercano (SNOM)
- Metamateriales y estructuras de índice de refracción negativo

Electivo

- Pinzas ópticas: Fuerza y torque sobre nanopartículas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Describir el decaimiento de las ondas evanescentes y su papel en la óptica de campo cercano [Familiarizarse]
2. Analizar el desplazamiento en la frecuencia LSPR debido a cambios en el índice de refracción local [Evaluar]
3. Calcular la longitud de onda de un SPP en una interfaz oro-aire para una luz de excitación dada [Usar]

5.10. Física Computacional y Basada en Datos (CDP)

Este área de conocimiento se centra en la intersección del análisis numérico, la computación de alto rendimiento y la ciencia de datos moderna. Proporciona a los físicos las herramientas para simular sistemas que carecen de soluciones analíticas y para extraer leyes fundamentales de conjuntos masivos de datos mediante el aprendizaje automático y la inferencia estadística.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.10.1 Métodos Numéricos para Ecuaciones Diferenciales en Física (Pág. 268)	1	1
⭐ 5.10.2 Métodos y Simulaciones de Monte Carlo (Pág. 268)	1	1
⭐ 5.10.3 Computación de Alto Rendimiento (HPC) para Simulaciones a Gran Escala (Pág. 268)	1	1
⭐ 5.10.4 Aprendizaje Automático para el Descubrimiento Físico (Pág. 269)	1	1
⭐ 5.10.5 Análisis de Datos y Cuantificación de la Incertidumbre (Pág. 269)	1	1
⭐ 5.10.6 Modelado de Sistemas Adaptativos Complejos (Pág. 269)	1	1
⭐ 5.10.7 Redes Neuronales para Sistemas Cuánticos (Pág. 270)	1	1
⭐ 5.10.8 Redes Neuronales Informadas por la Física (PINNs) (Pág. 270)	1	1

Cuadro 5.10: Lista de KUs del área de Física Computacional y Basada en Datos.

5.10.1. CDP/Métodos Numéricos para Ecuaciones Diferenciales en Física (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aproximaciones discretas para resolver las ecuaciones fundamentales de movimiento y teorías de campos.

Temas:

Core

- Métodos de Runge-Kutta (RK2, RK4) para problemas de valor inicial
- Diferencias Finitas en el Dominio del Tiempo (FDTD) para las ecuaciones de Maxwell
- Problemas de valor en la frontera: Métodos de disparo y técnicas de relajación
- Integradores simplécticos: Algoritmos de Verlet y leapfrog para sistemas Hamiltonianos
- Análisis de estabilidad y la condición de Courant-Friedrichs-Lewy (CFL)
- Método de Elementos Finitos (FEM) para geometrías complejas en elasticidad y flujo de fluidos

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Implementar un algoritmo de Verlet de velocidad para simular un oscilador armónico 1D [Usar]
2. Evaluar la escala de los errores de discretización para diferentes órdenes de integración numérica [Evaluar]
3. Resolver la ecuación de Laplace en 2D para un conjunto dado de potenciales de frontera mediante la iteración de Jacobi [Usar]

5.10.2. CDP/Métodos y Simulaciones de Monte Carlo (Core Tier1: 1 hr, Core Tier2: 1 hr)

Técnicas de muestreo estocástico para la integración multidimensional y la física estadística.

Temas:

Core

- Algoritmos de generación de números pseudoaleatorios y cuasialeatorios
- El algoritmo de Metropolis-Hastings para Monte Carlo por Cadenas de Markov (MCMC)
- Simulación de Monte Carlo del modelo de Ising y transiciones de fase
- Muestreo por importancia y técnicas de reducción de varianza
- Monte Carlo Cinético (KMC) para procesos estocásticos dependientes del tiempo
- Estimación de error estadístico y tiempo de autocorrelación en cadenas MCMC

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Estimar el valor de π utilizando un método de integración de Monte Carlo de acierto o error [Usar]
2. Analizar el efecto del ralentizamiento crítico cerca de la temperatura de Curie en una simulación de Ising [Evaluar]
3. Desarrollar un esquema de muestreo por importancia para integrar una función multidimensional con picos pronunciados [Usar]

5.10.3. CDP/Computación de Alto Rendimiento (HPC) para Simulaciones a Gran Escala (Core Tier1: 1 hr, Core Tier2: 1 hr)

Arquitecturas y modelos de programación para la paralelización masiva de código de física.

Temas:

Core

- Programación de memoria compartida mediante directivas OpenMP
- Sistemas de memoria distribuida y la Interfaz de Paso de Mensajes (MPI)
- Aceleración por GPU y modelos de programación GPGPU (CUDA, OpenCL)
- Análisis de escalabilidad: Ley de Amdahl y Ley de Gustafson
- Métodos de descomposición de dominio para la paralelización de mallas espaciales
- Optimización para la localidad de caché y límites de ancho de banda de memoria

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Paralelizar una adición de vectores a gran escala utilizando sincronización de hilos OpenMP [Usar]
2. Medir la aceleración (speedup) y la eficiencia paralela de un código en múltiples nodos de cómputo [Evaluar]
3. Implementar comunicación MPI no bloqueante para un intercambio de celdas fantasma en una simulación de fluidos [Usar]

5.10.4. CDP/Aprendizaje Automático para el Descubrimiento Físico (Core Tier1: 1 hr, Core Tier2: 1 hr)

Aplicación del aprendizaje estadístico para identificar patrones y modelos subrogados en los datos.
Temas:
Core

- Aprendizaje supervisado: Regresión lineal, Lasso y Ridge para modelos físicos
- Aprendizaje no supervisado: Clustering y PCA para la detección de transiciones de fase
- Ingeniería de características físicas y reducción dimensional
- Regresión por Procesos Gaussianos para modelado subrogado consciente de la incertidumbre
- Regresión simbólica para el descubrimiento de leyes físicas analíticas a partir de datos
- Métodos de kernel para la aproximación de estados cuánticos de muchos cuerpos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Entrenar un modelo de regresión para predecir la superficie de energía potencial de una molécula [Usar]
2. Evaluar el sobreajuste (overfitting) del modelo mediante validación cruzada en un conjunto de datos espectroscópicos [Evaluar]
3. Aplicar Análisis de Componentes Principales (PCA) para visualizar configuraciones de espín de alta dimensión [Usar]

5.10.5. CDP/Análisis de Datos y Cuantificación de la Incertidumbre (Core Tier1: 1 hr, Core Tier2: 1 hr)

Marcos estadísticos para la validación de modelos y la cuantificación de errores experimentales.
Temas:
Core

- Inferencia bayesiana: Priores, verosimilitudes y distribuciones a posteriori
- Estimación de Máxima Verosimilitud (MLE) para el ajuste de parámetros
- Propagación lineal y no lineal de incertidumbres experimentales
- Técnicas de remuestreo: Métodos de Bootstrapping y Jackknife
- Criterios de selección de modelos: AIC, BIC y razones de evidencia
- Análisis de sensibilidad global e índices de Sobol

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Realizar un ajuste de parámetros bayesiano para un conjunto de datos de desintegración radiactiva [Usar]
2. Calcular intervalos de confianza para una constante física derivada de mediciones ruidosas [Evaluar]
3. Calcular la matriz de Información de Fisher para determinar los límites de precisión de una estimación de parámetros [Usar]

5.10.6. CDP/Modelado de Sistemas Adaptativos Complejos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Dinámica de sistemas con comportamiento emergente e interacciones no lineales.
Temas:

Core
- Modelado basado en agentes (ABM) para sistemas sociales y biológicos
- Ciencia de redes: Dinámica de redes de mundo pequeño y libres de escala
- Criticidad autoorganizada (SOC) y distribuciones de ley de potencia
- Autómatas celulares: Juego de la Vida y modelos de flujo de fluidos (LBM)
- Sincronización colectiva y el modelo de Kuramoto

Electivo
- Teoría de juegos evolutiva y estabilidad ecológica

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Simular el flujo de un fluido en 2D alrededor de un obstáculo mediante el método de Lattice Boltzmann [Usar]
2. Analizar la robustez de una red libre de escala frente a ataques dirigidos [Evaluar]
3. Modelar una pila de arena para demostrar los principios de la criticidad autoorganizada [Usar]

5.10.7. CDP/Redes Neuronales para Sistemas Cuánticos (Core Tier1: 1 hr, Core Tier2: 1 hr)

Representación y resolución de problemas cuánticos de muchos cuerpos mediante redes neuronales artificiales.

Temas:
Core
- Máquinas de Boltzmann Restringidas (RBM) como Estados Cuánticos Neuronales (NQS)
- Monte Carlo Variacional utilizando funciones de onda de prueba de redes neuronales
- Tomografía de estado cuántico basada en redes neuronales
- CNNs para identificar fases topológicas en redes de espín
- Autoencoders para comprimir datos de funciones de onda de muchos cuerpos

Electivo
- Modelos generativos (GANs, VAEs) para el muestreo de configuraciones cuánticas

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Entrenar una RBM para aproximar el estado fundamental de una cadena de Ising de campo transversal [Usar]
2. Comparar la precisión de los Estados Cuánticos Neuronales frente a los métodos tradicionales de Hartree-Fock [Evaluar]
3. Reconstruir una matriz de densidad a partir de datos de medición experimentales utilizando una red neuronal [Usar]

5.10.8. CDP/Redes Neuronales Informadas por la Física (PINNs) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Integración de leyes físicas directamente en las funciones de pérdida de arquitecturas de aprendizaje profundo.

Temas:
Core
- Incorporación de PDEs como residuales en las funciones de pérdida de redes neuronales
- Aprovechamiento de la diferenciación automática para restricciones de derivadas exactas
- Aprendizaje eficiente en datos: Resolución de problemas inversos con datos escasos
- Aplicación de PINNs a las ecuaciones de Navier-Stokes
- PINNs bayesianas para la cuantificación de la incertidumbre en soluciones de PDE

Electivo
- Introducción a los Operadores Neuronales (DeepONet) para PDEs paramétricas

Aprendizaje esperado (*Learning Outcomes*):
Core:
1. Resolver la ecuación de Burgers 1D utilizando una arquitectura PINN [Usar]

2. Comparar el coste computacional y la precisión de una PINN frente a un resolvedor tradicional de Elementos Finitos [Evaluar]
3. Usar una PINN para estimar el coeficiente de viscosidad de un fluido a partir de datos de visualización de flujo [Usar]

5.11. Biofísica y Física Médica (BMP)

La Biofísica y la Física Médica aplican las teorías y métodos de la física para comprender los sistemas biológicos y mejorar la salud humana. Desde las fuerzas mecánicas dentro de una sola célula hasta la compleja física de las radiaciones utilizada en la terapia contra el cáncer.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.11.1 Física de las Macromoléculas Biológicas (Pág. 271)	1	1
⭐ 5.11.2 Biomecánica y Fuerzas Celulares (Pág. 271)	1	1
⭐ 5.11.3 Física Neuronal y Procesamiento de Información en el Cerebro (Pág. 272)	1	1
⭐ 5.11.4 Física de Imágenes (MRI, CT, PET, Ultrasonido) (Pág. 272)	1	1
⭐ 5.11.5 Radioterapia y Dosimetría (Pág. 272)	1	1
⭐ 5.11.6 Mecánica Estadística de Redes Biológicas (Pág. 273)	1	1
⭐ 5.11.7 Microfluídica y Lab-on-a-Chip (Pág. 273)	1	1

Cuadro 5.11: Lista de KUs del área de Biofísica y Física Médica.

5.11.1. BMP/Física de las Macromoléculas Biológicas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades estructurales y termodinámicas del ADN, proteínas y lípidos.

Temas:

Core

- Plegamiento de proteínas: Energética, entropía y la paradoja de Levinthal
- Elasticidad del ADN: El modelo de cadena tipo gusano (WLC)
- Mecánica de las bicapas lipídicas y transiciones de fase en membranas
- Electrostática en células: Teoría de Poisson-Boltzmann y Debye-Hückel
- Motores moleculares: Kinesina y Miosina como trinquetes brownianos
- Técnicas de molécula individual: Pinzas ópticas y magnéticas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la longitud de persistencia de una molécula de ADN a partir de una curva de fuerza-extensión [Usar]
2. Evaluar la estabilidad térmica de una proteína utilizando su paisaje de energía libre de Gibbs [Evaluar]
3. Modelar la eficiencia de un motor molecular utilizando un enfoque estocástico de Langevin [Usar]

5.11.2. BMP/Biomecánica y Fuerzas Celulares (Core Tier1: 1 hr, Core Tier2: 1 hr)

Fuerzas físicas que rigen la forma, motilidad y adhesión celular.

Temas:

Core

- Elasticidad del citoesqueleto: Filamentos de actina y microtúbulos
- Modelos viscoelásticos de células: Representaciones de Maxwell y Kelvin-Voigt
- Principios de la Microscopía de Fuerza de Tracción (TFM)
- Mecanismos de motilidad celular: Protrusión y retracción de lamelipodios
- Mecanotransducción: Cómo las células convierten estímulos mecánicos en señales químicas
- Presión osmótica y la regulación del volumen celular

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el módulo de Young de una célula utilizando datos de Microscopía de Fuerza Atómica (AFM) [Usar]
2. Analizar la deformación por fluencia (creep) de una membrana celular bajo estrés constante [Evaluar]
3. Modelar el hinchamiento dependiente del tiempo de una célula en una solución hipotónica [Usar]

5.11.3. BMP/Física Neuronal y Procesamiento de Información en el Cerebro (Core Tier1: 1 hr, Core Tier2: 1 hr)

Propiedades eléctricas y estadísticas de las neuronas y redes neuronales.

Temas:

Core

- El modelo de Hodgkin-Huxley de la propagación del potencial de acción
- Potenciales de equilibrio: Las ecuaciones de Nernst y de Goldman-Hodgkin-Katz
- Transmisión sináptica y modelos de "Integrate-and-Fire" con fugas
- Teoría de cables para la atenuación de señales en dendritas y axones
- Codificación neuronal: Estrategias de codificación por tasa frente a codificación temporal
- Redes de Hopfield y la física de la memoria asociativa

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el potencial de Nernst para iones de potasio a través de una membrana neuronal [Usar]
2. Analizar el papel de los canales dependientes de voltaje en la iniciación de un potencial de acción [Evaluar]
3. Aplicar la ecuación de cable para determinar la constante de longitud de una dendrita pasiva [Usar]

5.11.4. BMP/Física de Imágenes (MRI, CT, PET, Ultrasonido) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Principios físicos de la visualización no invasiva del cuerpo humano.

Temas:

Core

- Atenuación de rayos X, ley de Beer-Lambert y Tomografía Computarizada (CT)
- Resonancia Magnética Nuclear (NMR) y las ecuaciones de Bloch para MRI
- Mecanismos de contraste: Tiempos de relajación T1 y T2 en MRI
- Impedancia acústica y reflexión en ultrasonido médico
- Física de la Tomografía por Emisión de Positrones (PET) y detección por coincidencia
- Reconstrucción de imágenes: Retroproyección filtrada y métodos de Fourier

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el número CT (unidades Hounsfield) para hueso y tejido blando [Usar]
2. Contrastar la apariencia de un tumor en imágenes de MRI ponderadas en T1 frente a T2 [Evaluar]
3. Determinar la resolución axial de una sonda de ultrasonido basada en la duración del pulso [Usar]

5.11.5. BMP/Radioterapia y Dosimetría (Core Tier1: 1 hr, Core Tier2: 1 hr)

Interacción de la radiación ionizante con el tejido para fines terapéuticos.

Temas:

Core

- Efecto fotoeléctrico, dispersión de Compton y producción de pares en el tejido
- Definiciones de Dosis Absorbida (Gray), Dosis Equivalente (Sievert) y KERMA
- Poder de frenado y el pico de Bragg en protonterapia
- Radiobiología: El modelo lineal-cuadrático para la supervivencia celular
- Principios de la Radioterapia de Intensidad Modulada (IMRT)

- Protocolos de dosimetría y calibración de cámaras de ionización

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la dosis absorbida a una profundidad específica utilizando una curva de Dosis en Profundidad Porcentual (PDD) [Usar]
2. Comparar la distribución de dosis de un haz de rayos X de 6MV frente a un haz de protones de 200 MeV [Evaluar]
3. Aplicar el modelo lineal-cuadrático para estimar el efecto del fraccionamiento de la dosis [Usar]

5.11.6. BMP/Mecánica Estadística de Redes Biológicas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Comportamiento colectivo en la regulación genética, el metabolismo e interacciones proteicas.
Temas:
Core

- Dinámica de redes de regulación genética: La ecuación de Hill y cooperatividad
- Modelado basado en restricciones y Análisis de Flujo Metabólico (MFA)
- Algoritmo de Gillespie para la expresión genética estocástica
- Identificación de motivos de red: Bucles de retroalimentación y alimentación directa
- Teoría de la información en la señalización biológica: Información mutua y capacidad
- Adaptación perfecta y robustez en la señalización por quimiotaxis

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Modelar un interruptor genético (toggle switch) utilizando un par de ecuaciones diferenciales acopladas [Usar]
2. Analizar cómo los bucles de retroalimentación negativa suprimen el ruido estocástico en la expresión genética [Evaluar]
3. Calcular la capacidad de información de un sensor bioquímico simple [Usar]

5.11.7. BMP/Microfluídica y Lab-on-a-Chip (Core Tier1: 1 hr, Core Tier2: 1 hr)

Física de fluidos a escala micrométrica para el análisis biológico.
Temas:
Core

- Flujo a bajo número de Reynolds y la reversibilidad del flujo de Stokes
- Tensión superficial, ecuación de Young-Laplace y acción capilar
- Flujo electroosmótico y potencial Zeta en microcanales
- Microfluídica basada en gotas y cribado de alto rendimiento
- Mezclado a microescala: Número de Peclet y límites difusivos
- Principios de Organ-on-a-Chip: Imitación de microambientes fisiológicos

Aprendizaje esperado (*Learning Outcomes*):
Core:

1. Calcular la altura de ascenso capilar de la sangre en un tubo de vidrio de 50 micrómetros [Usar]
2. Evaluar el tiempo requerido para que una proteína se difunda a través de un canal microfluídico [Evaluar]
3. Diseñar una geometría de micro-mezclador para optimizar la eficiencia de mezclado a $Re < 1$ [Usar]

5.12. Astrofísica y Ciencias del Espacio (ASS)

La Astrofísica y las Ciencias del Espacio aplican las leyes de la física al estudio del universo. Este área abarca el nacimiento, vida y muerte de las estrellas, la formación de sistemas planetarios, la estructura de las galaxias y los fenómenos de alta energía.

Área de Conocimiento (KA)	Core Tier1	Core Tier2
⭐ 5.12.1 Estructura y Evolución Estelar (Pág. 274)	1	1
⭐ 5.12.2 Ciencia Planetaria y Exoplanetas (Pág. 274)	1	1
⭐ 5.12.3 Dinámica y Estructura Galáctica (Pág. 274)	1	1
⭐ 5.12.4 Astrofísica de Altas Energías (Rayos X, Rayos Gamma) (Pág. 275)	1	1
⭐ 5.12.5 Técnicas Observacionales y Telescopios (Pág. 275)	1	1
⭐ 5.12.6 Física Solar y Clima Espacial (Pág. 275)	1	1
⭐ 5.12.7 Astronomía de Multimensajeros (Ondas Gravitacionales + Neutrinos) (Pág. 276)	1	1

Cuadro 5.12: Lista de KUs del área de Astrofísica y Ciencias del Espacio.

5.12.1. ASS/Estructura y Evolución Estelar (Core Tier1: 1 hr, Core Tier2: 1 hr)

Los procesos físicos que rigen los ciclos de vida de las estrellas.

Temas:

Core

- Equilibrio hidrostático y el Teorema del Virial para estrellas
- Nucleosíntesis estelar: cadena pp, ciclo CNO y proceso triple alfa
- El diagrama de Hertzsprung-Russell (H-R) y clasificación estelar
- Transporte de energía: Transferencia radiativa y zonas de convección
- Presión de degeneración de electrones/neutrones y los límites de Chandrasekhar y TOV
- Evolución post-secuencia principal: Gigantes rojas, enanas blancas y supernovas

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Estimar el tiempo de vida en la secuencia principal de una estrella basado en su masa y luminosidad [Usar]
2. Analizar por qué las estrellas de alta masa tienen núcleos convectivos mientras que las de baja masa tienen núcleos radiativos [Evaluar]
3. Determinar la masa máxima de una enana blanca no rotatoria [Usar]

5.12.2. ASS/Ciencia Planetaria y Exoplanetas (Core Tier1: 1 hr, Core Tier2: 1 hr)

Formación y características de los planetas dentro y fuera de nuestro Sistema Solar.

Temas:

Core

- La Hipótesis Nebular: Acreción y formación de planetesimales
- Detección de exoplanetas: Fotometría de tránsito y análisis de curvas de luz
- Método de velocidad radial: Desplazamientos Doppler y el movimiento reflejo de las estrellas
- Composición atmosférica, efecto invernadero y la Zona Habitable
- Interacciones de marea: Migración orbital y rotación síncrona
- Imágenes directas de alto contraste e interferometría de anulación

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular el radio de un exoplanet dada la profundidad del tránsito de su estrella anfitriona [Usar]
2. Evaluar si un planeta a 0.5 UA de una estrella tipo K se encuentra en la zona habitable [Evaluar]
3. Estimar la masa mínima ($M \sin i$) de un planeta a partir de la amplitud de su velocidad radial [Usar]

5.12.3. ASS/Dinámica y Estructura Galáctica (Core Tier1: 1 hr, Core Tier2: 1 hr)

La organización, cinemática y evolución de las galaxias.

Temas:

Core

- La secuencia de Hubble: Galaxias elípticas, espirales e irregulares
- Curvas de rotación galáctica y evidencia de la Materia Oscura

- Núcleos Galácticos Activos (AGN) y agujeros negros supermasivos
- Cinemática estelar y las ecuaciones de Jeans para el equilibrio galáctico
- Teoría de la estructura espiral: Ondas de densidad frente a formación estocástica
- Interacciones de galaxias, fusiones y formación jerárquica de estructuras

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la masa de materia oscura requerida para explicar una curva de rotación plana a 20 kpc [Usar]
2. Comparar las poblaciones estelares y el contenido de gas de las galaxias elípticas frente a las espirales [Evaluar]
3. Aplicar el Teorema del Virial para estimar la masa total de un cúmulo de galaxias [Usar]

5.12.4. ASS/Astrofísica de Altas Energías (Rayos X, Rayos Gamma) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Procesos físicos extremos y aceleradores cósmicos.

Temas:

Core

- Física de discos de acreción: Luminosidad de Eddington
- Emisión no térmica: Radiación de sincrotrón y de curvatura
- Dispersión Compton inversa y el efecto Sunyaev-Zel'dovich
- Brotes de rayos gamma (GRB): Hipernovas y fusiones de estrellas de neutrones
- Física de púlsares: Radiación de dipolo magnético y frenado por rotación
- Mecanismo de aceleración de Fermi en remanentes de supernova

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la luminosidad de Eddington para un agujero negro de 10 masas solares [Usar]
2. Analizar la evolución del periodo de un púlsar debido a la pérdida de energía por radiación de dipolo [Evaluar]
3. Determinar la energía máxima de un fotón dispersado por electrones ultra-relativistas [Usar]

5.12.5. ASS/Técnicas Observacionales y Telescopios (Core Tier1: 1 hr, Core Tier2: 1 hr)

Métodos para detectar y medir la radiación electromagnética del espacio.

Temas:

Core

- Telescopios refractores y reflectores: Apertura y poder de resolución
- Principios de los detectores CCD, eficiencia cuántica y fuentes de ruido
- Técnicas espectroscópicas: Dispersión, resolución y perfiles de línea
- Radioastronomía: Síntesis de apertura y técnicas de VLBI
- Óptica adaptativa: Sensor de frente de onda y espejos deformables
- Instrumentación para observatorios espaciales de rayos X y rayos gamma

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la relación señal-ruido para una estrella dado un tiempo de exposición y fondo de cielo [Usar]
2. Evaluar el límite de difracción teórico de un telescopio terrestre de 30 metros [Evaluar]
3. Diseñar una configuración de arreglo interferométrico para lograr una resolución angular específica a 1.4 GHz [Usar]

5.12.6. ASS/Física Solar y Clima Espacial (Core Tier1: 1 hr, Core Tier2: 1 hr)

La actividad del Sol y su impacto en la heliosfera y la Tierra.

Temas:

Core

- Estructura solar: Núcleo, zona radiativa, zona convectiva y fotosfera
- El ciclo solar de 11 años y el modelo de dinamo de Babcock-Leighton
- El viento solar: Teoría de Parker de la expansión isotérmica
- Llamaradas solares y eyecciones de masa coronal (CME): Reconexión magnética
- Clima espacial: Tormentas geomagnéticas y perturbaciones ionosféricas
- Helioseismología: Uso de ondas acústicas para sondear el interior solar

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la constante solar (irradiancia) a la distancia de Marte [Usar]
2. Analizar cómo la reconexión magnética desencadena la liberación de energía en las llamaradas solares [Evaluar]
3. Modelar la velocidad del viento solar como función de la distancia radial al Sol [Usar]

5.12.7. ASS/Astronomía de Multimensajeros (Ondas Gravitacionales + Neutrinos) (Core Tier1: 1 hr, Core Tier2: 1 hr)

Integración de señales electromagnéticas, gravitacionales y de partículas de eventos cósmicos.

Temas:

Core

- Fuentes de ondas gravitacionales: Fusiones de sistemas binarios de agujeros negros y estrellas de neutrones
- Interferometría LIGO/Virgo: Configuración Michelson con cavidades Fabry-Perot
- Neutrinos cósmicos de alta energía y detección en IceCube
- Sirenas estándar: Uso de señales GW para medir la constante de Hubble
- La conexión física entre rayos cósmicos de ultra-alta energía y neutrinos

Electivo

- Detección de GW en el espacio: LISA y arreglos de cronometría de púlsares

Aprendizaje esperado (*Learning Outcomes*):

Core:

1. Calcular la masa de chirp de un sistema binario a partir de la evolución de su frecuencia de onda gravitacional [Usar]
2. Evaluar el flujo de neutrinos esperado de una supernova por colapso de núcleo cercana [Evaluar]
3. Estimar la distancia de luminosidad a una fusión de estrellas de neutrones binarias usando datos de deformación de GW [Usar]

Capítulo 6

Plan de estudios

6.1. Codificación de los cursos

Los cursos son clasificados en 4 niveles:

- **Introductorios** Código 1.. ejemplo: CS100
- **Intermedios** Código 2.. ejemplo: MA200
- **Avanzados** Código 3.. y ejemplo: IS370
- **Trabajo de final de carrera** Código 4.. ejemplo: CE403

Los cursos se encuentran codificados bajo el esquema que se muestra en la Figura 6.1.

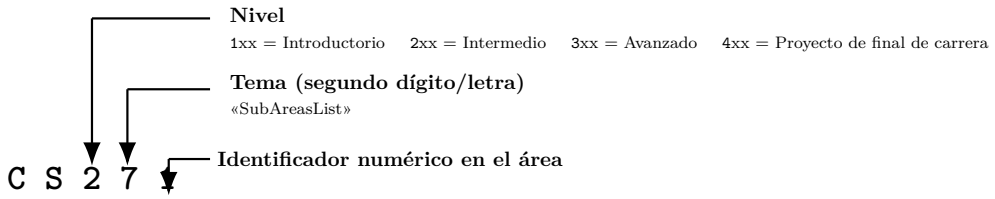


Figura 6.1: Esquema de codificación para los cursos.

El tipo de curso esta determinado por las letras iniciales. Los posibles códigos para estos tipos son:

1. **CS** Área de Ciencia de la Computación (*Computer Science* – CS);
2. **AI** Área de Inteligencia Artificial (*Artificial Intelligence* – AI);
3. **MA** Matemáticas (MA);
4. **STA** Estadística (STA);
5. **FI** Área de Física;
6. **QC** Computación Cuántica
7. **FG** Área de Formación General;
8. **ID** Área de Idiomas.

Al mismo tiempo, los cursos están clasificados en:

- Estudios generales
- Específicos
- Especialización

6.2. Estructura Curricular

La relación de cursos se muestra a continuación:

HT = Horas de teoría, HP = Horas de práctica, CR = Créditos, T = Tipo de curso.

O = Obligatorio, E = Electivo.

Primer Semestre							
Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
CS111	Introducción a la Programación (Pág 287)	Específicos	3	4	5	O	
MA100	Matemática Básica (Pág 292)	Estudios generales	3	4	5	O	
FG101	Lenguaje (Pág 295)	Estudios generales	1	2	2	O	
FG105	Género y Sociedad (Pág 298)	Estudios generales	1	2	2	O	
FG109	Ambiente y Desarrollo Sostenible (Pág 301)	Estudios generales	2	2	3	O	
FG112	Métodos de gestión personal universitaria (Pág 304)	Estudios generales	1	2	2	O	
FG140	Fundamentos de Investigación Científica (Pág 307)	Estudios generales	2	2	3	O	
					22		

Segundo Semestre							
Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
CS100	Introducción a la Ciencia de la Computación (Pág 310)	Específicos	2	2	3	O	
CS112	Programación Orientada a Objetos I (Pág 321)	Específicos	2	4	4	O	CS111 (1^{er} Sem)
AI161	IA Aplicada (Pág 326)	Específicos	2	4	4	O	CS111 (1^{er} Sem)
MA111	Cálculo I (Pág 331)	Estudios generales	3	4	5	O	MA100 (1^{er} Sem)
FI101	Física General (Pág 333)	Estudios generales	3	4	5	O	MA111 (2^{do} Sem)
					21		

Tercer Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
CS113	Programación Orientada a Objetos II (Pág 336)	Específicos	1	4	3	O	CS112 (2do Sem)
CS1D1	Estructuras Discretas I (Pág 342)	Específicos	2	4	4	O	MA100 (1er Sem)
CS2B1	Desarrollo Basado en Plataformas (Pág 345)	Específicos	2	4	4	O	CS112 (2do Sem)
MA112	Cálculo II (Pág 350)	Estudios generales	3	2	4	O	MA111 (2do Sem)
MA121	Algebra Lineal I (Pág 352)	Específicos	2	4	4	O	MA100 (1er Sem)
ID101	Inglés I (Pág 355)	Estudios generales	1	4	3	O	
					22		

Cuarto Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
CS1D2	Estructuras Discretas II (Pág 359)	Específicos	2	2	3	O	CS1D1 (3er Sem)
CS210	Algoritmos y Estructuras de Datos (Pág 361)	Específicos	2	4	4	O	CS113 (3er Sem)
CS271	Bases de Datos I (Pág 366)	Específicos	1	4	3	O	CS112 (2do Sem), CS1D1 (3er Sem)
MA122	Algebra Lineal II (Pág 371)	Específicos	2	4	4	O	MA121 (3er Sem)
MA210	Cálculo III (Pág 374)	Estudios generales	2	4	4	O	MA112 (3er Sem)
ID102	Inglés II (Pág 377)	Estudios generales	1	6	4	O	ID101 (3er Sem)
					22		

Quinto Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
CS251	Computación Gráfica (Pág 381)	Especialización	1	4	3	O	CS210 (4^{to} Sem)
MA211	Cálculo IV (Pág 384)	Específicos	1	4	3	O	MA210 (4^{to} Sem), MA121 (3^{er} Sem)
MA221	Análisis Numérico I[1] (Pág 387)	Específicos	1	4	3	O	MA112 (3^{er} Sem), MA122 (4^{to} Sem), AI161 (2^{do} Sem)
MA222	Análisis Real I (Pág 390)	Específicos	3	4	5	O	MA112 (3^{er} Sem), ID101 (3^{er} Sem)
MA251	Introducción a la Estadística (Pág 393)	Estudios generales	1	4	3	O	AI161 (2^{do} Sem)
MA281	Teoría de números (Pág 395)	Estudios generales	1	2	2	O	MA100 (1^{er} Sem)
ID201	Inglés III (Pág 399)	Estudios generales	1	4	3	O	ID102 (4^{to} Sem)
					22		

Sexto Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
MA212	Ecuaciones Diferenciales Ordinarias (Pág 402)	Específicos	4	4	6	O	CS112 (2^{do} Sem), MA121 (3^{er} Sem), MA222 (5^{to} Sem)
MA325	Análisis Numérico II (Pág 405)	Específicos	2	4	4	O	MA221 (5^{to} Sem)
MA351	Estadística y Probabilidades (Pág 408)	Específicos	2	4	4	O	MA251 (5^{to} Sem), CS112 (2^{do} Sem)
MA371	Análisis funcional I (Pág 414)	Específicos	3	4	5	O	MA222 (5^{to} Sem), ID201 (5^{to} Sem), MA122 (4^{to} Sem)
ID202	Inglés IV (Pág 417)	Estudios generales	1	4	3	O	ID201 (5^{to} Sem)
					22		

[1]Se recomienda que el alumno haya llevado: CS210 Algoritmos y Estructuras de Datos(4^{to} Sem).

Séptimo Semestre							
Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
CS261	Inteligencia Artificial (Pág 421)	Especialización	1	4	3	O	MA251 (5to Sem)
CS3P1	Computación Paralela y Distribuída (Pág 428)	Especialización	1	4	3	O	CS210 (4to Sem)
MA213	Ecuaciones Diferenciales Parciales (Pág 438)	Específicos	2	4	4	O	MA212 (6to Sem), MA211 (5to Sem)
MA326	Análisis Numérico III (Pág 441)	Específicos	2	4	4	O	MA325 (6to Sem)
MA361	Modelado Matemático (Pág 444)	Estudios generales	2	4	4	O	MA325 (6to Sem)
ID203	Inglés V (Pág 448)	Estudios generales	1	6	4	O	ID202 (6to Sem)
					22		

	Octavo Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	AI263	Introducción al Aprendizaje de Máquina (Pág 451)	Específicos	2	4	4	O	CS261 (7mo Sem)
	MA401	Proyecto de final de carrera I (Pág 454)	Especialización	1	4	3	O	MA361 (7mo Sem), MA371 (6to Sem), 140cr
Electivo	CS2S1	Sistemas Operativos (Pág 457)	Específicos	3	4	5	E	CS210 (4to Sem)
Electivo	CS3I1	Seguridad en Computación (Pág 462)	Específicos	3	4	5	E	CS210 (4to Sem)
	QC251	Mecánica Cuántica (Pág 468)	Especialización	4	2	5	E	FI101 (2do Sem), MA371 (6to Sem)
Electivo	CS253	Análisis exploratorio de datos (Pág 472)	Especialización	3	4	5	E	MA351 (6to Sem)
Electivo	CS370	Big Data y Analítica de Datos (Pág 476)	Especialización	3	4	5	E	CS271 (4to Sem)
Electivo	STA252	Estadística Inferencial (Pág 480)	Especialización	3	4	5	E	MA351 (6to Sem)
Electivo	MA327	Análisis Numérico IV (Pág 483)	Específicos	4	2	5	E	MA326 (7mo Sem), MA213 (7mo Sem)
Electivo	MA352	Procesos estocásticos (Pág 486)	Específicos	3	4	5	E	MA351 (6to Sem), MA210 (4to Sem), CS261 (7mo Sem)
Electivo	MA372	Análisis funcional II (Pág 489)	Específicos	3	4	5	E	MA371 (6to Sem)
						7		

	Noveno Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	AI264	Aprendizaje Profundo (Pág 492)	Específicos	3	4	5	O	AI263 (8^{vo} Sem)
	MA402	Proyecto de final de carrera II (Pág 495)	Especialización	1	8	5	O	MA401 (8^{vo} Sem), ID203 (7^{mo} Sem)
Electivo	CS231	Redes y Comunicación (Pág 499)	Específicos	2	4	4	E	CS2S1 (8^{vo} Sem)
	MA381	Estructuras Algebraicas (Pág 504)	Estudios generales	2	4	4	E	MA281 (5^{to} Sem)
Electivo	AI262	Procesamiento de Lenguaje Natural (Pág 508)	Específicos	2	4	4	E	AI263 (8^{vo} Sem)
	CS273	Análisis de series temporales (Pág 511)	Especialización	2	4	4	E	MA351 (6^{to} Sem)
Electivo	MA341	Optimización I (Pág 515)	Especialización	2	4	4	E	MA221 (5^{to} Sem)
	MA355	Teoría de Aprendizaje Estadístico (Pág 518)	Estudios generales	2	4	4	E	MA352 (8^{vo} Sem), MA372 (8^{vo} Sem)
						10		

	Décimo Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS400	Prácticas Pre-profesionales (Pág 521)	Especialización	1	4	3	O	110Cr
	AI268	Visión Computacional (Pág 523)	Especialización	3	4	5	O	AI264 (9^{no} Sem)
	AI367	Tópicos en Inteligencia Artificial (Pág 529)	Especialización	3	4	5	O	AI264 (9^{no} Sem)
	MA403	Proyecto de final de carrera III (Pág 536)	Especialización	1	8	5	O	MA402 (9^{no} Sem)
Electivo	QC381	Introduction a la Computación Cuántica (Pág 540)	Especialización	2	4	4	E	QC251 (8^{vo} Sem), MA381 (9^{no} Sem)
Electivo	CS254	Visualización de Datos Avanzada (Pág 544)	Especialización	2	4	4	E	CS253 (8^{vo} Sem)
						18		

Total de créditos de la carrera: 188 .

6.3. Distribución de cursos en la carrera

Esta propuesta puede ser analizada por el número de créditos dedicados a cada área, por niveles de cursos (Introductorios, Intermedios, Avanzados y Proyectos).

	Obligatorio			Electivo		
	Estudios generales	Específicos	Especialización	Estudios generales	Específicos	Especialización
1er Sem	17	5				
2do Sem	10	11				
3er Sem	7	15				
4to Sem	8	14				
5to Sem	8	11	3			
6to Sem	3	19				
7mo Sem	8	8	6			
8vo Sem		4	3		10	5
9no Sem		5	5		8	4
10mo Sem			18			8
Total	61	92	35	0	18	17

Cuadro 6.1: Distribución de créditos por área por semestre

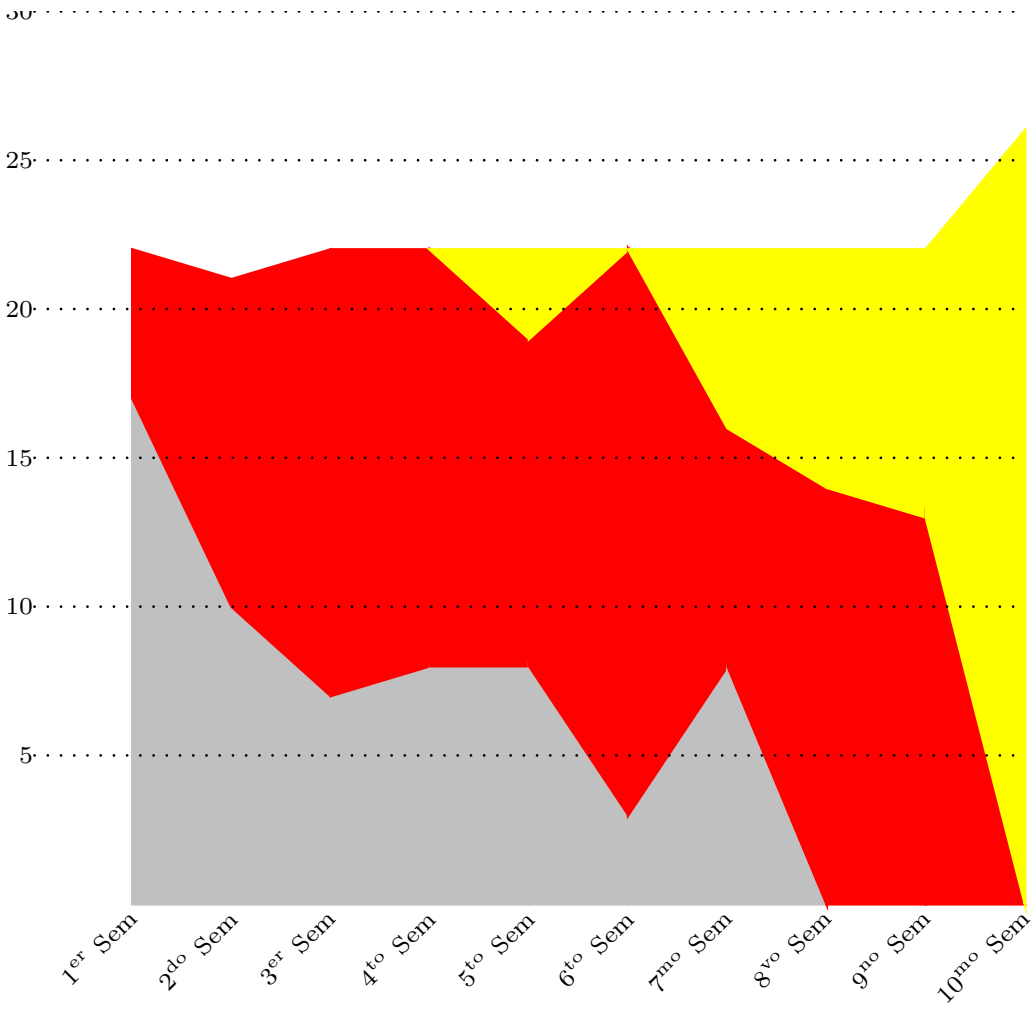


Figura 6.2: Créditos por área por semestre

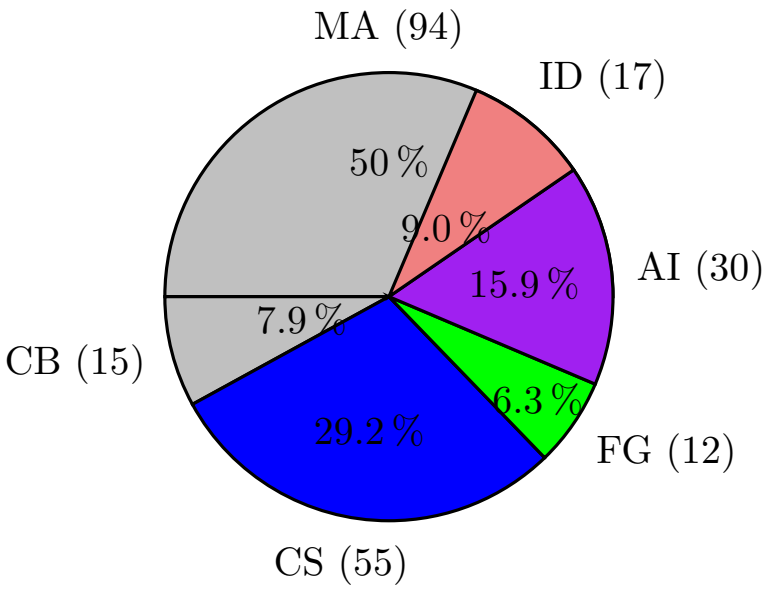


Figura 6.3: Distribución de cursos por áreas considerando creditaje.

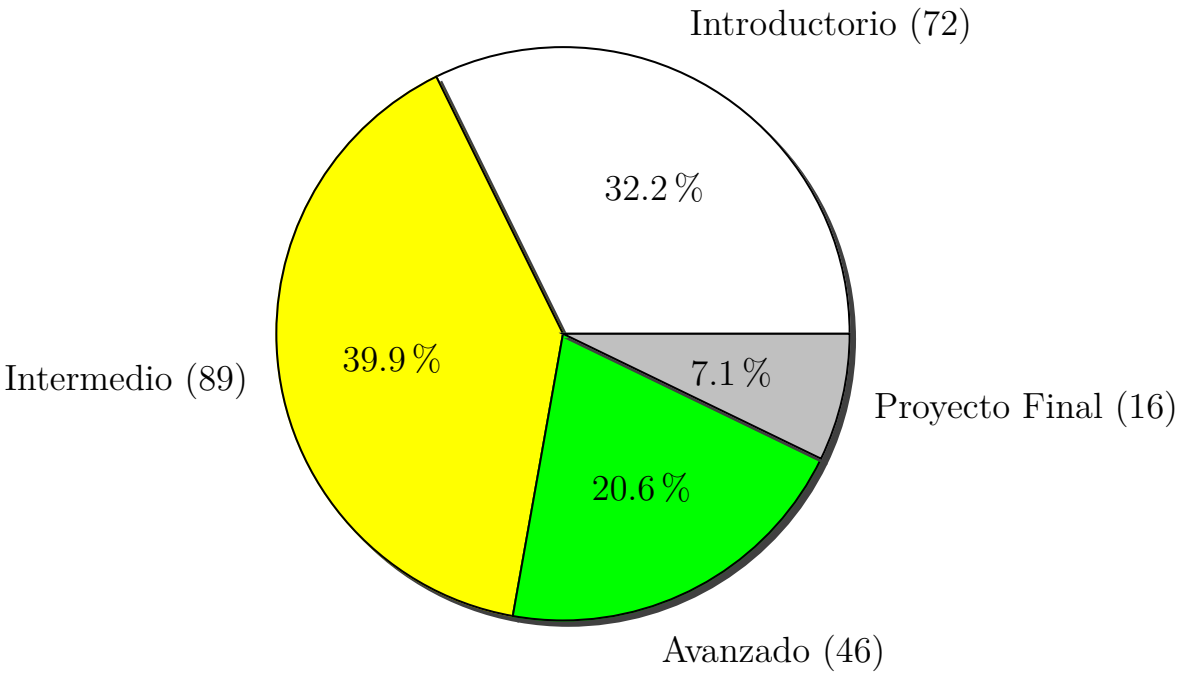


Figura 6.4: Distribución de créditos por niveles de cursos.

Referencias Bibliográficas

Capítulo 7

Contenido detallado por curso

Primer Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS111 Introducción a la Programación (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**: Ninguno

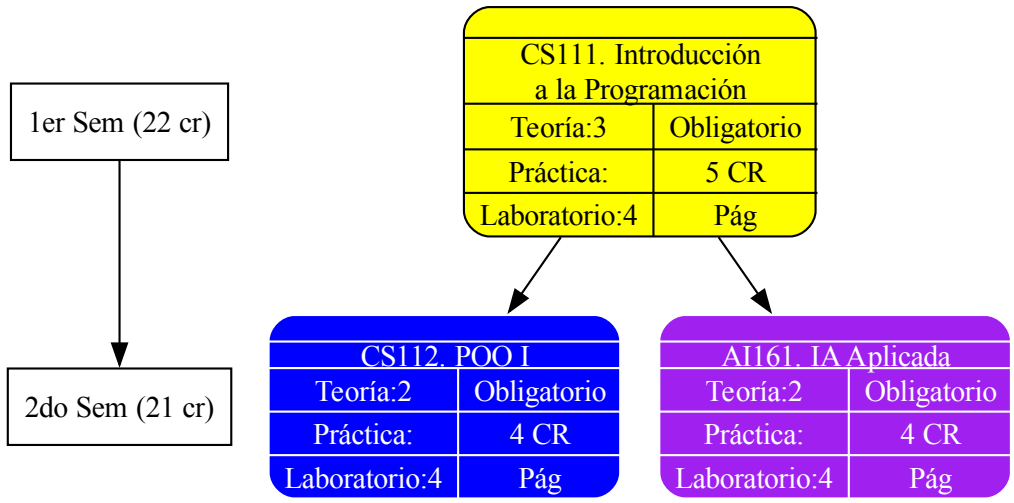


Figura 7.1: Mapa de Conexión. CS111 Introducción a la Programación

2. Fundamentación

Este es el primer curso de la secuencia de cursos introductorios a la Ciencia de la Computación. Este curso introduce a los participantes a los conceptos fundamentales de programación usando Python. Los temas incluyen tipos de datos, estructuras de control, funciones, listas, recursión, y la mecánica de

ejecución, prueba y depuración. La programación es uno de los pilares de la Ciencia de la Computación; cualquier profesional en el área necesitará programar para materializar sus modelos y propuestas.

3. Objetivos del curso.

- Introducir los conceptos fundamentales de programación.
- Desarrollar la capacidad de abstracción usando lenguaje de programación.
- Desarrollar habilidades de resolución de problemas mediante el pensamiento algorítmico.
- Comprender estructuras de datos básicas y sus aplicaciones.
- Dominar el lenguaje de programación Python para la resolución de problemas.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Introducción a la Programación y su Historia, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Breve historia de la computación y los lenguajes de programación. **II:** Visión general del lenguaje de programación Python. **III:** Entornos y herramientas de programación. **IV:** Estructura básica y ejecución de programas. **V:** Escritura, ejecución y depuración de programas simples.

Objetivos de Aprendizaje I: Identificar tendencias importantes en la historia de la computación. [Familiarizarse] **II:** Discutir el contexto histórico de los paradigmas de lenguajes de programación. [Familiarizarse] **III:** Comparar la vida diaria antes y después de las computadoras personales e Internet. [Evaluar] **IV:** Escribir, compilar y ejecutar un programa simple en Python. [Usar]

Bibliografía: [Zelle, 2010, Guttag, 2013, Brookshear and Brylow, 2019]

5.2 Fundamentos de Sistemas de Tipos, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Un tipo como un conjunto de valores junto con un conjunto de operaciones: *a*) Tipos primitivos (por ejemplo, números, booleanos) *b*) Tipos compuestos construidos a partir de otros tipos (por ejemplo, registros/estructuras, uniones, arreglos, listas, funciones, referencias usando operaciones de conjuntos) **II:** Seguridad de tipos y errores causados por usar valores inconsistentemente dados sus tipos pretendidos **III:** Objetivos y limitaciones del tipado estático y dinámico: detectar y eliminar errores lo antes posible. **IV:** Tipos de datos primitivos en Python (int, float, str, bool). **V:** Conversión de tipos y casting.

Objetivos de Aprendizaje I: Describir, tanto para un tipo primitivo como para uno compuesto, los valores que tienen ese tipo [Familiarizarse] **II:** Describir ejemplos de errores de programa detectados por un sistema de tipos [Familiarizarse] **III:** Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] **IV:** Explicar el sistema de tipado dinámico de Python. [Familiarizarse]

Bibliografía: [Zelle, 2010, Guttag, 2013]

5.3 Fundamentos de Programación Orientada a Objetos, 14 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Programación imperativa como subconjunto de la programación orientada a objetos. **II:** Definición de clases: campos, métodos y constructores. **III:** Propiedades dinámicas vs estáticas. **IV:** Variables, expresiones y asignaciones. **V:** Operaciones de entrada y salida. **VI:** Sentencias condicionales (if, elif, else). **VII:** Estructuras de iteración (bucles for, bucles while). **VIII:** Bucles anidados y sentencias de control de bucle. **IX:** Fundamentos del manejo de excepciones.

Objetivos de Aprendizaje **I:** Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Usar] **II:** Predecir y validar el flujo de control en un programa usando despacho dinámico [Evaluar] **III:** Usar mecanismos de encapsulación orientados a objetos como interfaces y miembros privados [Usar] **IV:** Implementar programas usando estructuras condicionales e iterativas. [Usar] **V:** Depurar programas con errores de sintaxis y lógicos. [Usar] **VI:** Diseñar e implementar algoritmos simples usando bucles. [Usar]

Bibliografía: [Zelle, 2010, Guttag, 2013, Brookshear and Brylow, 2019]

5.4 Funciones y paso de parámetros, 10 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Definición e invocación de funciones. **II:** Mecanismos de paso de parámetros (por valor, por referencia). **III:** Valores de retorno y múltiples retornos. **IV:** Alcance y tiempo de vida de variables (local, global, nonlocal). **V:** Fundamentos de recursión. **VI:** Expresiones lambda y funciones de orden superior. **VII:** Funciones como objetos de primera clase. **VIII:** Documentación y prueba de funciones.

Objetivos de Aprendizaje **I:** Definir e implementar funciones con paso de parámetros adecuado. [Usar] **II:** Explicar la diferencia entre paso por valor y paso por referencia. [Familiarizarse] **III:** Implementar soluciones recursivas a problemas simples. [Usar] **IV:** Usar expresiones lambda para construcciones de programación funcional. [Usar] **V:** Diseñar funciones con alcance y documentación apropiados. [Evaluar]

Bibliografía: [Zelle, 2010, Guttag, 2013, Hunt and Thomas, 1999]

5.5 Estructuras de datos básicas, 10 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Listas y operaciones con listas (indexación, segmentación, mutabilidad). **II:** Tuplas e inmutabilidad. **III:** Diccionarios y mapeo clave-valor. **IV:** Conjuntos y operaciones con conjuntos. **V:** Cadenas como secuencias. **VI:** Comprensión de listas y expresiones generadoras. **VII:** Algoritmos básicos de búsqueda y ordenamiento. **VIII:** Introducción a la complejidad algorítmica para operaciones básicas.

Objetivos de Aprendizaje **I:** Implementar programas usando listas, diccionarios y conjuntos. [Usar] **II:** Elegir estructuras de datos apropiadas para problemas dados. [Evaluar] **III:** Explicar las compensaciones entre diferentes estructuras de datos. [Familiarizarse] **IV:** Implementar operaciones básicas de búsqueda y ordenamiento en listas. [Usar]

Bibliografía: [Zelle, 2010, Guttag, 2013]

5.6 Fundamentos de Programación Orientada a Objetos, 10 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. **II:** Despacho dinámico: definición de llamada a método. **III:** Idiomas orientados a objetos para encapsulación: *a*) Privacidad, ocultación de datos y visibilidad de miembros de clase. *b*) Interfaces que revelan solo firmas de métodos. *c*) Clases base abstractas, rasgos (*traits*) y mixins. **IV:** Clases y objetos en Python. **V:** Atributos y métodos. **VI:** Fundamentos de herencia y polimorfismo. **VII:** Encapsulamiento y ocultación de información.

Objetivos de Aprendizaje **I:** Construir una jerarquía de clases simple utilizando subclases que permita reutilizar código para subclases distintas [Usar] **II:** Explicar la relación entre la herencia orientada a objetos (compartición de código y sobrescritura) y el subtipado (la idea de que un subtipo sea usable en un contexto que espera el supertipo) [Familiarizarse] **III:** Comparar y contrastar los beneficios y costos/impacto de usar herencia (subclases) y composición (específicamente, cómo basar la composición en funciones de orden superior) [Evaluar] **IV:** Diseñar e implementar clases simples en Python. [Usar] **V:** Explicar los beneficios de la POO para la organización del código. [Familiarizarse]

Bibliografía: [Zelle, 2010, Guttag, 2013]

5.7 Estrategias Algorítmicas, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Paradigmas: *a*) Fuerza Bruta (por ejemplo, búsqueda lineal, ordenamiento por selección, viajante de comercio, mochila) *b*) Disminuir y Vencer: 1) Por una Constante (por ejemplo, ordenamiento por inserción, ordenamiento topológico) 2) Por un Factor Constante (por ejemplo, búsqueda binaria) 3) Por un Tamaño Variable (por ejemplo, Euclides) *c*) Dividir y Vencer (por ejemplo, búsqueda binaria, quicksort, mergesort, Strassen) *d*) Voraz (por ejemplo, Dijkstra, Kruskal, Mochila) *e*) Transformar y Vencer: 1) Simplificación de instancia (por ejemplo, encontrar duplicados mediante preordenamiento de lista) 2) Cambio de representación (por ejemplo, heapsort) 3) Reducción de problema (por ejemplo, mínimo común múltiplo, programación lineal) 4) Programación dinámica (por ejemplo, Floyd, Marshall, Bellman-Ford) *f*) Compromisos espacio vs tiempo (por ejemplo, hash) **II:** Iteración vs recursión (por ejemplo, factorial, búsqueda en árbol) **III:** Estrategias de descomposición de problemas. **IV:** Técnicas de diseño de algoritmos. **V:** Estrategias de depuración y prueba. **VI:** Seguimiento de la ejecución del programa.

Objetivos de Aprendizaje **I:** Para cada uno de los paradigmas en esta unidad: *a*) Explicar sus características definitorias *b*) Explicar un ejemplo que demuestre el paradigma incluyendo cómo este ejemplo satisface las características del paradigma. [Familiarizarse] **II:** Dar ejemplos de algoritmos iterativos y recursivos que resuelvan el mismo problema, explicar los beneficios y desventajas de cada enfoque [Usar] **III:** Evaluar si un enfoque voraz conduce a una solución óptima [Evaluar] **IV:** Aplicar técnicas sistemáticas de depuración para localizar y corregir errores. [Usar] **V:** Analizar la complejidad algorítmica para problemas simples. [Familiarizarse]

Bibliografía: [Guttag, 2013, Cormen et al., 2009]

5.8 Métodos Modernos de Desarrollo y Uso de Herramientas, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Uso de entornos de programación modernos (IDEs, notebooks). **II:** Mejores prácticas de búsqueda y reutilización de código. **III:** Uso de bibliotecas estándar y APIs. **IV:** Técnicas de depuración y prueba. **V:** Fundamentos de control de versiones. **VI:** Documentación de código y guías de estilo. **VII:** Entornos virtuales y gestión de paquetes.

Objetivos de Aprendizaje **I:** Usar herramientas de desarrollo modernas efectivamente. [Usar] **II:** Aplicar principios de reutilización de código mediante bibliotecas y APIs. [Usar] **III:** Depurar y probar programas sistemáticamente. [Usar] **IV:** Seguir estándares de codificación y prácticas de documentación. [Familiarizarse] **V:** Gestionar paquetes y entornos de Python. [Familiarizarse]

Bibliografía: [Zelle, 2010, Hunt and Thomas, 1999]

6. Bibliografía

[Brookshear and Brylow, 2019] Brookshear, J. G. and Brylow, D. (2019). *Computer Science: An Overview*. Pearson, global edition edition.

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, 3rd edition.

[Guttag, 2013] Guttag, J. V. (2013). *Introduction to Computation and Programming Using Python*. MIT Press, 1st edition.

[Hunt and Thomas, 1999] Hunt, A. and Thomas, D. (1999). *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley, 1st edition.

[Zelle, 2010] Zelle, J. M. (2010). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2nd edition.

[Brookshear and Brylow, 2019] Brookshear, J. G. and Brylow, D. (2019). *Computer Science: An Overview*. Pearson, global edition edition.

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, 3rd edition.

[Guttag, 2013] Guttag, J. V. (2013). *Introduction to Computation and Programming Using Python*. MIT Press, 1st edition.

[Hunt and Thomas, 1999] Hunt, A. and Thomas, D. (1999). *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley, 1st edition.

[Zelle, 2010] Zelle, J. M. (2010). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA100 Matemática Básica (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**: Ninguno

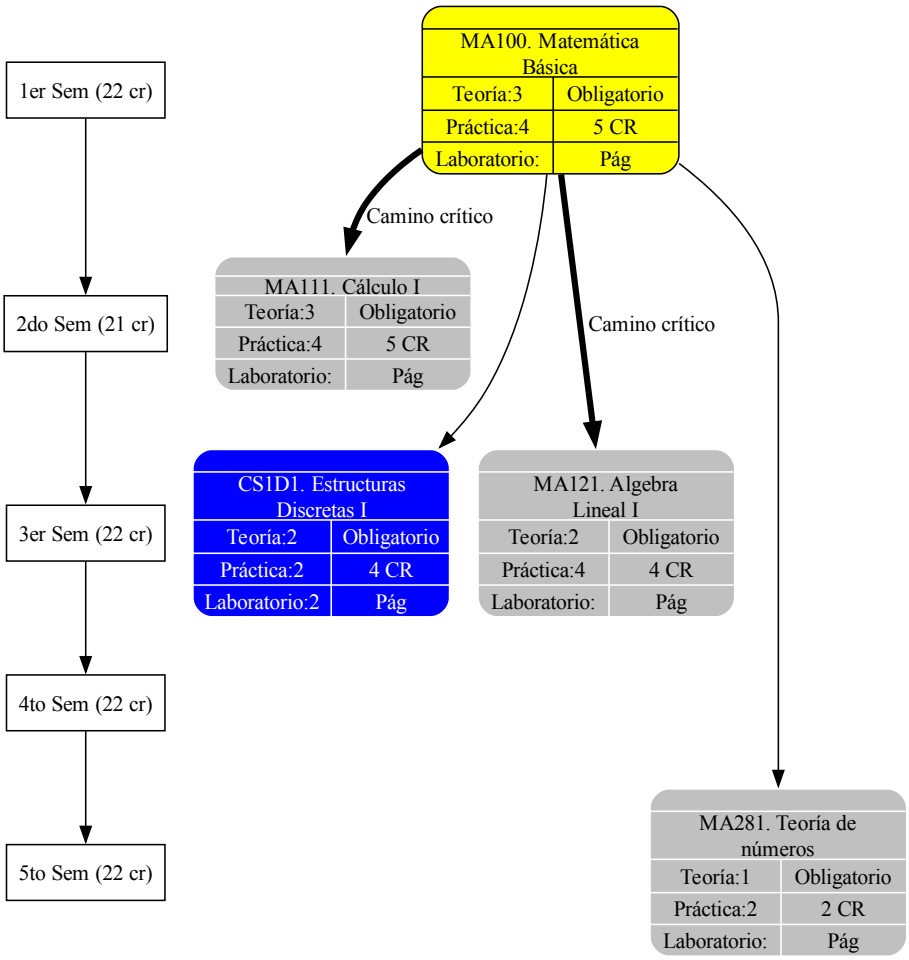


Figura 7.2: Mapa de Conexión. MA100 Matemática Básica

2. Fundamentación

El curso incluye conocimientos sobre Nociones de Lógica Proposicional y teoría de conjuntos, sistemas de los números Reales, números Enteros, Principio de Inducción Matemática, polinomios de una variable. Matrices, determinantes y sus aplicaciones en problemas que involucren a los seres vivos. Introducción a la estadística.

3. Objetivos del curso.

- Desarrollar habilidades de razonamiento lógico y pensamiento crítico mediante el uso de lenguajes formales.

- Comprender y aplicar las propiedades de los sistemas numéricos (reales y complejos) en la solución de problemas.
- Manipular estructuras polinomiales y matriciales para resolver sistemas de ecuaciones e interpretar resultados.
- Introducir conceptos básicos de estadística y probabilidad para el análisis de datos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

5. Contenido

5.1 Lógica y Conjuntos, 16 hr(s)

Competencias: ABET-1

Tópicos: I: Lógica proposicional: Conectivos, tablas de verdad y validez. **II:** Cuantificadores lógico-matemáticos. **III:** Teoría de conjuntos: Operaciones, relaciones y funciones. **IV:** Familias de conjuntos y producto cartesiano.

Objetivos de Aprendizaje I: Formalizar enunciados del lenguaje natural al lenguaje proposicional [Familiarizarse]. **II:** Operar con conjuntos utilizando leyes algebraicas y diagramas de Venn [Usar]. **III:** Demostrar propiedades básicas de conjuntos y relaciones lógicas [Evaluar].

Bibliografía: [Lipschutz, 1991, Rojo, 1995]

5.2 Sistemas Numéricos, 16 hr(s)

Competencias: ABET-1

Tópicos: I: El sistema de los números reales: Axiomas de cuerpo y orden. **II:** Inducción matemática y el binomio de Newton. **III:** Números complejos: Forma binómica, polar y exponencial. **IV:** Raíces de números complejos y teorema de De Moivre.

Objetivos de Aprendizaje I: Identificar la estructura algebraica de los números reales y complejos [Familiarizarse]. **II:** Aplicar el principio de inducción matemática en demostraciones de sumatorias [Usar]. **III:** Resolver ecuaciones en el campo de los números complejos utilizando diversas representaciones [Evaluar].

Bibliografía: [Gentile, 1985, Álvaro Pinzón, 1973]

5.3 Polinomios y Matrices, 16 hr(s)

Competencias: ABET-1

Tópicos: I: Polinomios: Algoritmo de la división y raíces de polinomios. **II:** Matrices: Tipos, operaciones y determinantes. **III:** Matriz inversa y rango de una matriz. **IV:** Sistemas de ecuaciones lineales: Método de Gauss-Jordan.

Objetivos de Aprendizaje I: Reconocer la estructura de los anillos de polinomios y el álgebra matricial [Familiarizarse]. **II:** Utilizar operaciones elementales de fila para calcular la inversa de una matriz [Usar]. **III:** Modelar y resolver sistemas de ecuaciones lineales aplicados a problemas de contexto [Evaluar].

Bibliografía: [Gentile, 1984, Rojo, 1995]

5.4 Estadística y Probabilidad, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Estadística descriptiva: Medidas de tendencia central. **II:** Medidas de dispersión y representación gráfica. **III:** Conceptos de probabilidad: Espacio muestral y eventos. **IV:** Probabilidad condicional e independencia.

Objetivos de Aprendizaje **I:** Describir conjuntos de datos mediante indicadores estadísticos básicos [Familiarizarse]. **II:** Calcular probabilidades de eventos simples y compuestos en experimentos aleatorios [Usar]. **III:** Interpretar resultados estadísticos para la toma de decisiones en entornos científicos [Evaluar].

Bibliografía: [Peterson, 2005]

6. Bibliografía

[Gentile, 1984] Gentile, E. (1984). *Notas de álgebra*. Editorial Universitaria de Buenos Aires.

[Gentile, 1985] Gentile, E. (1985). *Aritmética Elemental*. Colección de Monografías Científicas Ser.

[Lipschutz, 1991] Lipschutz, S. (1991). *Teoría de conjuntos y temas afines*. McGraw-Hill.

[Peterson, 2005] Peterson, J. (2005). *Matemática básica*. Prentice-Hall.

[Rojo, 1995] Rojo, A. (1995). *Álgebra I*. El Ateneo.

[Álvaro Pinzón, 1973] Álvaro Pinzón (1973). *Conjuntos y estructuras*. Blaisdell Publishing Company.

[Gentile, 1984] Gentile, E. (1984). *Notas de álgebra*. Editorial Universitaria de Buenos Aires.

[Gentile, 1985] Gentile, E. (1985). *Aritmética Elemental*. Colección de Monografías Científicas Ser.

[Lipschutz, 1991] Lipschutz, S. (1991). *Teoría de conjuntos y temas afines*. McGraw-Hill.

[Peterson, 2005] Peterson, J. (2005). *Matemática básica*. Prentice-Hall.

[Rojo, 1995] Rojo, A. (1995). *Álgebra I*. El Ateneo.

[Álvaro Pinzón, 1973] Álvaro Pinzón (1973). *Conjuntos y estructuras*. Blaisdell Publishing Company.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FG101 Lenguaje (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

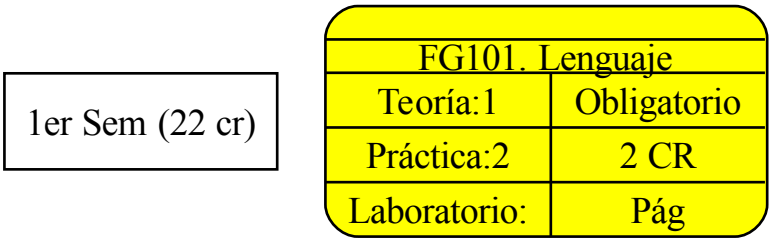


Figura 7.3: Mapa de Conexión. FG101 Lenguaje

2. Fundamentación

El curso de Lenguaje es fundamental para el desarrollo de las habilidades comunicativas del estudiante universitario. Se centra en el dominio de las reglas y principios que regulan el uso del español estándar, abarcando los niveles fonético, sintáctico, léxico y pragmático. Asimismo, busca fortalecer la capacidad de comprensión lectora, síntesis y redacción académica bajo normas internacionales, asegurando que el egresado pueda comunicar sus ideas con coherencia, ética y rigor académico.

3. Objetivos del curso.

- Conocer y aplicar las normas de escritura de la lengua española en la redacción y edición de textos diversos.
- Comprender los conceptos fundamentales de la comunicación humana y adaptar el mensaje a la realidad académica.
- Aplicar conocimientos básicos de investigación y redacción para elaborar informes académicos estructurados.
- Ejecutar alocuciones formales con responsabilidad, ética y domi- nio de las técnicas de expresión oral.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Usar)

5. Contenido

5.1 Lengua Española y Normativa, 16 hr(s)

Competencias: ABET-4

Tópicos: **I:** Lenguaje, lengua, habla y norma. **II:** Funciones del lenguaje y realidad lingüística del Perú. **III:** El acento escrito: Reglas generales y especiales. **IV:** La lectura: Estrategias de comprensión e ideas principales. **V:** Ortografía de expresiones numéricas.

Objetivos de Aprendizaje **I:** Reconocer los términos lingüísticos y las funciones del len- guaje en diversos contextos [Familiarizarse]. **II:** Utilizar correctamente las normas de ortografía y acentuación en la producción de textos [Usar]. **III:** Analizar la realidad multilingüe del Perú y caracterizar el lenguaje académico [Evaluar].

Bibliografía: [Española, 2010, Española, 2014]

5.2 La Redacción Académica, 16 hr(s)

Competencias: ABET-4

Tópicos: **I:** El informe académico: Planificación y estructura. **II:** Citas directas, indirectas y parafraseo según normas APA. **III:** La oración como unidad sintáctica: Sujeto y predicado. **IV:** Signos de puntuación: Punto, coma y punto y coma. **V:** Resumen de artículos científicos.

Objetivos de Aprendizaje **I:** Identificar la estructura sintáctica de oraciones simples y compuestas [Familiarizarse]. **II:** Emplear citas y referencias bibliográficas siguiendo la nor- mativa APA [Usar]. **III:** Elaborar resúmenes coherentes de artículos científicos y plani- ficar informes académicos [Evaluar].

Bibliografía: [Martínez, 2002, Nariola et al., 2007]

5.3 Redacción de Textos y Gramática, 16 hr(s)

Competencias: ABET-4

Tópicos: **I:** Casos de redundancia, coloquialismos y solecismos. **II:** Oración compuesta subordinada: Sustantiva, adjetiva y adverbial. **III:** Normas APA y Vancouver en la redacción científica. **IV:** Coherencia y cohesión textual: Dequeísmo y queísmo. **V:** Revisión y edición del informe académico.

Objetivos de Aprendizaje **I:** Detectar errores comunes de redacción como imprecisión léxica y vicios del lenguaje [Familiarizarse]. **II:** Organizar párrafos con cohesión y coherencia gramatical adecuada [Usar]. **III:** Corregir y coevaluar textos académicos para asegurar el uso estándar de la lengua [Evaluar].

Bibliografía: [Marín, 2008, Martínez, 2002]

5.4 Expresión Oral y Oratoria, 16 hr(s)

Competencias: ABET-4

Tópicos: **I:** El arte de hablar en público: La oratoria y el discurso. **II:** Características y tipos de oratoria. **III:** Elaboración de apoyos visuales (PPT) para exposiciones. **IV:** Responsabilidades éticas del orador. **V:** Presentación final del informe académico.

Objetivos de Aprendizaje **I:** Internalizar las técnicas fundamentales del discurso y la escucha activa [Familiarizarse]. **II:** Diseñar presentaciones visuales efectivas que complementen la comunicación oral [Usar]. **III:** Exponer informes académicos con fluidez, propiedad y ética ante una audiencia [Evaluar].

Bibliografía: [Verderber et al., 2010]

6. Bibliografía

[Española, 2010] Española, R. A. (2010). *Ortografía de la lengua española*. Espasa Libros.

[Española, 2014] Española, R. A. (2014). *Diccionario de la lengua española*. Espasa Libros, 23rd edition.

[Martínez, 2002] Martínez, M. C. (2002). *Lectura y escritura de textos. Perspectivas teóricas y talleres*. Universidad del Valle.

[Marín, 2008] Marín, M. (2008). *Lingüística y enseñanza de la lengua*. Aique Grupo Editor, 2nd edition.

[Nariola et al., 2007] Nariola, N., Naters, L., and Romero, L. (2007). *Iniciarse en la Investigación Universitaria: Exámenes, trabajos y reseñas*. Pontificia Universidad Católica del Perú, 1st edition.

[Verderber et al., 2010] Verderber, R. F., Verderber, K. S., and Sellnow, D. D. (2010). *Comunicación oral efectiva*. Cengage Learning Editores S.A., 14th edition.

[Española, 2010] Española, R. A. (2010). *Ortografía de la lengua española*. Espasa Libros.

[Española, 2014] Española, R. A. (2014). *Diccionario de la lengua española*. Espasa Libros, 23rd edition.

[Martínez, 2002] Martínez, M. C. (2002). *Lectura y escritura de textos. Perspectivas teóricas y talleres*. Universidad del Valle.

[Marín, 2008] Marín, M. (2008). *Lingüística y enseñanza de la lengua*. Aique Grupo Editor, 2nd edition.

[Nariola et al., 2007] Nariola, N., Naters, L., and Romero, L. (2007). *Iniciarse en la Investigación Universitaria: Exámenes, trabajos y reseñas*. Pontificia Universidad Católica del Perú, 1st edition.

[Verderber et al., 2010] Verderber, R. F., Verderber, K. S., and Sellnow, D. D. (2010). *Comunicación oral efectiva*. Cengage Learning Editores S.A., 14th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FG105 Género y Sociedad (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

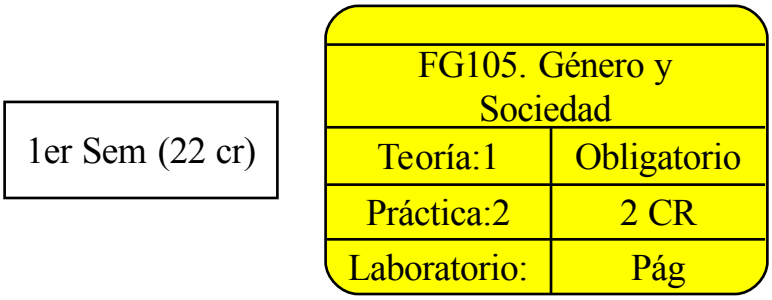


Figura 7.4: Mapa de Conexión. FG105 Género y Sociedad

2. Fundamentación

La asignatura de Género y Sociedad busca familiarizar al estudiante con la categoría de género como herramienta de análisis para comprender la organización social. El curso aborda de manera crítica las implicancias de la desigualdad y la discriminación en la producción de conocimiento y en la estructura sociohistórica. A través de este análisis, se promueve una visión reflexiva sobre el entorno, fomentando el respeto por la diversidad y la búsqueda de la equidad en el ejercicio de la ciudadanía y la práctica profesional.

3. Objetivos del curso.

- Comprender el concepto de género y su importancia para el análisis de las estructuras sociales y el pensamiento crítico.
- Identificar los procesos históricos de dominación y la formación de estereotipos de género en la sociedad occidental y peruana.
- Analizar la relación entre género, división del trabajo y el acceso a los espacios de poder público y privado.
- Evaluar las políticas de igualdad y el marco normativo nacional e internacional orientado a reducir las brechas de género.

4. Competencias Generales (*Outcomes*)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Familiarizarse)

5. Contenido

5.1 Conceptos Fundamentales de Género, 8 hr(s)

Competencias:

Tópicos: **I:** Sexo y género: Distinciones conceptuales y biológicas. **II:** El género como categoría analítica en las ciencias sociales. **III:** Socialización de género y construcción de identidades. **IV:** Estereotipos, prejuicios y roles de género en la actualidad.

Objetivos de Aprendizaje **I:** Distinguir las bases biológicas del sexo frente a la cons- trucción social y cultural del género [Familiarizarse]. **II:** Explicar cómo los estereotipos de género influyen en la percepción del entorno y en la comunicación [Usar]. **III:** Analizar críticamente la producción de conocimiento desde una perspectiva de género inclusiva [Evaluar].

Bibliografía: [Butler, 2007, Lamas, 1996]

5.2 Género, Historia y Estructura Social, 8 hr(s)

Competencias:

Tópicos: **I:** Evolución histórica de las relaciones de género. **II:** La división sexual del trabajo: Espacio público y privado. **III:** Género y colonialidad en América Latina. **IV:** Patriarcado y sistemas de dominación en la historia peruana.

Objetivos de Aprendizaje **I:** Identificar hitos históricos que han configurado las brechas de género en la sociedad latinoamericana [Familiarizarse]. **II:** Describir el impacto de la división del trabajo en el desa- rrollo económico y social de las mujeres [Usar]. **III:** Evaluar cómo las estructuras de poder históricas afectan la sostenibilidad y la equidad social actual [Evaluar].

Bibliografía: [Federici, 2010, Quijano, 1988]

5.3 Género, Poder y Ciudadanía, 8 hr(s)

Competencias:

Tópicos: **I:** Género y ejercicio de la ciudadanía en el siglo XXI. **II:** Participación política y acceso a puestos de toma de decisión. **III:** Violencia de género: Definiciones, tipos y estadísticas. **IV:** Masculinidades y cambios en las relaciones de poder.

Objetivos de Aprendizaje **I:** Reconocer las diversas formas de violencia de género y los mecanismos de prevención existentes [Familiarizarse]. **II:** Aplicar principios éticos en el análisis de las relaciones de poder entre géneros [Usar]. **III:** Argumentar sobre la importancia de la diversidad y la inclusión en el desarrollo de la sociedad civil [Evaluar].

Bibliografía: [Fuller, 2004, Pateman, 2016]

5.4 Políticas Públicas y Marco Normativo, 8 hr(s)

Competencias:

Tópicos: **I:** Marco jurídico internacional: La CEDAW y los ODS. **II:** Política Nacional de Igualdad de Género en el Perú. **III:** Enfoque de género en la gestión pública y proyectos sociales. **IV:** Desafíos actuales para la igualdad sustantiva.

Objetivos de Aprendizaje **I:** Identificar los principales tratados internacionales ratifi- cados por el Perú en materia de género [Familiarizarse]. **II:** Integrar el enfoque de género en la planificación y gestión de proyectos multidisciplinarios [Usar]. **III:** Evaluar el impacto de las políticas públicas de igualdad en la salud, seguridad y economía nacional [Evaluar].

Bibliografía: [Durand, 2014, MIMP,]

6. Bibliografía

[Butler, 2007] Butler, J. (2007). *El género en disputa: El feminismo y la subversión de la identidad*. Paidós.

[Durand, 2014] Durand, A. (2014). *Género y Ciudadanía: Retos para la inclusión en el Perú*. UNMSM.

[Federici, 2010] Federici, S. (2010). *Calibán y la bruja: Mujeres, cuerpo y acumulación originaria*. Traficantes de Sueños.

[Fuller, 2004] Fuller, N. (2004). *Jerarquías de género y ciudadanía*. Pontificia Universidad Católica del Perú.

[Lamas, 1996] Lamas, M. (1996). *El género: La construcción cultural de la diferencia sexual*. PUEG.

[MIMP,] MIMP. Política nacional de igualdad de género. Technical report, Ministerio de la Mujer y Poblaciones Vulnerables.

[Pateman, 2016] Pateman, C. (2016). *El contrato sexual*. Anthropos.

[Quijano, 1988] Quijano, A. (1988). *Modernidad, identidad y utopía en América Latina*. Sociedad y Política.

[Butler, 2007] Butler, J. (2007). *El género en disputa: El feminismo y la subversión de la identidad*. Paidós.

[Durand, 2014] Durand, A. (2014). *Género y Ciudadanía: Retos para la inclusión en el Perú*. UNMSM.

[Federici, 2010] Federici, S. (2010). *Calibán y la bruja: Mujeres, cuerpo y acumulación originaria*. Traficantes de Sueños.

[Fuller, 2004] Fuller, N. (2004). *Jerarquías de género y ciudadanía*. Pontificia Universidad Católica del Perú.

[Lamas, 1996] Lamas, M. (1996). *El género: La construcción cultural de la diferencia sexual*. PUEG.

[MIMP,] MIMP. Política nacional de igualdad de género. Technical report, Ministerio de la Mujer y Poblaciones Vulnerables.

[Pateman, 2016] Pateman, C. (2016). *El contrato sexual*. Anthropos.

[Quijano, 1988] Quijano, A. (1988). *Modernidad, identidad y utopía en América Latina*. Sociedad y Política.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FG109 Ambiente y Desarrollo Sostenible (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

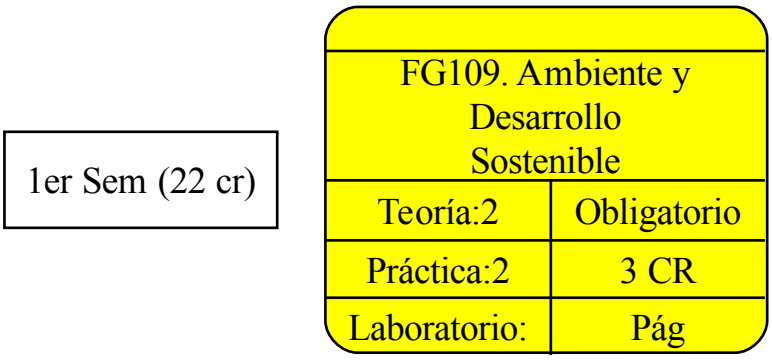


Figura 7.5: Mapa de Conexión. FG109 Ambiente y Desarrollo Sostenible

2. Fundamentación

La asignatura de Medio Ambiente y Desarrollo Sostenible es de naturaleza teórico-práctica y busca sensibilizar al estudiante sobre la importancia de la conservación de los ecosistemas. El curso aborda los principios de la ecología, el impacto de las actividades humanas en la biosfera y las estrategias globales para alcanzar un desarrollo que equilibre el crecimiento económico, la equidad social y la protección ambiental. Estos conocimientos son cruciales para que el profesional de computación diseñe soluciones tecnológicas con un enfoque de responsabilidad ambiental y sostenibilidad.

3. Objetivos del curso.

- Comprender los conceptos fundamentales de ecología y el funcionamiento de los ciclos biogeoquímicos.
- Analizar las causas y consecuencias de los principales problemas ambientales globales como el cambio climático.
- Evaluar el marco normativo nacional e internacional orientado a la gestión ambiental y el desarrollo sostenible.
- Aplicar indicadores de sostenibilidad en la evaluación de proyectos científicos y tecnológicos.

4. Competencias Generales (*Outcomes*)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Familiarizarse)

5. Contenido

5.1 Ecología y Biodiversidad, 16 hr(s)

Competencias: ABET-5

Tópicos: I: Ecología: Conceptos básicos, ecosistemas y biomas. **II:** Flujo de energía y ciclos biogeoquímicos en la naturaleza. **III:** Biodiversidad: Importancia, niveles y amenazas actuales. **IV:** Servicios ecosistémicos y valorización del capital natural.

Objetivos de Aprendizaje I: Identificar los componentes de un ecosistema y las relaciones de interdependencia entre los seres vivos [Familiarizarse]. **II:** Explicar el impacto de la pérdida de biodiversidad en la estabilidad de los sistemas naturales [Usar]. **III:** Evaluar la salud de un ecosistema local utilizando indicadores biológicos y parámetros fisicoquímicos [Evaluar].

Bibliografía: [Miller and Spoolman, 2012, Smith and Smith, 2001]

5.2 Impacto Ambiental y Cambio Climático, 16 hr(s)

Competencias: ABET-5

Tópicos: I: Contaminación ambiental: Aire, agua, suelo y residuos. **II:** El efecto invernadero y el calentamiento global. **III:** Impacto de la actividad industrial y tecnológica. **IV:** Huella ecológica y huella de carbono: Medición y reducción.

Objetivos de Aprendizaje I: Reconocer las principales fuentes de contaminación y sus efectos en la salud pública y el clima [Familiarizarse]. **II:** Calcular la huella de carbono personal e institucional para proponer medidas de mitigación [Usar]. **III:** Analizar críticamente el impacto ambiental de los centros de datos y la infraestructura tecnológica [Evaluar].

Bibliografía: [Miller and Spoolman, 2012, Starr and Taggart,]

5.3 Gestión Ambiental y Normativa, 16 hr(s)

Competencias: ABET-5

Tópicos: I: Marco jurídico ambiental en el Perú y el Ministerio del Ambiente. **II:** Evaluación de Impacto Ambiental (EIA) y certificaciones. **III:** Gestión de recursos naturales: Agua, energía y áreas protegidas. **IV:** Ética ambiental y responsabilidad social corporativa.

Objetivos de Aprendizaje I: Describir las funciones de los organismos reguladores del medio ambiente en el ámbito nacional [Familiarizarse]. **II:** Aplicar principios éticos y legales en el diseño de planes de gestión ambiental para proyectos [Usar]. **III:** Argumentar sobre la importancia de la transparencia y la participación ciudadana en temas ambientales [Evaluar].

Bibliografía: [Strange and Bayley, , Primack,]

5.4 Estrategias para el Desarrollo Sostenible, 16 hr(s)

Competencias: ABET-5

Tópicos: I: Desarrollo Sostenible: Evolución del concepto y los ODS. **II:** Economía circular y tecnologías limpias (Green IT). **III:** Energías renovables y eficiencia energética. **IV:** Educación ambiental y estilos de vida sostenibles.

Objetivos de Aprendizaje I: Reconocer los 17 Objetivos de Desarrollo Sostenible y su relación con la computación científica [Familiarizarse]. **II:** Integrar criterios de sostenibilidad y economía circular en el ciclo de vida del software [Usar]. **III:** Proponer soluciones tecnológicas innovadoras para resolver desafíos socioambientales contemporáneos [Evaluar].

Bibliografía: [Strange and Bayley, , Miller and Spoolman, 2012]

6. Bibliografía

[Miller and Spoolman, 2012] Miller, G. T. and Spoolman, S. (2012). *Living in the Environment*. Cengage Learning.

[Primack,] Primack, R. *Essentials of Conservation Biology*. Sinauer Associates Inc.

[Smith and Smith, 2001] Smith, R. L. and Smith, T. (2001). *Ecología*. Pearson Educación.

[Starr and Taggart,] Starr, C. and Taggart, R. *Biología: La unidad y diversidad de la vida*. Thomson Editors.

[Strange and Bayley,] Strange, T. and Bayley, A. *Desarrollo sostenible: Integrar la economía, la sociedad y el medio ambiente*. OCDE / UNAM.

[Miller and Spoolman, 2012] Miller, G. T. and Spoolman, S. (2012). *Living in the Environment*. Cengage Learning.

[Primack,] Primack, R. *Essentials of Conservation Biology*. Sinauer Associates Inc.

[Smith and Smith, 2001] Smith, R. L. and Smith, T. (2001). *Ecología*. Pearson Educación.

[Starr and Taggart,] Starr, C. and Taggart, R. *Biología: La unidad y diversidad de la vida*. Thomson Editors.

[Strange and Bayley,] Strange, T. and Bayley, A. *Desarrollo sostenible: Integrar la economía, la sociedad y el medio ambiente*. OCDE / UNAM.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FG112 Métodos de gestión personal universitaria (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

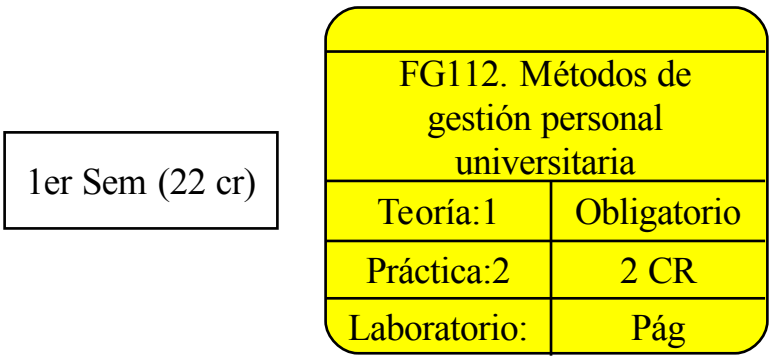


Figura 7.6: Mapa de Conexión. FG112 Métodos de gestión personal universitaria

2. Fundamentación

La asignatura de Métodos de Gestión Personal Universitaria es de naturaleza teórico-práctica y tiene como propósito empoderar al estudiante mediante el desarrollo de competencias intrapersonales y estrategias de trabajo intelectual. El curso integra procesos de autoconocimiento y planificación estratégica con técnicas avanzadas de aprendizaje y herramientas digitales. A través de este enfoque integral, el estudiante podrá gestionar su comportamiento ante el cambio, optimizar su rendimiento académico y realizar investigaciones con rigor ético, fomentando una visión estratégica para enfrentar los retos del entorno universitario y profesional contemporáneo.

3. Objetivos del curso.

- Desarrollar un plan estratégico personal integrando técnicas de aprendizaje autónomo y diagnóstico FODA.
- Aplicar estrategias cognitivas y metacognitivas para optimizar el procesamiento, retención y comunicación de la información científica.
- Potenciar habilidades de liderazgo, ética y trabajo colaborativo en entornos presenciales y virtuales.
- Elaborar trabajos de investigación monográfica y presentaciones efectivas siguiendo normas de integridad académica y uso de TICs.

4. Competencias Generales (*Outcomes*)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 Gestión Estratégica y Aprendizaje, 12 hr(s)

Competencias:

Tópicos: **I:** Conceptos básicos de gestión y el aprendizaje universitario. **II:** Autoconocimiento y diagnóstico personal (FODA). **III:** Visión, misión y valores personales. **IV:** Procesos cognitivos, atención y estrategias de aprendizaje. **V:** Metacognición: Aprender a aprender y autorregulación.

Objetivos de Aprendizaje **I:** Identificar los principios de gestión y factores cognitivos que intervienen en el éxito académico [Familiarizarse]. **II:** Elaborar una matriz de planificación estratégica orientada al desarrollo individual [Usar]. **III:** Evaluar el alineamiento entre los valores personales y los procesos de aprendizaje continuo [Evaluar].

Bibliografía: [Alcaide, , Bernardo and Méndez, , Fischman,]

5.2 Técnicas de Trabajo Intelectual y TICs, 12 hr(s)

Competencias:

Tópicos: **I:** Lectura comprensiva, subrayado y toma de apuntes. **II:** Organizadores visuales: Mapas conceptuales y mentales. **III:** Gestión del tiempo y productividad personal. **IV:** Entorno Personal de Aprendizaje (PLE) y herramientas Web 2.0. **V:** Trabajo colaborativo en entornos virtuales.

Objetivos de Aprendizaje **I:** Reconocer la utilidad de los organizadores visuales y herramientas TIC en el estudio [Familiarizarse]. **II:** Utilizar técnicas de síntesis y gestión del tiempo para optimizar el rendimiento [Usar]. **III:** Diseñar un entorno personal de aprendizaje eficiente integrando herramientas digitales y de productividad [Evaluar].

Bibliografía: [Pallarés, , Novak, , Lazzati,]

5.3 Liderazgo, Ética e Investigación, 12 hr(s)

Competencias:

Tópicos: **I:** Liderazgo: Estilos, teorías y manejo de conflictos. **II:** La monografía: Estructura, búsqueda y selección de fuentes. **III:** Uso ético de la información y prevención del plagio (Normas de citación). **IV:** Responsabilidad social y ética profesional en la gestión de proyectos.

Objetivos de Aprendizaje **I:** Comprender la estructura metodológica de una investigación y los principios del liderazgo efectivo [Familiarizarse]. **II:** Seleccionar fuentes de información confiables para la resolución de problemas de gestión [Usar]. **III:** Redactar documentos académicos respetando la propiedad intelectual y los principios éticos [Evaluar].

Bibliografía: [Cortina, , Montenegro and Núñez, , Sampieri and Mendoza,]

5.4 Comunicación y Toma de Decisiones, 12 hr(s)

Competencias:

Tópicos: **I:** El proceso de toma de decisiones: Modelos y riesgos. **II:** Comunicación académica inclusiva: Sincrónica y asincrónica. **III:** Exposición oral y preparación de apoyos visuales. **IV:** Inteligencia emocional y adaptación proactiva al cambio.

Objetivos de Aprendizaje **I:** Identificar las etapas críticas para la toma de decisiones y la comunicación efectiva [Familiarizarse]. **II:** Aplicar estrategias de comunicación asertiva para liderar y exponer resultados de proyectos [Usar]. **III:** Resolver situaciones conflictivas y tomar decisiones basadas en pensamiento crítico e inteligencia emocional [Evaluar].

Bibliografía: [Bernardo and Méndez, , Lazzati, , Zimbardo and Gerrig,]

6. Bibliografía

[Alcaide,] Alcaide, F. *Aprendiendo de los mejores: Tu desarrollo personal es tu destino.* Alienta Editorial.

[Bernardo and Méndez,] Bernardo, J. and Méndez, J. *Estrategias de aprendizaje para universitarios: Un enfoque práctico y digital.* Pirámide.

[Cortina,] Cortina, A. *Ética para líderes: La responsabilidad de gobernar.* Paidós.

[Fischman,] Fischman, D. *El camino del líder: Historias ancestrales y vivencias personales.* Planeta.

[Lazzati,] Lazzati, S. *Decisiones gerenciales: Anatomía y proceso.* Granica.

[Montenegro and Núñez,] Montenegro, M. and Núñez, R. *Guía metodológica para la investigación y elaboración de monografías.* Fondo Editorial PUCP.

[Novak,] Novak, J. *Aprender a aprender: 30 años después.* Morata.

[Pallarés,] Pallarés, E. *Estudiar con éxito: Técnicas y hábitos para aprender mejor.* Paraninfo.

[Sampieri and Mendoza,] Sampieri, R. H. and Mendoza, C. *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta.* McGraw-Hill.

[Zimbardo and Gerrig,] Zimbardo, P. and Gerrig, R. *Psicología y vida (Edición 19).* Pearson.

[Alcaide,] Alcaide, F. *Aprendiendo de los mejores: Tu desarrollo personal es tu destino.* Alienta Editorial.

[Bernardo and Méndez,] Bernardo, J. and Méndez, J. *Estrategias de aprendizaje para universitarios: Un enfoque práctico y digital.* Pirámide.

[Cortina,] Cortina, A. *Ética para líderes: La responsabilidad de gobernar.* Paidós.

[Fischman,] Fischman, D. *El camino del líder: Historias ancestrales y vivencias personales.* Planeta.

[Lazzati,] Lazzati, S. *Decisiones gerenciales: Anatomía y proceso.* Granica.

[Montenegro and Núñez,] Montenegro, M. and Núñez, R. *Guía metodológica para la investigación y elaboración de monografías.* Fondo Editorial PUCP.

[Novak,] Novak, J. *Aprender a aprender: 30 años después.* Morata.

[Pallarés,] Pallarés, E. *Estudiar con éxito: Técnicas y hábitos para aprender mejor.* Paraninfo.

[Sampieri and Mendoza,] Sampieri, R. H. and Mendoza, C. *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta.* McGraw-Hill.

[Zimbardo and Gerrig,] Zimbardo, P. and Gerrig, R. *Psicología y vida (Edición 19).* Pearson.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FG140 Fundamentos de Investigación Científica (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

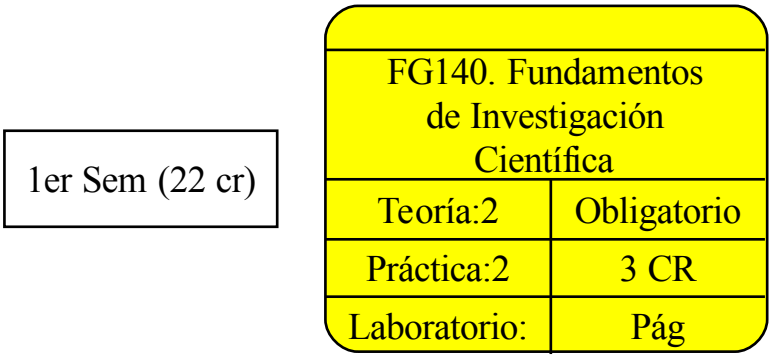


Figura 7.7: Mapa de Conexión. FG140 Fundamentos de Investigación Científica

2. Fundamentación

La asignatura de Fundamentos de Investigación Científica brinda al estudiante los conocimientos y herramientas básicas para el desarrollo de una investigación científica de calidad. El curso aborda la problematización de la realidad, la fundamentación teórica, la metodología y la ética en la investigación. Busca desarrollar en el estudiante la capacidad de observación crítica, el rigor académico y la honestidad intelectual necesarios para producir nuevos conocimientos que contribuyan al avance de la ciencia y la tecnología.

3. Objetivos del curso.

- Comprender el concepto de ciencia, el método científico y las etapas del proceso de investigación.
- Identificar y plantear problemas de investigación relevantes basándose en la observación de la realidad.
- Aplicar técnicas de búsqueda y gestión de información científica en bases de datos especializadas.
- Conocer las normas de redacción científica y los principios éticos que rigen la publicación de resultados.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Familiarizarse)

5. Contenido

5.1 Ciencia y Método Científico, 12 hr(s)

Competencias: ABET-3

Tópicos: I: Conocimiento científico: Características y niveles. **II:** La ciencia: Clasificación y objetivos. **III:** El método científico: Etapas y aplicación. **IV:** Ética en la investigación y buenas prácticas.

Objetivos de Aprendizaje I: Diferenciar el conocimiento científico del conocimiento empírico o cotidiano [Familiarizarse]. **II:** Aplicar de forma secuencial las etapas del método científico en casos prácticos [Usar]. **III:** Evaluar la importancia de la ética y la integridad científica en el desarrollo de proyectos [Evaluar].

Bibliografía: [Bunge, 2014, Sampieri et al., 2014]

5.2 Gestión de la Información Científica, 12 hr(s)

Competencias: ABET-3

Tópicos: I: Búsqueda de información: Estrategias y bases de datos. **II:** Herramientas de gestión bibliográfica (Mendeley, Zotero). **III:** Lectura crítica de artículos científicos. **IV:** Normas de citación y referencia (APA, Vancouver, IEEE).

Objetivos de Aprendizaje I: Identificar fuentes de información confiables y bases de datos académicas de alto impacto [Familiarizarse]. **II:** Utilizar gestores bibliográficos para organizar y citar las fuentes en un documento académico [Usar]. **III:** Redactar una revisión bibliográfica coherente y bien fundamentada sobre un tema específico [Evaluar]

Bibliografía: [Rodríguez, 2013, Upegui, 2011]

5.3 Planteamiento del Proyecto de Investigación, 12 hr(s)

Competencias: ABET-3

Tópicos: I: Identificación y delimitación del problema de investigación. **II:** Formulación de la pregunta de investigación y objetivos. **III:** Justificación y viabilidad de la investigación. **IV:** Formulación de hipótesis y definición de variables.

Objetivos de Aprendizaje I: Reconocer los componentes fundamentales de un problema de investigación científica [Familiarizarse]. **II:** Formular hipótesis matemáticas o lógicas que puedan ser contrastadas experimentalmente [Usar]. **III:** Diseñar el marco metodológico preliminar para un estudio científico riguroso [Evaluar].

Bibliografía: [Sampieri et al., 2014, Kothari, 2009]

5.4 Redacción y Comunicación Científica, 12 hr(s)

Competencias: ABET-3

Tópicos: I: Estructura del artículo científico (IMRyD). **II:** Redacción técnica: Claridad, precisión y brevedad. **III:** Tablas, figuras y representación de datos. **IV:** El proceso de publicación y revisión por pares.

Objetivos de Aprendizaje I: Comprender la estructura estándar de los reportes y comunicaciones científicas [Familiarizarse]. **II:** Elaborar informes efectivos utilizando lenguaje técnico y representaciones visuales adecuadas [Usar]. **III:** Presentar y defender oralmente los resultados de una investigación de forma inclusiva [Evaluar].

Bibliografía: [Olson, 2014, Russey et al., 2006]

6. Bibliografía

[Bunge, 2014] Bunge, M. (2014). *La ciencia, su método y su filosofía*. DeBolsillo.

[Kothari, 2009] Kothari, C. (2009). *Research Methodology: Methods and Techniques*. New Age International Publishers, 2nd edition.

[Olson, 2014] Olson, L. (2014). *Guide to Academic and Scientific Publication*. eacademia.

[Rodríguez, 2013] Rodríguez, L. J. M. (2013). *Cómo buscar y usar información científica: Guía para estudiantes universitarios*. Universidad de Cantabria.

[Russey et al., 2006] Russey, W., Ebel, H., and Bliefert, C. (2006). *How to Write a Successful Science Thesis*. Wiley-VCH.

[Sampieri et al., 2014] Sampieri, R. H., Collado, C. F., and Baptista, M. P. (2014). *Metodología de la investigación*. McGraw-Hill, 6th edition.

[Upegui, 2011] Upegui, A. A. S. (2011). *Manual de redacción académica e investigativa*. Católica del Norte Fundación Universitaria.

[Bunge, 2014] Bunge, M. (2014). *La ciencia, su método y su filosofía*. DeBolsillo.

[Kothari, 2009] Kothari, C. (2009). *Research Methodology: Methods and Techniques*. New Age International Publishers, 2nd edition.

[Olson, 2014] Olson, L. (2014). *Guide to Academic and Scientific Publication*. eacademia.

[Rodríguez, 2013] Rodríguez, L. J. M. (2013). *Cómo buscar y usar información científica: Guía para estudiantes universitarios*. Universidad de Cantabria.

[Russey et al., 2006] Russey, W., Ebel, H., and Bliefert, C. (2006). *How to Write a Successful Science Thesis*. Wiley-VCH.

[Sampieri et al., 2014] Sampieri, R. H., Collado, C. F., and Baptista, M. P. (2014). *Metodología de la investigación*. McGraw-Hill, 6th edition.

[Upegui, 2011] Upegui, A. A. S. (2011). *Manual de redacción académica e investigativa*. Católica del Norte Fundación Universitaria.

Segundo Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS100 Introducción a la Ciencia de la Computación (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

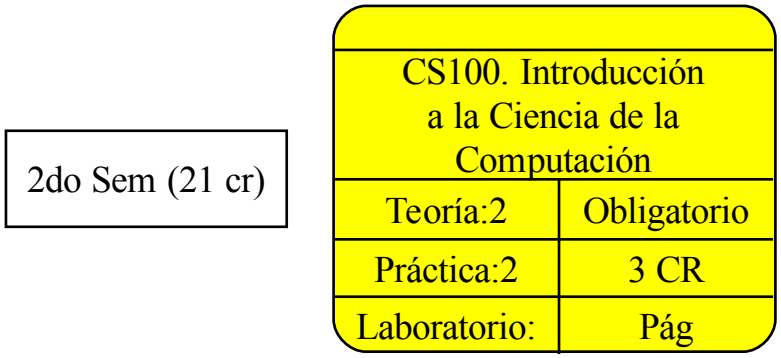


Figura 7.8: Mapa de Conexión. CS100 Introducción a la Ciencia de la Computación

2. Fundamentación

Este curso sirve como base para comprender los conceptos fundamentales del pensamiento computacional aplicables en diversas profesiones.

El curso proporciona, partiendo desde cero, una visión panorámica de: pensamiento computacional introductorio, almacenamiento de datos, arquitectura de computadoras, sistemas operativos, redes e Internet, algoritmos, métodos de ordenamiento, ingeniería de software, bases de datos, estructuras de datos, ingeniería de software, gráficos por computadora. Además, todos los conceptos son integrados con los nuevos conceptos de la inteligencia artificial.

Diseñado como un curso introductorio a la Ciencia de la Computación, los conceptos se presentan de manera lúdica y utilizando una metodología de Aprendizaje Activo. A lo largo del curso, se fomenta la participación activa de la audiencia, similar a una representación teatral.

Las áreas de conocimiento relacionadas cubiertas están directamente alineadas con los Planes de Estudio de Computación ACM/IEEE-CS.

El curso **no requiere** conocimientos previos en temas de manejo de computadoras y puede ser tomado por estudiantes de cualquier campo.

3. Objetivos del curso.

- Introducir los conceptos fundamentales del Pensamiento Computacional y la Ciencia de la Computación a estudiantes de cualquier formación profesional.
- Desarrollar su capacidad de abstracción.
- Comprender cómo se aplica el Pensamiento Computacional en cada una de sus profesiones.
- Aplicar conceptos avanzados de manera simplificada en cualquier carrera.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Familiarizarse)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Familiarizarse)

5. Contenido

5.1 Pensamiento Computacional (Parte I), 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Explicación del sistema de evaluación. **II:** Instrucciones generales del curso. **III:** Definición de Computación. **IV:** La Computación como un binomio Humano-Computadora. **V:** Distorsiones en la definición de computación. **VI:** La Computación como la automatización de la abstracción. **VII:** Computación e Ingeniería: similitudes y diferencias. **VIII:** Resolución de problemas algorítmicos. **IX:** Dinámica: Comprender la ejecución de un algoritmo a velocidad humana.

Objetivos de Aprendizaje **I:** Aplicar los conceptos fundamentales de la computación en situaciones de la vida real. [Usar] **II:** Identificar distorsiones de la Computación en situaciones de la vida real. [Usar] **III:** Identificar claramente al menos 3 contextos de uso de la palabra .[En]gineer.[en] inglés. [Evaluar] **IV:** Identificar las limitaciones humanas para resolver problemas computacionales. [Usar]

Bibliografía: [Brookshear and Brylow, 2019]

5.2 Pensamiento Computacional (Parte II), 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Descomposición (Dividir un problema en piezas más pequeñas). **II:** Abstracción (Enfocarse en lo importante). **III:** Reconocimiento de patrones (Identificar secuencias similares basadas en problemas previos). **IV:** Diseño de algoritmos.

Objetivos de Aprendizaje **I:** Aplicar el Pensamiento Computacional a problemas del mundo real. [Usar]

Bibliografía: [Brookshear and Brylow, 2019]

5.3 Conceptos Generales, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Numeración binaria vs. decimal. **II:** Representación de caracteres: la tabla ASCII. **III:** Búsqueda binaria. **IV:** Complejidad computacional de un algoritmo.

Objetivos de Aprendizaje **I:** Aplicar varios sistemas de numeración a problemas del mundo real. [Usar] **II:** Comprender la representación interna de caracteres en las tablas ASCII y UTF-8. [Familiarizarse] **III:** Aplicar la estrategia algorítmica de Divide y Vencerás. [Usar] **IV:** Determinar análisis básicos de complejidad algorítmica. [Usar]

Bibliografía: [Brookshear and Brylow, 2019]

5.4 Representación de Datos a Nivel de Máquina, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Visión general e historia de la arquitectura de computadoras **II:** Bits, bytes y palabras **III:** Representaciones sin signo, con signo y complemento a dos **IV:** Representación de datos numéricos y bases numéricas: *a*) Punto fijo *b*) Punto flotante **V:** Representación de datos no numéricos **VI:** Representación de registros, arreglos y tipos de datos UTF

Objetivos de Aprendizaje **I:** Discutir por qué todo en las computadoras son datos, incluyendo instrucciones [Debatir] **II:** Explicar cómo las representaciones numéricas de longitud fija pueden afectar la exactitud y precisión [Explicar] **III:** Describir cómo se almacenan los enteros negativos en representaciones de magnitud con signo y complemento a dos [Describir] **IV:** Discutir cómo diferentes formatos pueden representar datos numéricos [Debatir] **V:** Explicar la representación a nivel de bits de datos no numéricos, como caracteres, cadenas, registros y arreglos [Explicar] **VI:** Traducir datos numéricos de un formato a otro [Traducir] **VII:** Describir cómo un sumador único (sin detección de desbordamiento) puede manejar tanto entrada con signo (complemento a dos) como sin signo (binario) sin ”saber” qué formato está usando una entrada dada [Describir]

Bibliografía: [Brookshear and Brylow, 2019]

5.5 Abstracción y Representación de Programas, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Programas que toman (otros) programas como entrada, como intérpretes, compiladores, verificadores de tipos, generadores de documentación **II:** Componentes de un lenguaje: *a*) Definiciones de alfabetos, delimitadores, oraciones, sintaxis y semántica *b*) Sintaxis vs semántica **III:** Programa como un conjunto de oraciones significativas no ambiguas **IV:** Abstracciones básicas de programación: constantes, variables, declaraciones (incluyendo declaraciones anidadas), comando, expresión, asignación, selección, iteración definida e indefinida, iteradores, función, procedimiento, módulos, manejo de excepciones **V:** Tipos de variables: estáticas, locales, no locales (*nonlocal*), globales; necesidad y problemas con variables no locales y globales. **VI:** Reglas de ámbito (*scope*): estático vs dinámico; visibilidad de variables; efectos secundarios (*side-effects*). **VII:** Entorno (*environment*) vs almacén (*store*) y sus propiedades **VIII:** Abstracción de datos y de control **IX:** Mecanismos para intercambio de información entre unidades de programa como procedimientos, funciones y módulos: variables no locales, variables globales, paso de parámetros, importación-exportación entre módulos **X:** Representación de instrucciones de bajo nivel como instrucciones de máquina virtual, lenguaje ensamblador y representación binaria

Objetivos de Aprendizaje **I:** Ilustrar el ámbito de variables y visibilidad usando programas simples [Aplicar] **II:** Ilustrar diferentes tipos de paso de parámetros usando un lenguaje de programación pseudo simple [Aplicar] **III:** Explicar el efecto secundario usando variables globales y no locales y cómo corregir tales programas [Explicar] **IV:** Explicar cómo los programas que procesan otros programas tratan a los otros programas como sus datos de entrada [Explicar] **V:** Describir una gramática y un árbol de sintaxis abstracta para un lenguaje pequeño [Describir]

Bibliografía: [Brookshear and Brylow, 2019]

5.6 Fundamentos de Criptografía, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Basic concepts on Cryptography **II:** Public-keys, Private-keys

Objetivos de Aprendizaje **I:** Explicar el papel de la criptografía en el soporte de la seguridad y la privacidad [Explicar] **II:** Discutir los riesgos de inventar los propios métodos criptográficos [Debatir] **III:** Discutir la importancia de los números primos en criptografía y explicar su

uso en algoritmos criptográficos [Debatir] **IV:** Implementar y criptoanalizar cifrados clásicos [Implementar]

Bibliografía: [Brookshear and Brylow, 2019]

5.7 Gestión de Memoria, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas **II:** Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento **III:** Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas **IV:** Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas y marcos **V:** Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad *a*) Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes) **VI:** Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento) **VII:** Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia

Objetivos de Aprendizaje **I:** Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar] **II:** Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir] **III:** Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar] **IV:** Describir la razón y el uso de la memoria caché (rendimiento y proximidad, cómo las cachés complican el aislamiento y la abstracción de máquina virtual) [Describir]

Bibliografía: [Brookshear and Brylow, 2019]

5.8 Rol y Propósito de los Sistemas Operativos, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación. **II:** Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos) **III:** Influencias de seguridad, redes, multimedia, computación paralela y distribuida

Objetivos de Aprendizaje **I:** Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar] **II:** Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar] **III:** Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.9 Principios del Sistema Operativo, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Diseño y enfoques de software del sistema operativo (por ejemplo, monolítico, en capas, modular, micro-kernel, unikernel) **II:** Abstracciones, procesos y recursos **III:** La evolución del vínculo entre la arquitectura del hardware y las funciones del sistema operativo **IV:** Aprovechamiento de interrupciones desde el nivel de hardware: rutinas de servicio e implementaciones. Ver también: Organización de Máquina a Nivel de Ensamblador *a*) Interrupciones de temporizador

para implementar segmentos de tiempo *b*) Interrupciones de E/S para poner hilos bloqueados en espera sin sondeo

Objetivos de Aprendizaje I: Comprender cómo la aplicación de enfoques de diseño de software al diseño/implementación de sistemas operativos (por ejemplo, en capas, modular, etc.) afecta la robustez y mantenibilidad de un sistema operativo [Explicar] **II:** Categorizar llamadas al sistema por propósito [Categorizar] **III:** Aplicar técnicas de SO para aislamiento, protección y rendimiento a través de las funciones del SO (por ejemplo, similitudes de inanición en planificación de procesos, planificación de solicitudes de disco, semáforos, etc.) y más allá [Aplicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.10 Fundamentos de Redes y Comunicaciones, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Importancia de las redes en la informática contemporánea, y desafíos asociados. Ver también: Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad **II:** Organización de Internet (por ejemplo, usuarios, Proveedores de Servicios de Internet, sistemas autónomos, proveedores de contenido, redes de entrega de contenido) **III:** Capas y sus roles (aplicación, transporte, red, enlace de datos y física) **IV:** Elementos de red (por ejemplo, routers, switches, hubs, puntos de acceso y hosts) **V:** Conceptos básicos de colas (por ejemplo, relación con latencia, congestión, niveles de servicio, etc.)

Objetivos de Aprendizaje I: Articular la organización de Internet [Articular] **II:** Listar y definir la terminología de red apropiada [Listar/Enumerar] **III:** Describir la estructura en capas de una arquitectura de red típica [Describir]

Bibliografía: [Brookshear and Brylow, 2019]

5.11 Aplicaciones en Red, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Esquemas de denominación y direccionamiento (por ejemplo, DNS e Identificadores Uniformes de Recursos) **II:** Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación **III:** Diversidad de demandas de aplicaciones en red (por ejemplo, latencia, ancho de banda y tolerancia a pérdidas). Ver también: Comunicación, Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social, Equidad Digital y Accesibilidad, Impacto Social e IA **IV:** Cobertura de protocolos de capa de aplicación (por ejemplo, HTTP) **V:** Interacciones con APIs de TCP, UDP y Socket. Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades

Objetivos de Aprendizaje I: Definir los principios de denominación, direccionamiento y localización de recursos [Definir] **II:** Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar]

Bibliografía: [Brookshear and Brylow, 2019]

5.12 Marco de Análisis de Complejidad, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Marco de Análisis de Complejidad: *a*) Rendimiento de un algoritmo en el mejor caso, caso promedio y peor caso *b*) Mediciones empíricas y relativas (Orden de Crecimiento) *c*) Tamaño de entrada y operaciones primitivas *d*) Eficiencia de tiempo y espacio **II:** Mediciones empíricas de rendimiento **III:** Compromisos tiempo-espacio en algoritmos

Objetivos de Aprendizaje I: Preparar una presentación que explique a estudiantes de primer año los conceptos básicos de complejidad algorítmica incluyendo comportamiento de algoritmo en mejor caso, caso promedio y peor caso, notaciones Big-O, Omega y Theta, clases de complejidad, compromisos tiempo-espacio, medición empírica e impacto en problemas prácticos [Explicar] **II:** Para cada algoritmo en la unidad Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, explicar su clase de complejidad de tiempo de ejecución y por qué pertenece a esta clase [Explicar] **III:** Evaluar informalmente la clase de complejidad fundamental de algoritmos simples [Evaluar] **IV:** Desarrollar estudios empíricos para determinar y validar hipótesis sobre la complejidad de tiempo de ejecución de varios algoritmos ejecutando algoritmos con entradas de varios tamaños y comparando el rendimiento real con el análisis teórico [Crear] **V:** Explicar ejemplos que ilustren los compromisos tiempo-espacio de algoritmos [Explicar] **VI:** Explicar cómo el balance del árbol afecta la eficiencia de las operaciones de árbol de búsqueda binaria [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.13 Notación Asintótica y Clases de Complejidad, 1 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Análisis de complejidad asintótica (cotas promedio y peor caso): *a*) Notaciones formales Big-O, Big-Omega y Big-Theta *b*) Clases de Complejidad Fundamentales y Ejemplos/Problemas Representativos: 1) $O(1)$ Constante (por ejemplo, acceso a arreglo) 2) $O(\log_2 n)$ Logarítmica (por ejemplo, búsqueda binaria) 3) $O(n)$ Lineal (por ejemplo, búsqueda lineal) 4) $O(n \log_2 n)$ Log Lineal (por ejemplo, mergesort) 5) $O(n^2)$ Cuadrática (por ejemplo, ordenamiento por selección) 6) $O(n^c)$ Polinomial (por ejemplo, $O(n^3)$ eliminación gaussiana) 7) $O(2^n)$ Exponencial (por ejemplo, Mochila, Satisfactibilidad (SAT), Viajante de Comercio (TSP), todos los subconjuntos) 8) $O(n!)$ Factorial (por ejemplo, circuito hamiltoniano, todas las permutaciones)

Objetivos de Aprendizaje I: Usando ejemplos, explicar cada una de las clases de complejidad fundamentales en esta unidad [Explicar] **II:** Para cada clase de complejidad fundamental en esta unidad, explicar un algoritmo que demuestre la complejidad de tiempo de ejecución asociada [Explicar] **III:** Explicar a una audiencia no técnica la importancia de los algoritmos tratables versus intratables usando una explicación intuitiva de la complejidad Big-O [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.14 Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas, 1 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Notaciones Little-o, Little-Omega y Little Theta

Objetivos de Aprendizaje I: Dado un problema para programar para el cual puede haber varios enfoques algorítmicos, evaluarlos y determinar cuáles son factibles, y seleccionar uno que sea óptimo en implementación y comportamiento de tiempo de ejecución [Evaluar]

Bibliografía: [Brookshear and Brylow, 2019]

5.15 Estructuras de Datos Fundamentales, 3 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: *a*) Operaciones de diccionario (insertar, eliminar, encontrar) **II:** Arreglos: *a*) Numéricos vs no numéricos, cadenas de caracteres *b*) Unidimensionales (vector) vs multidimensionales (matriz) **III:** Registros/Estructuras/Tuplas y Objetos **IV:** Listas enlazadas (por razones históricas): *a*) Simples vs Dobles y Lineales vs Circulares **V:** Pilas **VI:** Colas y deques: *a*) Cola de prioridad basada en montículo **VII:** Tablas/mapas hash: *a*) Resolución de colisiones y complejidad (por ejemplo,

sondeo, encadenamiento, rehash) **VIII:** árboles: *a*) Binarios, n-arios y árboles de búsqueda *b*) Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo)

Objetivos de Aprendizaje I: Para cada ADT/Estructura de Datos en esta unidad: *a*) Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. *b*) Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Explicar] **II:** Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Explicar] **III:** Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.16 Algoritmos Fundamentales, 3 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Grafos (por ejemplo, [no]dirigidos, [a]cíclicos, [no]conexos y [no]ponderados): *a*) Representación de grafos: lista de adyacencia vs matriz **II:** Algoritmos de búsqueda: *a*) Complejidad $O(n)$ (por ejemplo, búsqueda lineal/secuencial en arreglo/lista) *b*) Complejidad $O(\log_2 n)$ (por ejemplo, búsqueda binaria) *c*) Complejidad $O(\log_b n)$ (por ejemplo, búsqueda en árbol no informada en profundidad/amplitud) **III:** Algoritmos de ordenamiento (por ejemplo, estables, inestables): *a*) Complejidad $O(n^2)$ (por ejemplo, inserción, selección) *b*) Complejidad $O(n \log n)$ (por ejemplo, quicksort, merge, timesort) **IV:** Algoritmos de grafos: *a*) Camino más corto (por ejemplo, Dijkstra, Floyd) *b*) árbol de expansión mínima (por ejemplo, Prim, Kruskal) **V:** Algoritmos de ordenamiento: *a*) Complejidad $O(n \log n)$ heapsort *b*) Pseudo $O(n)$ complejidad (por ejemplo, bucket, counting, radix) **VI:** Algoritmos de grafos: *a*) Clausura transitiva (por ejemplo, Warshall) *b*) Ordenamiento topológico

Objetivos de Aprendizaje I: Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar] **II:** Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar] **III:** Dados los requisitos para un problema, desarrollar múltiples soluciones usando varias estructuras de datos y algoritmos. Posteriormente, evaluar la idoneidad, fortalezas y debilidades seleccionando un enfoque que satisfaga mejor los requisitos [Crear] **IV:** Explicar factores más allá de la eficiencia computacional que influyen en la elección de algoritmos, como el tiempo de programación, la mantenibilidad y el uso de patrones específicos de la aplicación en los datos de entrada [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.17 Algoritmos Avanzados, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Algoritmos de criptografía (por ejemplo, SHA-256) **II:** Algoritmos paralelos

Objetivos de Aprendizaje I: Una apreciación de la computación cuántica y su aplicación a ciertos problemas [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.18 Conceptos Fundamentales de Sistemas de Bases de Datos, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Propósito y ventajas de los sistemas de bases de datos **II:** Componentes de los sistemas de bases de datos **III:** Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) **IV:** Gestión de transacciones **V:** Normalización **VI:** Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) **VII:** Bases de datos distribuidas/sistemas basados en la nube **VIII:** Uso de un lenguaje de consulta declarativo

Objetivos de Aprendizaje I: Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Analizar] **II:** Enumerar los componentes de un sistema de base de datos (relacional) [Enumerar] **III:** Defender el valor de la independencia de datos [Defender] **IV:** Componer una consulta simple de selección-proyección-unión en SQL [Componer] **V:** Describir las ventajas de eliminar datos duplicados repetidos [Describir] **VI:** Esbozar cómo MapReduce usa paralelismo para procesar datos eficientemente [Esquematizar/Esbozar]

Bibliografía: [Brookshear and Brylow, 2019]

5.19 Fundamentos de Programación Orientada a Objetos, 4 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Programación imperativa como subconjunto de la programación orientada a objetos. **II:** Diseño orientado a objetos: *a*) Descomposición en objetos que poseen estado y tienen comportamiento. *b*) Diseño de jerarquía de clases para modelado. **III:** Definición de clases: campos, métodos y constructores. **IV:** Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. **V:** Despacho dinámico: definición de llamada a método. **VI:** Idiomas orientados a objetos para encapsulación: *a*) Privacidad, ocultación de datos y visibilidad de miembros de clase. *b*) Interfaces que revelan solo firmas de métodos. *c*) Clases base abstractas, rasgos (*traits*) y mixins.

Objetivos de Aprendizaje I: Enumerar las diferencias entre los paradigmas de programación imperativa y orientada a objetos [Enumerar] **II:** Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Componer] **III:** Usar mecanismos de encapsulación orientados a objetos como interfaces y miembros privados [Usar]

Bibliografía: [Brookshear and Brylow, 2019]

5.20 Control de Versiones y CI/CD, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software *a*) Configuración en control de versiones, builds/configuración reproducibles. *b*) Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). *c*) Estrategias de fusión/rebase, cuando sean relevantes. **II:** Gestión de lanzamientos (*release management*). **III:** Automatización de procesos de software: *a*) Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. *b*) Integración Continua (CI): el uso de automatización y pruebas automatizadas para hacer una validación preliminar de que la revisión actual del tronco se construye y pasa las pruebas (básicas). *c*) Despliegue Continuo (CD): el uso de automatización para liberar de forma automática cada cambio que supera las pruebas hacia el entorno de producción, garantizando entregas frecuentes y confiables. *d*) Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. **IV:** Conceptos y mecanismos de integración de herramientas. Prácticas de Desarrollo de Software **V:** Uso de las facilidades de un IDE moderno: depuración, refactorización, búsqueda/indexación, asistentes de código basados en ML, etc. Prácticas de Desarrollo de Software

Objetivos de Aprendizaje I: Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] **II:** Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] **III:** Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] **IV:** Comprender el uso de sistemas de CI/CD como una fuente de verdad para el estado del código compartido del equipo (éxito en la construcción y pruebas) [Explicar] **V:** Demostrar la capacidad de usar herramientas de software para apoyar el desarrollo de un producto de software de tamaño mediano [Demostrar]

Bibliografía: [Brookshear and Brylow, 2019]

5.21 Ingeniería de Requisitos, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Describir requisitos funcionales usando, por ejemplo, casos de uso o historias de usuario. *a*) Usar al menos un método para documentar y estructurar requisitos funcionales. *b*) Comprender cómo el método apoya el diseño y la implementación. *c*) Fortalezas y debilidades de usar un enfoque específico. **II:** Obtención (*elicitation*) de requisitos. *a*) Fuentes de requisitos, por ejemplo, usuarios, administradores o personal de soporte. *b*) Métodos de recopilación de requisitos, por ejemplo, encuestas, entrevistas o análisis de comportamiento. **III:** Requisitos no funcionales, por ejemplo, seguridad, usabilidad o rendimiento, también llamados Atributos de Calidad. Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social **IV:** Identificación y gestión de riesgos, incluyendo consideraciones éticas en torno al producto propuesto. Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo **V:** Comunicar y/o formalizar especificaciones de requisitos. **VI:** Prototipado de una herramienta tanto para obtener como para validar/confirmar requisitos. **VII:** Evolución del producto: cuando cambian los requisitos, cómo entender qué efecto tiene eso y qué cambios deben realizarse. **VIII:** Estimación de esfuerzo: *a*) Aprender técnicas para estimar mejor el esfuerzo requerido para completar una tarea; *b*) Practicar la estimación y compararla con cuánto tiempo toman las tareas; *c*) La estimación de esfuerzo es bastante difícil, por lo que es probable que los estudiantes se equivoquen en muchos casos, pero ver el proceso desarrollarse con su propio trabajo es valioso.

Objetivos de Aprendizaje **I:** Comparar diferentes métodos de obtención de requisitos a lo largo de múltiples ejes [Comparar] **II:** Identificar diferencias entre dos métodos de describir requisitos funcionales (por ejemplo, entrevistas con clientes, estudios de usuarios) y las situaciones donde se preferiría cada uno [Analizar] **III:** Identificar qué comportamientos son requeridos, permitidos o prohibidos a partir de un conjunto dado de requisitos y una lista de comportamientos candidatos [Analizar] **IV:** Recopilar un conjunto de requisitos para un sistema de software simple [Analizar] **V:** Identificar áreas de un sistema de software que deben cambiarse, dada una descripción del sistema y un conjunto de nuevos requisitos a implementar [Analizar] **VI:** Identificar los requisitos funcionales y no funcionales en un conjunto de requisitos [Analizar] **VII:** Estimar el tiempo para completar un conjunto de tareas, luego comparar las estimaciones con el tiempo real tomado [Estimar] **VIII:** Determinar una secuencia de implementación para un conjunto de tareas, respetando las dependencias entre ellas, con el objetivo de retirar el riesgo lo antes posible [Determinar] **IX:** Escribir una especificación de requisitos para un sistema de software simple [Escribir]

Bibliografía: [Brookshear and Brylow, 2019]

5.22 Fundamentos de Aprendizaje Automático, 1 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: *a*) Aprendizaje supervisado: 1) Clasificación 2) Regresión *b*) Aprendizaje por refuerzo *c*) Aprendizaje no supervisado: 1) Agrupamiento **II:** Ideas fundamentales: *a*) Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. *b*) Fuentes de error e indecidibilidad en el aprendizaje automático **III:** Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: *a*) Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente

Objetivos de Aprendizaje **I:** Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar] **II:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] **III:** Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.23 Redes Neuronales y Ética, 1 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Redes neuronales básicas: *a*) Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos *b*) Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) **II:** ética para el Aprendizaje Automático: *a*) Enfocarse en datos reales, escenarios reales y estudios de caso *b*) Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas **III:** Aprendizaje profundo: *a*) Redes feed-forward profundas (solo intuición, sin matemáticas) *b*) Redes neuronales convolucionales (solo intuición, sin matemáticas) *c*) Visualización de representaciones de características aprendidas de redes profundas *d*) Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) **IV:** ética para el Aprendizaje Automático: *a*) Continuar enfocándose en datos reales, escenarios reales y estudios de caso *b*) Privacidad *c*) Equidad *d*) Propiedad intelectual *e*) Explicabilidad

Objetivos de Aprendizaje **I:** Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar] **II:** Visualizar el progreso del entrenamiento de una red neuronal a través de curvas de aprendizaje en un kit de herramientas establecido (por ejemplo, TensorBoard) y visualizar las características aprendidas de la red [Aplicar] **III:** Dada una aplicación real de aprendizaje automático, describir problemas éticos respecto a las elecciones de datos, pasos de preprocesamiento, selección de algoritmo y visualización/presentación de resultados [Aplicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.24 Conceptos Fundamentales de Gráficos y Técnicas Interactivas, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Descripción general del pipeline de gráficos por computadora y sistemas interactivos **II:** Conceptos básicos de renderizado: rasterización, trazado de rayos (*ray tracing*), modelos de sombreado **III:** Teoría del color y espacios de color **IV:** Sistemas de coordenadas y transformaciones **V:** Técnicas básicas de interacción: dispositivos de entrada, manejo de eventos **VI:** Conceptos avanzados de renderizado: iluminación global (*global illumination*), renderizado basado en física **VII:** Hardware gráfico y técnicas de aceleración **VIII:** Percepción humana y cognición visual

Objetivos de Aprendizaje **I:** Explicar las etapas básicas del pipeline de gráficos por computadora [Explicar] **II:** Describir técnicas básicas de renderizado como rasterización y trazado de rayos [Describir] **III:** Aplicar teoría del color para diseñar gráficos visualmente efectivos [Aplicar] **IV:** Implementar transformaciones de coordenadas para gráficos 2D y 3D [Implementar] **V:** Diseñar aplicaciones interactivas básicas utilizando dispositivos de entrada comunes y manejo de eventos [Diseñar] **VI:** Comparar técnicas avanzadas de renderizado como iluminación global y renderizado basado en física [Comparar] **VII:** Analizar el rol del hardware gráfico en acelerar tareas de renderizado [Analizar] **VIII:** Explicar cómo la percepción humana influye en el diseño de gráficos e interacción [Explicar]

Bibliografía: [Brookshear and Brylow, 2019]

5.25 Renderizado Aplicado y Técnicas, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: **I:** Trazado de rayos (*ray tracing*): estructuras de aceleración, muestreo (*sampling*), sombreado **II:** Mapeo de texturas (*texture mapping*) y filtrado

Objetivos de Aprendizaje I: Implementar algoritmos de rasterización para renderizado de primitivas [Implementar] **II:** Implementar trazado de rayos básico con estructuras de aceleración [Implementar]

Bibliografía: [Brookshear and Brylow, 2019]

5.26 Clase de cierre: ¿Cómo funciona un motor de búsqueda como Google?, 2 hr(s)

Competencias: ABET-1,ABET-2,ABET-5

Tópicos: I: Análisis del problema. **II:** El índice no crece linealmente con el tamaño de la información indexada. **III:** El tiempo de respuesta no depende del tamaño de la "base de datos". **IV:** El tiempo de respuesta no depende del número de ocurrencias encontradas. **V:** Combinar varias estructuras de datos para llegar a una solución. **VI:** Analizar la escalabilidad de la solución.

Objetivos de Aprendizaje I: Comprender los principios bajo los cuales se crea un motor de búsqueda [Usar] **II:** Aplicar correctamente estructuras de datos para resolver el problema [Usar] **III:** Aplicar conceptos relacionados con la complejidad algorítmica en un motor de búsqueda [Usar].

Bibliografía: [Brookshear and Brylow, 2019]

6. Bibliografía

[Brookshear and Brylow, 2019] Brookshear, J. G. and Brylow, D. (2019). *Computer Science: An Overview*. Pearson, global edition edition.

[Brookshear and Brylow, 2019] Brookshear, J. G. and Brylow, D. (2019). *Computer Science: An Overview*. Pearson, global edition edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS112 Programación Orientada a Objetos I (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS111 Introducción a la Programación (1er Sem-Pág 287)

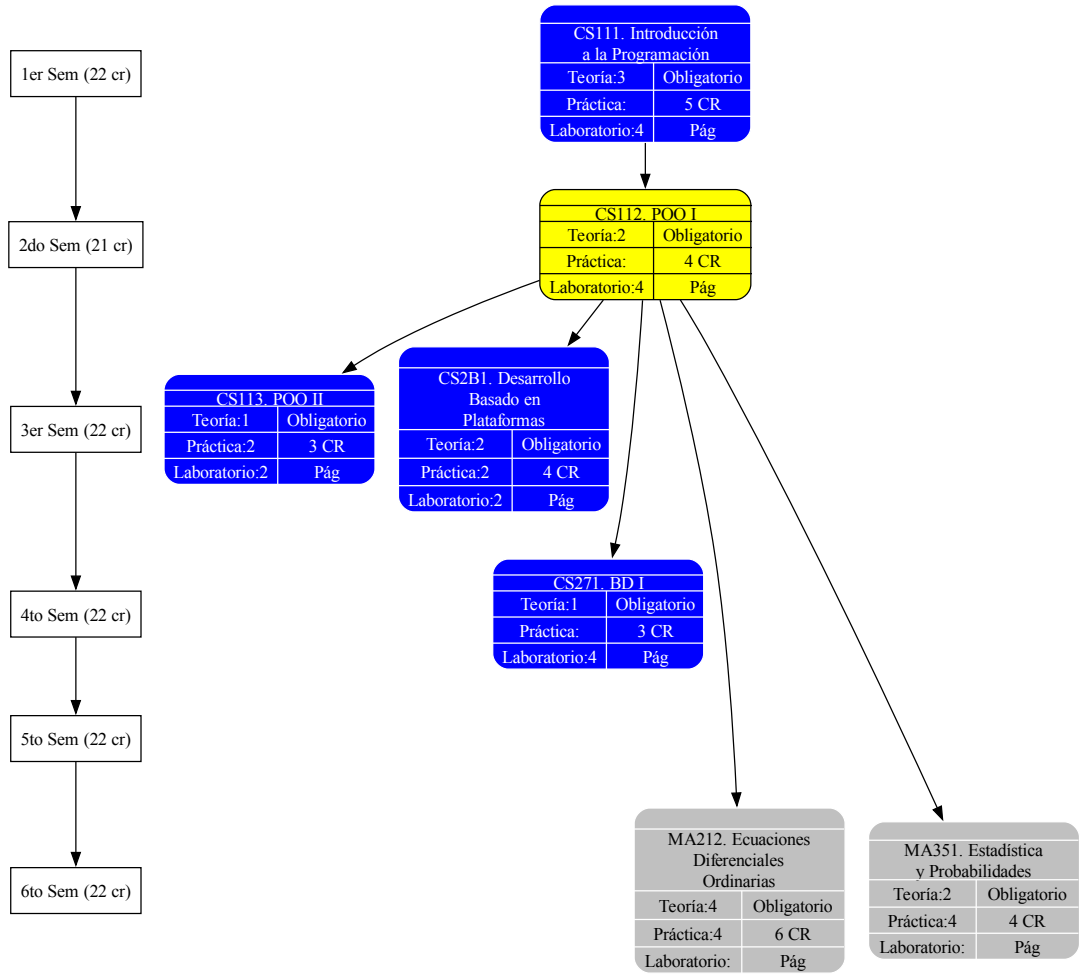


Figura 7.9: Mapa de Conexión. CS112 Programación Orientada a Objetos I

2. Fundamentación

Este es el segundo curso en la secuencia introductoria de programación en Ciencia de la Computación. Incorpora una transición de lenguaje de programación como estrategia pedagógica y se enfoca en los conceptos centrales de Programación Orientada a Objetos (POO) en C++17/C++20, expandiendo las bases del primer curso mientras introduce conceptos de programación de bajo nivel. El curso proporciona una base sólida para los temas avanzados cubiertos en el curso subsiguiente del plan de estudios.

3. Objetivos del curso.

- Introducir al estudiante a los fundamentos del paradigma de orientación a objetos en C++.
- Desarrollar habilidades en programación C++17/C++20, incluyendo plantillas y STL.
- Comprender la gestión de memoria y conceptos de programación de bajo nivel.
- Desarrollar habilidades de resolución de problemas usando principios de POO.
- Dominar técnicas de programación C++ intermedias.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Pragmática del Lenguaje, 2 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Dominios de problemas y paradigma de programación **II:** Criterios para un buen diseño de lenguaje de programación: *a*) Principios de diseño de lenguajes como la ortogonalidad *b*) Definir construcciones de control e iteración *c*) Modularización de software grande **III:** Breve revisión de paradigmas de programación. **IV:** Comparación entre programación funcional e imperativa. **V:** Historia de los lenguajes de programación (énfasis en C y C++).

Objetivos de Aprendizaje **I:** Discutir el contexto histórico de los paradigmas de lenguajes de programación [Familiarizarse]. **II:** Evaluar las compensaciones en el diseño de lenguajes para aplicaciones específicas [Evaluar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.2 Traducción y Ejecución de Lenguajes, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Modelos de ejecución para JIT (Just-In-Time), compilador, intérprete **II:** Uso de código intermedio, por ejemplo, bytecode **III:** Ejecución como código nativo o dentro de una máquina virtual **IV:** Máquinas virtuales y lenguajes intermedios. **V:** Modelos de compilación vs. interpretación.

Objetivos de Aprendizaje **I:** Explicar y comprender las diferencias entre implementaciones de lenguaje compilado, JIT e interpretado, incluyendo los beneficios y limitaciones de cada una [Familiarizarse] **II:** Diferenciar sintaxis y análisis (parsing) de semántica y evaluación [Usar] **III:** Ilustrar ambigüedad en el análisis usando if-then-else anidado/expresión aritmética y mostrar resolución usando orden de precedencia [Evaluar]

Bibliografía: [Stroustrup, 2013, Deitel and Deitel, 2017]

5.3 Fundamentos de Sistemas de Tipos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Un tipo como un conjunto de valores junto con un conjunto de operaciones: *a*) Tipos primitivos (por ejemplo, números, booleanos) *b*) Tipos compuestos construidos a partir de otros tipos (por ejemplo, registros/estructuras, uniones, arreglos, listas, funciones, referencias usando operaciones de conjuntos) **II:** Tipos genéricos (polimorfismo paramétrico): *a*) Definición y ventajas del polimorfismo: paramétrico, de subtipado (subtyping), de sobrecarga (overloading) y de

coerción *b*) Comparación de tipos monomórficos y polimórficos *c*) Comparación con polimorfismo ad-hoc (sobrecarga) y polimorfismo de subtipos *d*) Parámetros genéricos y tipado *e*) Uso de bibliotecas genéricas como colecciones *f*) Comparación con polimorfismo ad hoc (sobrecarga) y polimorfismo de subtipos *g*) Polimorfismo prescriptivo vs descriptivo *h*) Modelos de implementación de tipos polimórficos *i*) Subtipado **III:** Modelos de declaración (enlace, visibilidad, alcance, tiempo de vida). **IV:** Visión general de la verificación de tipos en C++. **V:** Conversión de tipos y casting en C++.

Objetivos de Aprendizaje **I:** Definir y usar piezas de programa (como funciones, clases, métodos) que usen tipos genéricos, incluyendo para colecciones [Usar] **II:** Discutir las diferencias entre genéricos, subtipado y sobrecarga [Familiarizarse] **III:** Explicar múltiples beneficios y limitaciones del tipado estático al escribir, mantener y depurar software [Evaluar]

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.4 Fundamentos de Programación Orientada a Objetos, 14 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Definición de clases: campos, métodos y constructores. **II:** Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. **III:** Despacho dinámico: definición de llamada a método. **IV:** Subtipado: *a*) Polimorfismo de subtipos; conversiones ascendentes (*upcasts*) implícitas en lenguajes tipados. *b*) Noción de reemplazo conductual: los subtipos actúan como el supertipo. *c*) Relación entre subtipado y herencia. **V:** Clases y objetos en C++. **VI:** Constructores y destructores. **VII:** Especificadores de acceso y encapsulamiento. **VIII:** Jerarquías de herencia. **IX:** Polimorfismo y funciones virtuales.

Objetivos de Aprendizaje **I:** Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Usar] **II:** Construir una jerarquía de clases simple utilizando subclases que permita reutilizar código para subclases distintas [Usar] **III:** Predecir y validar el flujo de control en un programa usando despacho dinámico [Evaluar] **IV:** Diseñar clases en C++ con encapsulamiento adecuado [Usar]. **V:** Implementar jerarquías de herencia para reutilización de código [Usar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.5 Modelo de Ejecución y Memoria de Sistemas, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Acceso directo, indirecto e indexado a ubicación de memoria **II:** Representación en tiempo de ejecución de abstracciones de datos como variables, arreglos, vectores, registros, elementos de datos basados en punteros como listas enlazadas y árboles, y objetos **III:** Asignación y acceso de bajo nivel a estructuras de datos de alto nivel como tipos de datos básicos, arreglos n-dimensionales, vectores, registros y objetos **IV:** Retorno de procedimiento como mecanismo de desasignación automática para elementos de datos locales en la pila **V:** Gestión manual de memoria: asignar, desasignar y reutilizar memoria del montón **VI:** Gestión automática de memoria: recolección de basura (garbage collection) como técnica automatizada usando la noción de alcanzabilidad (reachability) **VII:** Distribución de memoria (pila, montón, estática). **VIII:** Aritmética de punteros y operaciones. **IX:** Asignación dinámica de memoria (new/delete). **X:** Fugas de memoria y punteros colgantes. **XI:** Punteros inteligentes (unique_ptr, shared_ptr).

Objetivos de Aprendizaje **I:** Explicar por qué ocurren las fugas de memoria (memory leaks) y los problemas de punteros colgantes (dangling pointers), y qué puede hacer un programador para evitarlos/corregirlos [Usar] **II:** Explicar cómo las implementaciones de lenguajes de programación típicamente organizan la memoria en secciones de datos globales, texto, montón y pila, y cómo características como recursión y gestión de memoria se mapean a este modelo de memoria [Evaluar] **III:** Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Evaluar] **IV:** Implementar RAII para la gestión de recursos [Usar]. **V:** Depurar errores relacionados con memoria en programas C++ [Usar].

Bibliografía: [Stroustrup, 2013, Deitel and Deitel, 2017]

5.6 Programación Funcional, 10 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Expresiones lambda y evaluación: *a*) Enlace (binding) de variables y reglas de ámbito (scope). *b*) Paso de parámetros. *c*) Expresiones lambda anidadas y orden de reducción. **II:** Usar funciones de orden superior (tomar, devolver y almacenar funciones). **III:** Plantillas de funciones y plantillas de clases. **IV:** Especialización de plantillas. **V:** Expresiones lambda en C++11/14. **VI:** Conceptos de la Biblioteca de Plantillas Estándar (STL).

Objetivos de Aprendizaje **I:** Desarrollar funciones útiles que tomen y devuelvan otras funciones [Usar] **II:** Interpretar correctamente las variables y el ámbito léxico en un programa que usa clausuras de funciones [Evaluar] **III:** Usar mecanismos de encapsulación funcional como clausuras e interfaces modulares [Usar] **IV:** Implementar funciones genéricas usando plantillas [Usar]. **V:** Usar expresiones lambda para programación funcional [Usar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.7 Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Clases de colecciones, iteradores y otros componentes comunes de biblioteca. **II:** Contenedores de secuencia (vector, list, deque). **III:** Contenedores asociativos (set, map, unordered_set). **IV:** Adaptadores de contenedor (stack, queue, priority_queue). **V:** Iteradores y algoritmos. **VI:** Funtores y predicados.

Objetivos de Aprendizaje **I:** Usar clases de colecciones e iteradores efectivamente para resolver un problema [Usar] **II:** Definir y usar iteradores y otras operaciones en agregados, incluyendo operaciones que toman funciones como argumentos, en múltiples lenguajes de programación, seleccionando los idiomas más naturales para cada lenguaje [Usar] **III:** Comparar y contrastar los mecanismos para definir y proteger elementos de datos dentro de los enfoques procedural, funcional y orientado a objetos [Evaluar] **IV:** Seleccionar contenedores STL apropiados para necesidades específicas [Usar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.8 Construcciones de Programación Avanzadas, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) **II:** Metaprogramación: macros, programación generativa, desarrollo basado en modelos (model-based development) **III:** Sobrecarga de operadores. **IV:** Semántica de movimiento y referencias a valor-r. **V:** Manejo de excepciones en C++. **VI:** Espacios de nombres y programación modular.

Objetivos de Aprendizaje **I:** Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] **II:** Discutir cómo varias construcciones de programación avanzadas tienen como objetivo mejorar la estructura del programa, la calidad del software y la productividad del programador [Evaluar] **III:** Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] **IV:** Implementar semántica de movimiento para optimización de rendimiento [Usar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019]

5.9 Scripting de Shell, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Comandos del sistema: *a*) Interfaz con sistemas operativos **II:** Tuberías (piping) **III:** Abstracción de archivo y operadores **IV:** Programas y procesos **V:** Automatización de la compilación y pruebas de programas C++ mediante scripts (Makefiles, scripts de build/test).

Objetivos de Aprendizaje I: Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Usar]

Bibliografía: [Robbins and Beebe, 2005]

6. Bibliografía

[Deitel and Deitel, 2017] Deitel, H. and Deitel, P. (2017). *C++17: The Complete Guide*. Pearson, Boston, MA, 10th edition.

[Josuttis, 2019] Josuttis, N. M. (2019). *C++17: The Complete Guide*. Pearson Education, Indianapolis, IN, 1st edition.

[Robbins and Beebe, 2005] Robbins, A. and Beebe, N. H. (2005). *Classic Shell Scripting*. O'Reilly Media, Sebastopol, CA, 1st edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional, Upper Saddle River, NJ, 4th edition.

[Deitel and Deitel, 2017] Deitel, H. and Deitel, P. (2017). *C++17: The Complete Guide*. Pearson, Boston, MA, 10th edition.

[Josuttis, 2019] Josuttis, N. M. (2019). *C++17: The Complete Guide*. Pearson Education, Indianapolis, IN, 1st edition.

[Robbins and Beebe, 2005] Robbins, A. and Beebe, N. H. (2005). *Classic Shell Scripting*. O'Reilly Media, Sebastopol, CA, 1st edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional, Upper Saddle River, NJ, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI161 IA Aplicada (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS111 Introducción a la Programación (1er Sem-Pág 287)

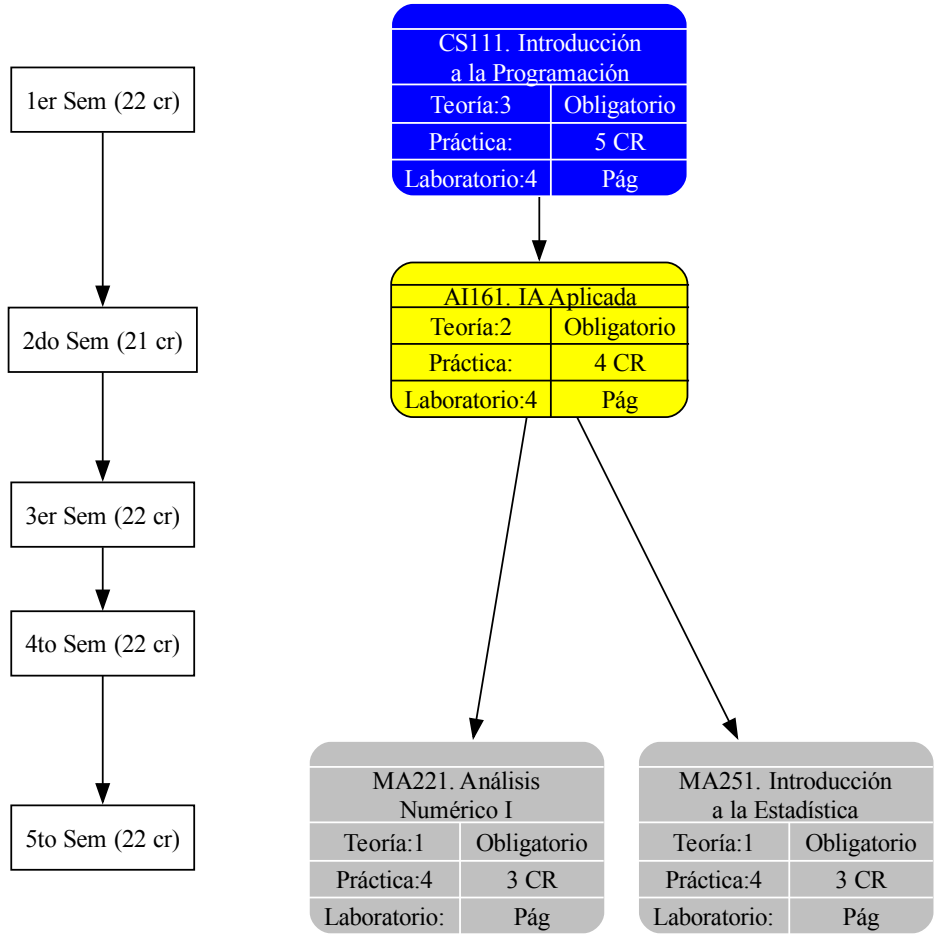


Figura 7.10: Mapa de Conexión. AI161 IA Aplicada

2. Fundamentación

Este curso proporciona una introducción práctica a la Inteligencia Artificial (IA) para estudiantes de todas las disciplinas científicas y de ingeniería. Centrado en desarrollar alfabetización en IA y habilidades prácticas, cubre conceptos fundamentales, herramientas modernas de IA (incluyendo plataformas occidentales y chinas), y uso responsable. Los estudiantes aprenderán a interactuar efectivamente con diversos sistemas de IA, escribir prompts de calidad y aplicar soluciones de IA a problemas en diversos dominios, comprendiendo las implicaciones éticas y contextos culturales del despliegue de IA.

3. Objetivos del curso.

- Desarrollar alfabetización integral en IA comprendiendo conceptos fundamentales, capacidades y limitaciones de los sistemas modernos de IA en diferentes plataformas y culturas.
- Dominar técnicas efectivas de ingeniería de prompts y patrones de interacción con diversas herramientas de IA (Occidentales: ChatGPT, Claude, Gemini; Chinas: DeepSeek, Kimi, ERNIE).
- Aplicar diversas herramientas de IA para resolver problemas prácticos en contextos científicos, de ingeniería y académicos, manteniendo evaluación crítica de los resultados entre plataformas.
- Comprender consideraciones éticas, sesgos, contextos culturales y uso responsable de tecnologías de IA en entornos profesionales y académicos globales.
- Desarrollar flujos de trabajo que integren múltiples herramientas de IA para mejorar la productividad y capacidades de resolución de problemas, comprendiendo fortalezas y especializaciones regionales.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Familiarizarse)

5. Contenido

5.1 Fundamentos de IA y Panorama Global, 16 hr(s)

Competencias: ABET-1,ABET-5

Tópicos: I: ¿Qué es la IA? Definiciones, historia y panorama global actual. **II:** Tipos de sistemas de IA: chatbots, generadores de imágenes, asistentes de investigación entre regiones. **III:** Ecosistemas occidentales de IA: OpenAI, Anthropic, Google, Microsoft. **IV:** Ecosistemas chinos de IA: DeepSeek, Kimi, ERNIE, Zhipu AI, Baidu. **V:** Capacidades y limitaciones de la IA: análisis comparativo entre plataformas. **VI:** Alfabetización digital en la era de la IA: pensamiento crítico sobre diversos resultados de IA.

Objetivos de Aprendizaje I: Aplicar conocimientos fundamentales de IA para identificar tipos de sistemas y sus capacidades [Usar]. **II:** Analizar sistemas inteligentes globales y sus características distintivas [Evaluar]. **III:** Evaluar críticamente los resultados de IA en diferentes contextos culturales [Evaluar].

Bibliografía: [Ng, 2019, Mollick and Mollick, 2024, UNESCO, 2023, Institute, 2024]

5.2 Interacción Efectiva con IA e Ingeniería de Prompts Multiplataforma, 20 hr(s)

Competencias: ABET-1,ABET-5

Tópicos: I: Principios de escritura efectiva de prompts: claridad, contexto, restricciones entre plataformas. **II:** Patrones de prompts: persona, plantilla, cadena de pensamiento, few-shot para diferentes sistemas de IA. **III:** Herramientas occidentales: técnicas específicas para ChatGPT, Claude, Gemini, Copilot. **IV:** Herramientas chinas: características específicas y mejores prácticas para DeepSeek, Kimi, ERNIE. **V:** Estrategias multiplataforma: aprovechar diferentes fortalezas de IA para tareas complejas. **VI:** Refinamiento iterativo: cómo mejorar prompts basándose en resultados de diversos sistemas. **VII:** Sesiones prácticas: talleres comparativos de escritura de prompts entre plataformas.

Objetivos de Aprendizaje I: Diseñar soluciones de prompts efectivas para diversos sistemas de IA [Evaluar]. **II:** Desarrollar flujos de trabajo multiplataforma que integren diferentes herramientas

de IA [Usar]. **III:** Implementar estrategias de refinamiento iterativo para optimizar resultados de IA [Usar].

Bibliografía: [White et al., 2023, Prompt Engineering Institute, 2024, Microsoft, 2024, DeepSeek, 2024]

5.3 Aplicaciones de IA entre Disciplinas y Plataformas, 16 hr(s)

Competencias: ABET-1,ABET-5

Tópicos: I: IA para investigación: revisión de literatura, análisis de datos usando múltiples asistentes de IA. **II:** IA para escritura: documentos académicos, informes con verificación multiplataforma. **III:** IA para resolución de problemas: cálculos científicos, diseño de ingeniería con herramientas especializadas. **IV:** IA para creatividad: lluvia de ideas, desarrollo de conceptos entre contextos culturales. **V:** Fortalezas específicas por plataforma: cuándo usar herramientas de IA occidentales vs. chinas. **VI:** Talleres disciplinares: aplicaciones personalizadas usando diversos ecosistemas de IA. **VII:** Estudios de caso: aplicaciones en el mundo real en investigación científica entre regiones.

Objetivos de Aprendizaje I: Aplicar fundamentos de IA para resolver problemas específicos de cada disciplina [Usar]. **II:** Diseñar soluciones que integren múltiples sistemas inteligentes para tareas complejas [Evaluar]. **III:** Desarrollar aplicaciones prácticas usando herramientas de IA occidentales y chinas [Usar]. **IV:** Implementar flujos de trabajo que aprovechen fortalezas específicas de cada plataforma [Evaluar].

Bibliografía: [Mollick, 2023, Google, 2024, Cloud, 2024, OpenAI, 2024]

5.4 Ética Global en IA y Uso Responsable, 12 hr(s)

Competencias: ABET-1,ABET-5

Tópicos: I: Comprensión de sesgos en IA: perspectivas culturales occidentales y orientales. **II:** Marcos éticos: comparando la Ley de IA de la UE, regulaciones chinas y estándares globales. **III:** Integridad académica: citación apropiada y uso de IA en trabajos académicos entre plataformas. **IV:** Privacidad y seguridad de datos: diferencias regionales en el manejo de datos de IA. **V:** Sensibilidad cultural: navegando resultados de IA en contextos globales. **VI:** Impacto ambiental: consideraciones de sostenibilidad de diferentes sistemas de IA. **VII:** Desarrollo de pautas personales para el uso ético de IA en entornos internacionales. **VIII:** Estudios de caso: dilemas éticos en el despliegue de IA entre regiones.

Objetivos de Aprendizaje I: Demostrar responsabilidad profesional en el uso de tecnologías de IA [Evaluar]. **II:** Aplicar principios éticos profesionales en contextos globales de IA [Usar]. **III:** Desarrollar pautas éticas para mitigar sesgos en sistemas de IA [Evaluar]. **IV:** Evaluar implicaciones éticas del despliegue de IA en diferentes culturas [Usar].

Bibliografía: [UNESCO, 2023, Union, 2024, Bender et al., 2021, of Cyberspace Studies, 2024, Tencent AI Lab, 2023]

6. Bibliografía

[Bender et al., 2021] Bender, E. M., Gebru, T., et al. (2021). On the dangers of stochastic parrots: Can language models be too big? *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.* Foundational paper on AI ethics with ongoing relevance.

[Cloud, 2024] Cloud, A. (2024). AI Ethics and best practices in chinese context. Chinese perspective on AI ethics and implementation guidelines.

[DeepSeek, 2024] DeepSeek (2024). Deepseek model documentation and best practices. Official documentation for DeepSeek AI models and usage guidelines.

[Google, 2024] Google (2024). Introduction to responsible ai. Google's updated framework for responsible AI development and use.

[Institute, 2024] Institute, T. R. (2024). Ai development in china: Current status and future trends. Technical report, Tencent Research Institute. Comprehensive analysis of Chinese AI ecosystem development.

[Microsoft, 2024] Microsoft (2024). Prompt crafting for ai systems. Official Microsoft prompt engineering guidance updated for 2024.

[Mollick, 2023] Mollick, E. (2023). Chatgpt and how ai disrupts industries. *Harvard Business Review*. Analysis of AI's practical impact across global sectors.

[Mollick and Mollick, 2024] Mollick, E. and Mollick, L. (2024). *Co-Intelligence: Living and Working with AI*. Penguin Random House. Practical guide to human-AI collaboration.

[Ng, 2019] Ng, A. (2019). AI for everyone. Technical report, DeepLearning.AI. Continuously updated online course, available on Coursera.

[of Cyberspace Studies, 2024] of Cyberspace Studies, C. A. (2024). Ai governance and ethics in china. Technical report, Cyberspace Administration of China. Official Chinese perspective on AI governance and ethical standards.

[OpenAI, 2024] OpenAI (2024). Best practices for prompt engineering. Updated official prompt engineering guidelines from OpenAI.

[Prompt Engineering Institute, 2024] Prompt Engineering Institute (2024). Prompt engineering guide. `https://www.promptingguide.ai/`. Recurso integral de ingeniería de prompts actualizado continuamente. Accedido: 15 de enero de 2024.

[Tencent AI Lab, 2023] Tencent AI Lab (2023). Responsible ai practices in chinese tech industry. Technical report, Tencent AI Lab. Industry perspective on AI ethics from leading Chinese tech company.

[UNESCO, 2023] UNESCO (2023). AI and education: Guidance for policy-makers. Technical report, UNESCO. Latest guidance on AI literacy and ethical implementation in education.

[Union, 2024] Union, E. (2024). AI Act: Regulatory framework. Technical report, European Commission. Comprehensive AI regulation framework effective 2024.

[White et al., 2023] White, J. et al. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. Peer-reviewed research on prompt engineering patterns.

[Bender et al., 2021] Bender, E. M., Gebru, T., et al. (2021). On the dangers of stochastic parrots: Can language models be too big? *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Foundational paper on AI ethics with ongoing relevance.

[Cloud, 2024] Cloud, A. (2024). AI Ethics and best practices in chinese context. Chinese perspective on AI ethics and implementation guidelines.

[DeepSeek, 2024] DeepSeek (2024). Deepseek model documentation and best practices. Official documentation for DeepSeek AI models and usage guidelines.

[Google, 2024] Google (2024). Introduction to responsible ai. Google's updated framework for responsible AI development and use.

[Institute, 2024] Institute, T. R. (2024). Ai development in china: Current status and future trends. Technical report, Tencent Research Institute. Comprehensive analysis of Chinese AI ecosystem development.

[Microsoft, 2024] Microsoft (2024). Prompt crafting for ai systems. Official Microsoft prompt engineering guidance updated for 2024.

[Mollick, 2023] Mollick, E. (2023). Chatgpt and how ai disrupts industries. *Harvard Business Review*. Analysis of AI's practical impact across global sectors.

[Mollick and Mollick, 2024] Mollick, E. and Mollick, L. (2024). *Co-Intelligence: Living and Working with AI*. Penguin Random House. Practical guide to human-AI collaboration.

[Ng, 2019] Ng, A. (2019). AI for everyone. Technical report, DeepLearning.AI. Continuously updated online course, available on Coursera.

[of Cyberspace Studies, 2024] of Cyberspace Studies, C. A. (2024). Ai governance and ethics in china. Technical report, Cyberspace Administration of China. Official Chinese perspective on AI governance and ethical standards.

[OpenAI, 2024] OpenAI (2024). Best practices for prompt engineering. Updated official prompt engineering guidelines from OpenAI.

[Prompt Engineering Institute, 2024] Prompt Engineering Institute (2024). Prompt engineering guide. `https://www.promptingguide.ai/`. Recurso integral de ingeniería de prompts actualizado continuamente. Accedido: 15 de enero de 2024.

[Tencent AI Lab, 2023] Tencent AI Lab (2023). Responsible ai practices in chinese tech industry. Technical report, Tencent AI Lab. Industry perspective on AI ethics from leading Chinese tech company.

[UNESCO, 2023] UNESCO (2023). AI and education: Guidance for policy-makers. Technical report, UNESCO. Latest guidance on AI literacy and ethical implementation in education.

[Union, 2024] Union, E. (2024). AI Act: Regulatory framework. Technical report, European Commission. Comprehensive AI regulation framework effective 2024.

[White et al., 2023] White, J. et al. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. Peer-reviewed research on prompt engineering patterns.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA111 Cálculo I (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA100 Matemática Básica (1er Sem-Pág 292)

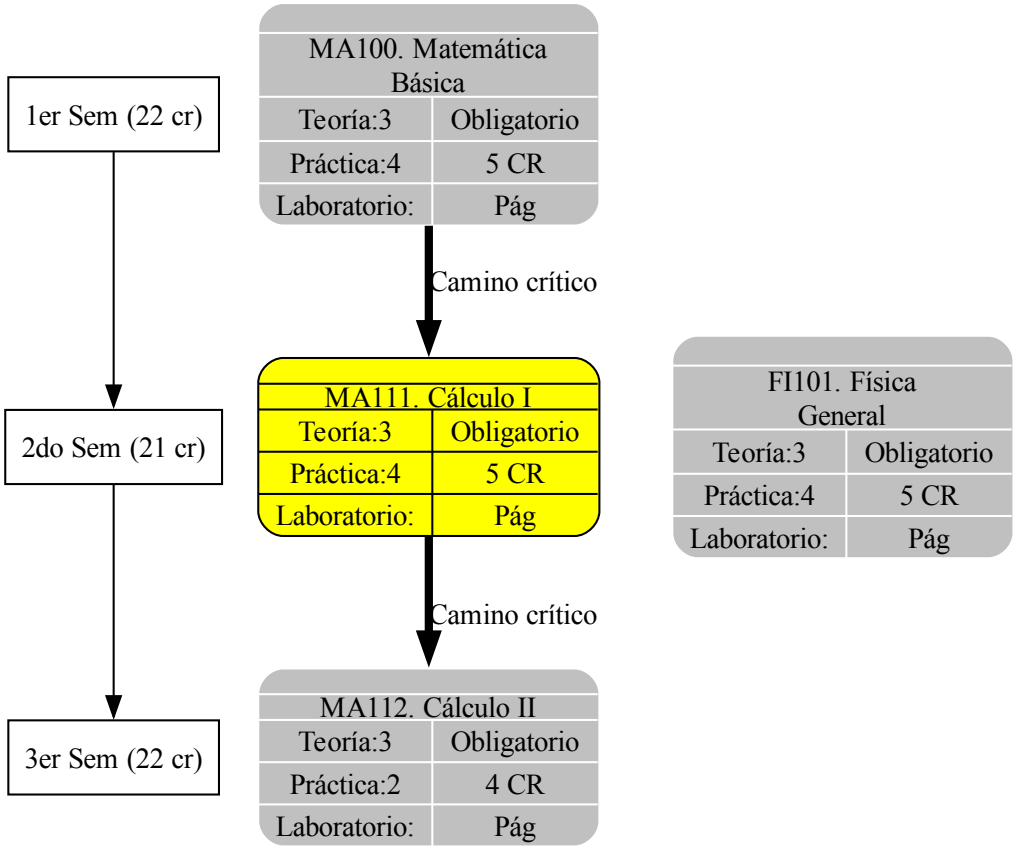


Figura 7.11: Mapa de Conexión. MA111 Cálculo I

2. Fundamentación

El curso incluye conocimientos sobre relaciones binarias, funciones reales de variable real. Límites, Continuidad y Derivadas de Funciones reales, aplicaciones (máximos y mínimos) y Diferenciales aplicados a la solución de problemas que involucren a los seres vivos.

3. Objetivos del curso.

- Comprender los conceptos fundamentales de límites y continuidad.
- Dominar las técnicas de diferenciación para funciones algebraicas y trascendentes.
- Aplicar la derivada para resolver problemas de optimización y tasas de cambio.
- Desarrollar un enfoque riguroso de la definición formal de la derivada.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

5. Contenido

5.1 Aritmética Computacional y Análisis de Error, 25 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aritmética de punto flotante y el estándar IEEE **II:** Fuentes de error: truncamiento, redondeo y discretización **III:** Propagación de error y número de condición **IV:** Estabilidad de algoritmos numéricos **V:** Error absoluto, relativo y directo/inverso

Objetivos de Aprendizaje **I:** Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse] **II:** Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse] **III:** Calcular el error absoluto y relativo de una aproximación [Usar] **IV:** Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar] **V:** Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar] **VI:** Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

Bibliografía: [Stewart, 2016, Larson and Edwards, 2018]

5.2 Búsqueda de Raíces y Optimización Numérica Básica, 25 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje **I:** Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

6. Bibliografía

[Larson and Edwards, 2018] Larson, R. and Edwards, B. H. (2018). *Cálculo*. Cengage Learning, 10th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de una variable: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de una variable*. Pearson, 13th edition.

[Larson and Edwards, 2018] Larson, R. and Edwards, B. H. (2018). *Cálculo*. Cengage Learning, 10th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de una variable: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de una variable*. Pearson, 13th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

FI101 Física General (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA111 Cálculo I (2do Sem-Pág 331)

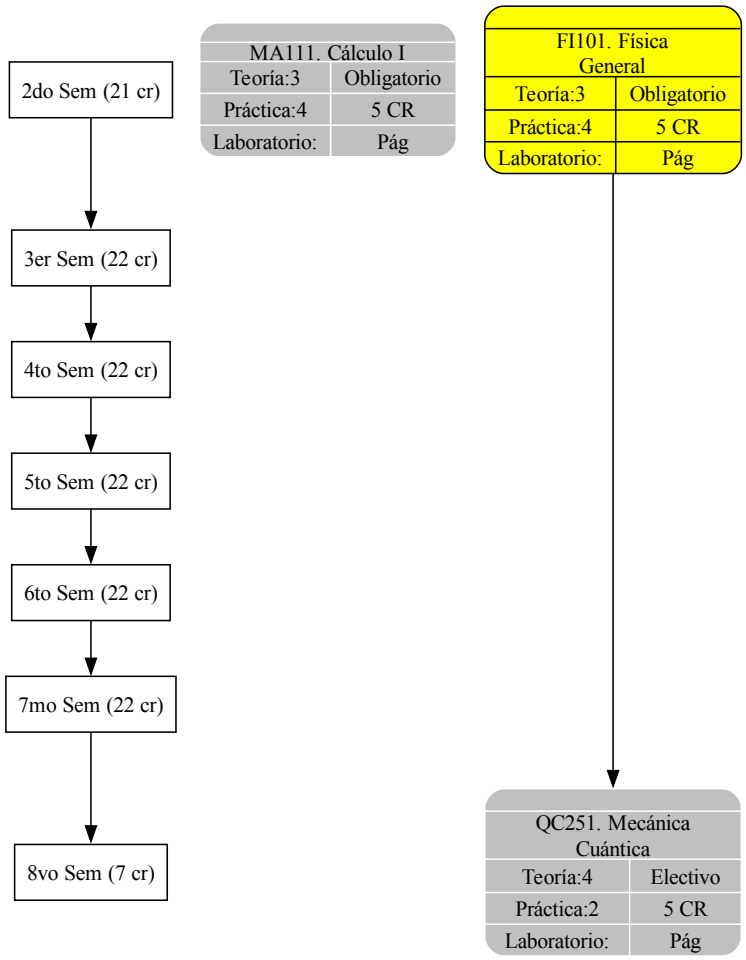


Figura 7.12: Mapa de Conexión. FI101 Física General

2. Fundamentación

El curso incluye conocimientos sobre los principios y leyes fundamentales de la Física, Mecánica Clásica de Newton, Cinemática, Estática, Dinámica, Energía y Trabajo y Fuerzas conservativas y no conservativas, Energía e Impulso. Pone énfasis en los fenómenos acústicos, ondas electromagnéticas y óptica con aplicaciones en las ciencias biológicas.

3. Objetivos del curso.

- Comprender y aplicar las leyes de Newton en el análisis del movi- miento y equilibrio de los cuerpos.
- Aplicar los conceptos de trabajo, energía y conservación en la resolución de problemas mecánicos complejos.
- Analizar las leyes de la termodinámica, el comportamiento de los fluidos y los fenómenos de transferencia de calor.
- Estudiar los principios de la óptica y los fenómenos ondulatorios para comprender la naturaleza de la luz y el sonido.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Mecánica de la Partícula y el Sólido, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Cantidades físicas, sistemas de unidades y vectores. **II:** Cinemática: Movimiento en una y dos dimensiones. **III:** Dinámica: Leyes de Newton y sus aplicaciones. **IV:** Estática y equilibrio del cuerpo rígido.

Objetivos de Aprendizaje **I:** Describir el movimiento de los cuerpos utilizando magnitudes cinemáticas y vectoriales [Familiarizarse]. **II:** Aplicar las leyes de Newton para determinar fuerzas y aceleraciones en sistemas físicos [Usar]. **III:** Resolver problemas de equilibrio estático mediante el análisis de fuerzas y momentos de torsión [Evaluar].

Bibliografía: [Sears et al., 2013, Tipler and Mosca, 2010]

5.2 Trabajo, Energía e Impulso, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Trabajo y energía cinética. **II:** Energía potencial y conservación de la energía mecánica. **III:** Cantidad de movimiento e impulso. **IV:** Colisiones elásticas e inelásticas.

Objetivos de Aprendizaje **I:** Identificar las formas de energía y su transformación en procesos físicos conservativos [Familiarizarse]. **II:** Utilizar el teorema del trabajo y la energía para analizar sistemas dinámicos [Usar]. **III:** Aplicar la conservación de la cantidad de movimiento en la resolución de problemas de colisión [Evaluar].

Bibliografía: [Sears et al., 2013, Resnick et al., 2005]

5.3 Mecánica de Fluidos y Termodinámica, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Estática y dinámica de fluidos (Bernoulli y Arquímedes). **II:** Temperatura, calor y leyes de la termodinámica. **III:** Teoría cinética de los gases. **IV:** Procesos termodinámicos y máquinas térmicas.

Objetivos de Aprendizaje I: Comprender las propiedades fundamentales de los fluidos y los mecanismos de transferencia de calor [Familiarizarse]. **II:** Calcular variables de estado en procesos térmicos utilizando las leyes de la termodinámica [Usar]. **III:** Modelar el comportamiento de fluidos y gases mediante la implementación de ecuaciones fundamentales [Evaluar].

Bibliografía: [Sears et al., 2013, Serway and Jewett, 2018]

5.4 Ondas y Óptica, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Movimiento ondulatorio y fenómenos acústicos. **II:** Naturaleza y propagación de la luz. **III:** Óptica geométrica: Reflexión y refracción. **IV:** Ondas electromagnéticas e interferencia.

Objetivos de Aprendizaje I: Reconocer las características de las ondas mecánicas y electromagnéticas en el entorno [Familiarizarse]. **II:** Analizar la formación de imágenes en espejos y lentes mediante trazado de rayos [Usar]. **III:** Evaluar fenómenos de interferencia y difracción basados en el modelo ondulatorio de la luz [Evaluar].

Bibliografía: [Alonso and Finn, 1971, Sears et al., 2009]

6. Bibliografía

[Alonso and Finn, 1971] Alonso, M. and Finn, E. (1971). *Física I.* Fondo Educativo Interamericano.

[Resnick et al., 2005] Resnick, R., Halliday, D., and Krane, K. (2005). *Física Vol. 1.* CECSA, 5th edition.

[Sears et al., 2009] Sears, F., Zemansky, M., Young, H., and Freedman, R. (2009). *Física Universitaria, Vol. 2.* Pearson, 12th edition.

[Sears et al., 2013] Sears, F., Zemansky, M., Young, H., and Freedman, R. (2013). *Física Universitaria, Vol. 1.* Pearson, 13th edition.

[Serway and Jewett, 2018] Serway, R. and Jewett, J. (2018). *Física para Ciencias e Ingeniería Vol. 1.* Cengage Learning, 10th edition.

[Tipler and Mosca, 2010] Tipler, P. and Mosca, G. (2010). *Física para la Ciencia y la Tecnología. Vol. 1.* Reverté, 6th edition.

[Alonso and Finn, 1971] Alonso, M. and Finn, E. (1971). *Física I.* Fondo Educativo Interamericano.

[Resnick et al., 2005] Resnick, R., Halliday, D., and Krane, K. (2005). *Física Vol. 1.* CECSA, 5th edition.

[Sears et al., 2009] Sears, F., Zemansky, M., Young, H., and Freedman, R. (2009). *Física Universitaria, Vol. 2.* Pearson, 12th edition.

[Sears et al., 2013] Sears, F., Zemansky, M., Young, H., and Freedman, R. (2013). *Física Universitaria, Vol. 1.* Pearson, 13th edition.

[Serway and Jewett, 2018] Serway, R. and Jewett, J. (2018). *Física para Ciencias e Ingeniería Vol. 1.* Cengage Learning, 10th edition.

[Tipler and Mosca, 2010] Tipler, P. and Mosca, G. (2010). *Física para la Ciencia y la Tecnología. Vol. 1.* Reverté, 6th edition.

Tercer Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS113 Programación Orientada a Objetos II (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 321)

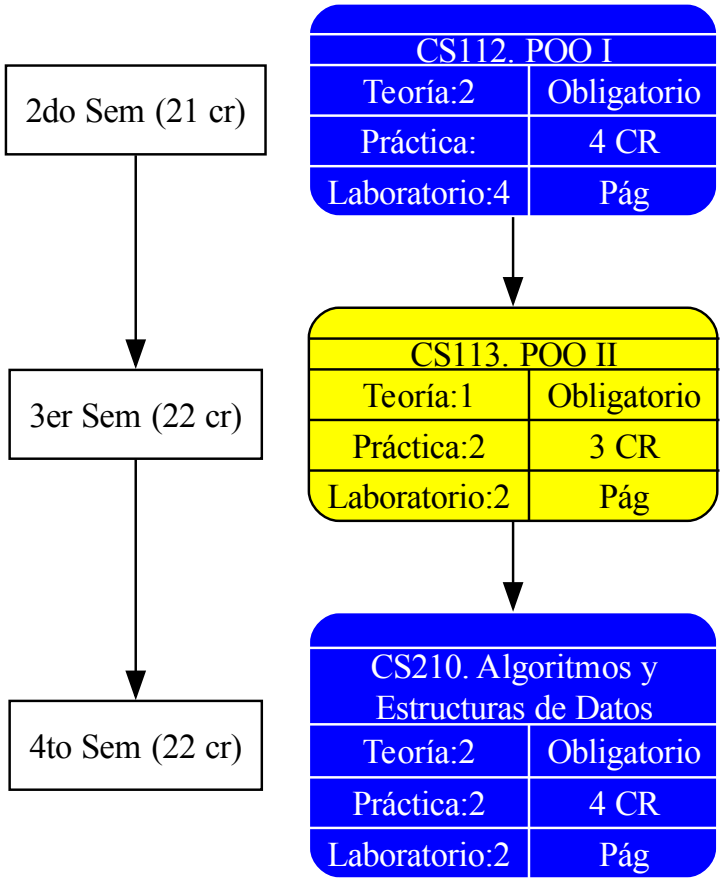


Figura 7.13: Mapa de Conexión. CS113 Programación Orientada a Objetos II

2. Fundamentación

Este es el tercer curso en la secuencia de cursos introductorios en ciencia de la computación. Este curso explora características avanzadas de C++17/C++20 centrándose en el desarrollo de sistemas de alto rendimiento. Los temas clave incluyen metaprogramación con plantillas, semántica de movimiento, reenvío perfecto, optimización RAII, patrones de herencia múltiple/virtual, concurrencia con

std::thread y async/await, programación sin bloqueo, paradigmas modernos de C++ como CRTP y plantillas de expresión, e interfaz de C++ con otros lenguajes. El curso prepara a los estudiantes para motores de juego, computación de alto rendimiento y desarrollo embebido donde C++ es dominante.

3. Objetivos del curso.

- Dominar características avanzadas de C++17/C++20, incluyendo plantillas y metaprogramación.
- Desarrollar habilidades en programación concurrente y de sistemas.
- Comprender técnicas de optimización de rendimiento.
- Prepararse para aplicaciones de computación de alto rendimiento.
- Dominar la metaprogramación con plantillas y el cómputo en tiempo de compilación.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Clases de colecciones, iteradores y otros componentes comunes de biblioteca. **II:** Contenedores asociativos (std::set, std::map, variantes no ordenadas). **III:** Adaptadores de contenedor (stack, queue, priority_queue). **IV:** Algoritmos avanzados de STL. **V:** Funtores y predicados. **VI:** Categorías de iteradores y rasgos.

Objetivos de Aprendizaje I: Usar clases de colecciones e iteradores efectivamente para resolver un problema [Usar] **II:** Definir y usar iteradores y otras operaciones en agregados, incluyendo operaciones que toman funciones como argumentos, en múltiples lenguajes de programación, seleccionando los idiomas más naturales para cada lenguaje [Evaluar] **III:** Comparar y contrastar los beneficios y costos/impacto de usar herencia (subclases) y composición (específicamente, cómo basar la composición en funciones de orden superior) [Usar] **IV:** Implementar asignadores personalizados para contenedores STL [Usar].

Bibliografía: [Stroustrup, 2013, Josuttis, 2019, Vandevoorde et al., 2017]

5.2 Sistemas de Tipos: Polimorfismo y Tipado Complementario, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Tipos genéricos (polimorfismo paramétrico): *a*) Definición y ventajas del polimorfismo: paramétrico, de subtipado (subtyping), de sobrecarga (overloading) y de coerción *b*) Comparación de tipos monomórficos y polimórficos *c*) Comparación con polimorfismo ad-hoc (sobrecarga) y polimorfismo de subtipos *d*) Parámetros genéricos y tipado *e*) Uso de bibliotecas genéricas como colecciones *f*) Comparación con polimorfismo ad hoc (sobrecarga) y polimorfismo de subtipos *g*) Polimorfismo prescriptivo vs descriptivo *h*) Modelos de implementación de tipos polimórficos *i*) Subtipado **II:** SFINAE (Fallo de Sustitución No es un Error): fundamento histórico de la programación genérica restringida en C++, base de bibliotecas anteriores a C++20. **III:** Concepts (C++20): sintaxis declarativa para restringir parámetros de plantilla, sucesor moderno de SFINAE con mejores diagnósticos de error. **IV:** if constexpr (C++17): ramificación en tiempo de compilación, reemplazo de gran parte del despacho por etiquetas (tag dispatch) antes resuelto con SFINAE. **V:** Reenvío perfecto y referencias universales. **VI:** Plantillas variádicas. **VII:** Cómputo en tiempo de compilación con constexpr y consteval.

Objetivos de Aprendizaje **I:** Definir y usar piezas de programa (como funciones, clases, métodos) que usen tipos genéricos, incluyendo para colecciones [Evaluar] **II:** Discutir las diferencias entre genéricos, subtipado y sobrecarga [Evaluar] **III:** Aplicar Concepts para restringir parámetros de plantilla y mejorar diagnósticos de error [Usar]. **IV:** Distinguir cuándo usar Concepts, if constexpr o SFINAE heredado al mantener código C++ existente [Evaluar]. **V:** Implementar reenvío perfecto para paso eficiente de argumentos [Usar].

Bibliografía: [Stroustrup, 2013, Vandevoorde et al., 2017, Stroustrup, 2022]

5.3 Gestión de Memoria, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Asignación y acceso de bajo nivel a estructuras de datos de alto nivel como tipos de datos básicos, arreglos n-dimensionales, vectores, registros y objetos **II:** Retorno de procedimiento como mecanismo de desasignación automática para elementos de datos locales en la pila **III:** Gestión manual de memoria: asignar, desasignar y reutilizar memoria del montón **IV:** Gestión automática de memoria: recolección de basura (garbage collection) como técnica automatizada usando la noción de alcanzabilidad (reachability) **V:** Lvalues y rvalues en C++11/14/17/20. **VI:** Constructores de movimiento y operadores de asignación de movimiento. **VII:** Referencias a valor-r y colapso de referencias. **VIII:** Patrón RAII (Adquisición de Recursos es Inicialización). **IX:** Personalización de punteros inteligentes. **X:** Detección y prevención de fugas de memoria.

Objetivos de Aprendizaje **I:** Explicar por qué ocurren las fugas de memoria (memory leaks) y los problemas de punteros colgantes (dangling pointers), y qué puede hacer un programador para evitarlos/corregirlos [Usar] **II:** Explicar cómo las implementaciones de lenguajes de programación típicamente organizan la memoria en secciones de datos globales, texto, montón y pila, y cómo características como recursión y gestión de memoria se mapean a este modelo de memoria [Evaluar] **III:** Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Evaluar] **IV:** Implementar semántica de movimiento para optimización de recursos [Usar]. **V:** Diseñar clases RAII para gestión de recursos segura ante excepciones [Usar].

Bibliografía: [Meyers, 2014, Stroustrup, 2013]

5.4 Computación Paralela y Distribuida, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Seguridad (safety) y vivacidad (liveness): *a*) Condiciones de carrera (race conditions) *b*) Dependencias/precondiciones *c*) Modelos de fallos (fault models) *d*) Terminación **II:** Modelos de programación: *a*) Modelo de actores (*actor models*) *b*) Modelos procedurales y reactivos *c*) Modelos de programación síncrona/asíncrona *d*) Paralelismo de datos (*data parallelism*) **III:** Comunicación y coordinación: *a*) Mutexes *b*) Paso de mensajes (*message-passing*) *c*) Memoria compartida (*shared memory*) *d*) Cobegin-coend *e*) Monitores *f*) Canales (*channels*) *g*) Hilos (*threads*) *h*) Guardas (*guards*) **IV:** std::thread y gestión de hilos. **V:** std::jthread (C++20): unión automática (RAII) y cancelación cooperativa vía stop_token. **VI:** Mutexes, candados y variables de condición. **VII:** Mecanismos de sincronización de C++20: std::latch, std::barrier, std::counting_semaphore. **VIII:** Operaciones atómicas y programación sin bloqueo. **IX:** std::async y futures. **X:** std::execution y el modelo de senders-receivers (C++26, experimental): panorama hacia dónde evoluciona la concurrencia estándar en C++, presentado a nivel conceptual, sin evaluación en laboratorio dada su disponibilidad limitada (parcial en GCC). **XI:** Patrones de diseño seguros para hilos. **XII:** Carreras de datos y prevención de interbloqueos.

Objetivos de Aprendizaje **I:** Implementar programas concurrentes correctos usando múltiples modelos de programación, como memoria compartida, actores, futuros, construcciones de sincronización y primitivas de paralelismo de datos [Usar] **II:** Usar construcciones de sincronización como métodos

monitorizados/sincronizados en un programa simple [Usar] **III:** Usar un modelo de paso de mensajes para analizar un protocolo de comunicación [Evaluar] **IV:** Implementar estructuras de datos sin bloqueo usando operaciones atómicas [Usar]. **V:** Diferenciar std::thread de std::jthread y justificar cuándo usar cada uno [Usar]. **VI:** Analizar la seguridad en hilos de algoritmos concurrentes [Evaluar].

Bibliografía: [Williams, 2019, Stroustrup, 2013]

5.5 Construcciones de Programación Avanzadas, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) **II:** Metaprogramación: macros, programación generativa, desarrollo basado en modelos (model-based development) **III:** Herencia múltiple y herencia virtual. **IV:** Patrón CRTP (Patrón de Plantilla Curiosamente Recursivo). **V:** deducing this (C++23): simplifica CRTP al eliminar la necesidad de static_cast<Derived&>(*this) para polimorfismo estático. **VI:** Plantillas de expresión. **VII:** Rasgos de tipos e introspección en tiempo de compilación. **VIII:** Diseño basado en políticas. **IX:** Diseño de interfaz para bibliotecas.

Objetivos de Aprendizaje **I:** Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] **II:** Discutir cómo varias construcciones de programación avanzadas tienen como objetivo mejorar la estructura del programa, la calidad del software y la productividad del programador [Evaluar] **III:** Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Evaluar] **IV:** Implementar CRTP para polimorfismo estático [Usar]. **V:** Diseñar interfaces de biblioteca extensibles usando diseño basado en políticas [Usar].

Bibliografía: [Stroustrup, 2013, Gregor, 2017]

5.6 Modelo de Ejecución y Memoria de Sistemas, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Representación en tiempo de ejecución de abstracciones de datos como variables, arreglos, vectores, registros, elementos de datos basados en punteros como listas enlazadas y árboles, y objetos **II:** Estructuras de datos para traducción, ejecución, traducción y movilidad de código como pila, montón (heap), alias (compartición usando punteros), secuencia indexada y cadena **III:** Implementación del despacho dinámico introducido en POO I: tabla de funciones virtuales (vtable), puntero vtable (vptr), y su costo en memoria y rendimiento frente al despacho estático. **IV:** Interfaz de C++ con otros lenguajes (FFI). **V:** Llamadas al sistema e interacción con el SO. **VI:** E/S mapeada en memoria. **VII:** Perfilado de rendimiento y evaluación comparativa. **VIII:** Técnicas de optimización para HPC.

Objetivos de Aprendizaje **I:** Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Usar] **II:** Explicar cómo las implementaciones de lenguajes de programación típicamente organizan la memoria en secciones de datos globales, texto, montón y pila, y cómo características como recursión y gestión de memoria se mapean a este modelo de memoria [Evaluar] **III:** Explicar por qué ocurren las fugas de memoria (memory leaks) y los problemas de punteros colgantes (dangling pointers), y qué puede hacer un programador para evitarlos/corregirlos [Usar] **IV:** Perfilar y optimizar programas C++ para rendimiento [Usar]. **V:** Explicar cómo el compilador implementa el polimorfismo en tiempo de ejecución mediante vtables [Evaluar]. **VI:** Interfazar código C++ con bibliotecas C [Usar].

Bibliografía: [Stroustrup, 2013, Williams, 2019]

5.7 Construcciones de Programación Avanzadas, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) **II:** Patrones creacionales, estructurales y de comportamiento. **III:** Patrones singleton seguros para hilos. **IV:** Modismos de diseño modernos en C++. **V:** Inyección de dependencias e inversión de control.

Objetivos de Aprendizaje **I:** Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] **II:** Discutir cómo varias construcciones de programación avanzadas tienen como objetivo mejorar la estructura del programa, la calidad del software y la productividad del programador [Evaluar] **III:** Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] **IV:** Implementar patrones de diseño seguros para hilos [Usar]. **V:** Evaluar compensaciones arquitectónicas en sistemas C++ [Evaluar].

Bibliografía: [Gamma et al., 1994, Stroustrup, 2013]

5.8 Construcciones de Programación Avanzadas, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Abstracciones de control: manejo de excepciones, continuaciones (continuations), mónadas (monads). **II:** Optimizaciones del compilador y banderas. **III:** Metodologías de evaluación comparativa. **IV:** Estructuras de datos amigables con la caché. **V:** Predicción de bifurcación y alineación. **VI:** Reducción del tiempo de compilación. **VII:** Integración continua para pruebas de rendimiento.

Objetivos de Aprendizaje **I:** Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] **II:** Discutir cómo varias construcciones de programación avanzadas tienen como objetivo mejorar la estructura del programa, la calidad del software y la productividad del programador [Evaluar] **III:** Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] **IV:** Aplicar técnicas de optimización para mejorar el rendimiento del programa [Usar]. **V:** Interpretar resultados de evaluación comparativa para guiar decisiones de optimización [Usar].

Bibliografía: [Meyers, 2014, Stroustrup, 2013]

5.9 Control de Versiones y CI/CD, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Automatización de procesos de software: *a*) Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. *b*) Integración Continua (CI): el uso de automatización y pruebas automatizadas para hacer una validación preliminar de que la revisión actual del tronco se construye y pasa las pruebas (básicas). *c*) Despliegue Continuo (CD): el uso de automatización para liberar de forma automática cada cambio que supera las pruebas hacia el entorno de producción, garantizando entregas frecuentes y confiables. *d*) Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. **II:** Sistemas de construcción para C++: CMake, el estándar de facto de la industria. **III:** Alternativas modernas: xmake, Meson y Bazel – comparación de enfoques declarativos frente a basados en Lua/Python. **IV:** Organización de proyectos multi-archivo: unidades de traducción, cabeceras, bibliotecas estáticas y dinámicas. **V:** Gestión de dependencias en C++ (vcpkg, Conan).

Objetivos de Aprendizaje **I:** Comprender el uso de sistemas de CI/CD como una fuente de verdad para el estado del código compartido del equipo (éxito en la construcción y pruebas) [Familiarizarse] **II:** Configurar un proyecto C++ multi-archivo usando un sistema de construcción declarativo

[Usar]. **III:** Comparar CMake con al menos una alternativa moderna (xmake, Meson o Bazel) en velocidad de compilación y reproducibilidad [Evaluar].

Bibliografía: [Stroustrup, 2013, Stroustrup, 2022]

6. Bibliografía

[Gamma et al., 1994] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, Reading, MA, 1st edition.

[Gregor, 2017] Gregor, D. (2017). *Advanced C++: Master the Technique of Template Metaprogramming*. Addison-Wesley Professional, Upper Saddle River, NJ, 1st edition.

[Josuttis, 2019] Josuttis, N. M. (2019). *C++17: The Complete Guide*. Pearson Education, Indianapolis, IN, 1st edition.

[Meyers, 2014] Meyers, S. (2014). *Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14*. O'Reilly Media, Sebastopol, CA, 1st edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional, Upper Saddle River, NJ, 4th edition.

[Stroustrup, 2022] Stroustrup, B. (2022). *A Tour of C++*. Addison-Wesley Professional, Boston, MA, 3rd edition.

[Vandevoorde et al., 2017] Vandevoorde, D., Josuttis, N. M., and Gregor, D. (2017). *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Upper Saddle River, NJ, 2nd edition.

[Williams, 2019] Williams, A. (2019). *C++ Concurrency in Action*. Manning Publications, Shelter Island, NY, 2nd edition.

[Gamma et al., 1994] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, Reading, MA, 1st edition.

[Gregor, 2017] Gregor, D. (2017). *Advanced C++: Master the Technique of Template Metaprogramming*. Addison-Wesley Professional, Upper Saddle River, NJ, 1st edition.

[Josuttis, 2019] Josuttis, N. M. (2019). *C++17: The Complete Guide*. Pearson Education, Indianapolis, IN, 1st edition.

[Meyers, 2014] Meyers, S. (2014). *Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14*. O'Reilly Media, Sebastopol, CA, 1st edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional, Upper Saddle River, NJ, 4th edition.

[Stroustrup, 2022] Stroustrup, B. (2022). *A Tour of C++*. Addison-Wesley Professional, Boston, MA, 3rd edition.

[Vandevoorde et al., 2017] Vandevoorde, D., Josuttis, N. M., and Gregor, D. (2017). *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Upper Saddle River, NJ, 2nd edition.

[Williams, 2019] Williams, A. (2019). *C++ Concurrency in Action*. Manning Publications, Shelter Island, NY, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS1D1 Estructuras Discretas I (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - MA100 Matemática Básica (1er Sem-Pág 292)

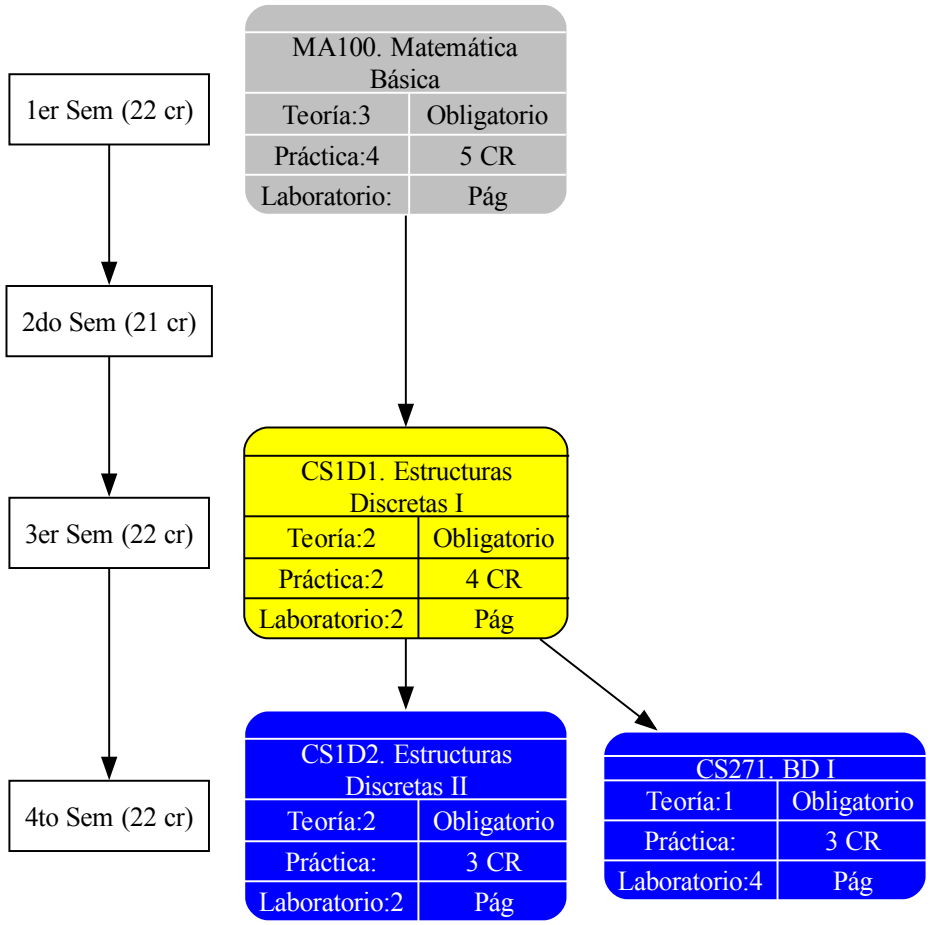


Figura 7.14: Mapa de Conexión. CS1D1 Estructuras Discretas I

2. Fundamentación

Las estructuras discretas son la columna vertebral de la ciencia de la computación. Este primer curso se centra en la lógica fundamental y el razonamiento formal necesarios para evaluar la corrección de los procesos computacionales. Los estudiantes dominarán el lenguaje formal de las matemáticas para evitar ambigüedad y aprenderán diversas técnicas de demostración esenciales para el análisis riguroso de algoritmos y especificaciones de software.

3. Objetivos del curso.

- Comprender los principios de la lógica matemática y del cálculo proposicional.
- Aplicar la lógica de predicados para modelar restricciones complejas en computación.
- Dominar múltiples técnicas de demostración, incluyendo directa, por contradicción e inducción.
- Desarrollar la capacidad de argumentar formalmente la corrección de enunciados matemáticos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Familiarizarse)

5. Contenido

5.1 Lógica y Fundamentos, 24 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones **II:** Conjuntos: operaciones, cardinalidad, conjuntos potencia, productos cartesianos **III:** Relaciones: propiedades, relaciones de equivalencia, órdenes parciales **IV:** Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa **V:** Lógica avanzada: sistemas de demostración, completitud, decidibilidad **VI:** Teoría avanzada de conjuntos: conjuntos infinitos, argumento diagonal de Cantor

Objetivos de Aprendizaje **I:** Explicar lógica proposicional y de predicados [Explicar] **II:** Construir demostraciones lógicas simples [Aplicar] **III:** Definir operaciones y relaciones básicas de conjuntos [Explicar] **IV:** Definir propiedades de las relaciones [Explicar] **V:** Definir tipos de funciones [Explicar]

Bibliografía: [Rosen, 2019, Lehman et al., 2018]

5.2 Técnicas de Demostración y Estructuras Avanzadas, 24 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones **II:** Relaciones avanzadas: clausuras, retículos **III:** Funciones avanzadas: funciones recursivas, computabilidad

Objetivos de Aprendizaje **I:** Construir demostraciones lógicas simples [Aplicar] **II:** Aplicar operaciones de conjuntos para resolver problemas [Aplicar] **III:** Determinar si una relación es una relación de equivalencia [Aplicar] **IV:** Componer funciones y encontrar inversas [Aplicar]

Bibliografía: [Rosen, 2019, Velleman, 2019]

6. Bibliografía

[Lehman et al., 2018] Lehman, E., Leighton, F. T., and Meyer, A. R. (2018). *Mathematics for Computer Science*. MIT OpenCourseWare.

[Rosen, 2019] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*. McGraw-Hill Education, 8th edition.

[Velleman, 2019] Velleman, D. J. (2019). *How to Prove It: A Structured Approach*. Cambridge University Press, 3rd edition.

[Lehman et al., 2018] Lehman, E., Leighton, F. T., and Meyer, A. R. (2018). *Mathematics for Computer Science*. MIT OpenCourseWare.

[Rosen, 2019] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*. McGraw-Hill Education, 8th edition.

[Velleman, 2019] Velleman, D. J. (2019). *How to Prove It: A Structured Approach*. Cambridge University Press, 3rd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS2B1 Desarrollo Basado en Plataformas (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 321)

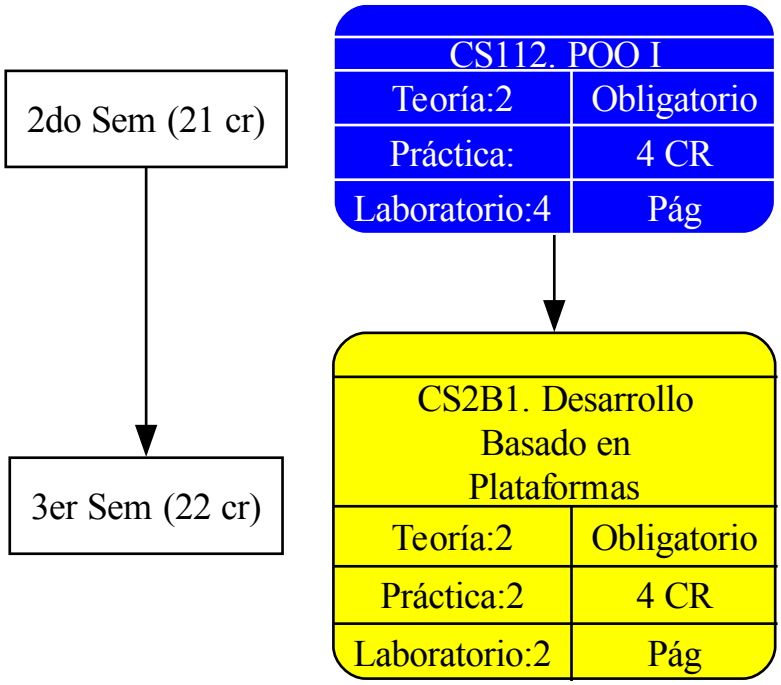


Figura 7.15: Mapa de Conexión. CS2B1 Desarrollo Basado en Plataformas

2. Fundamentación

El desarrollo basado en plataformas se enfoca en el diseño y la implementación de aplicaciones dentro de entornos informáticos específicos. Este curso utiliza el marco de Proceso y Diseño de Software (SPD) para explorar las restricciones y capacidades únicas de las plataformas Web, Móviles e Interactivas. Los estudiantes aprenderán a aplicar preocupaciones arquitectónicas compartidas y APIs específicas de plataforma para construir soluciones de software robustas, escalables y centradas en el usuario.

3. Objetivos del curso.

- Comprender los aspectos compartidos y las restricciones únicas de las plataformas modernas.
- Desarrollar aplicaciones profesionales para entornos Web y Móviles.
- Aplicar principios de computación interactiva para mejorar la experiencia del usuario.
- Diseñar arquitecturas de software que aprovechen los servicios específicos de la plataforma.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

5. Contenido

5.1 Aspectos Comunes: Plataformas, APIs y Restricciones, 5 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Visión general de las plataformas de desarrollo (es decir, web, móvil, juegos, robótica, embebida e interactiva). *a*) Entrada/sensores/dispositivos de control/dispositivos hápticos *b*) Limitaciones de recursos 1) Computacional 2) Almacenamiento de datos 3) Memoria 4) Comunicación *c*) Requisitos - seguridad, disponibilidad de tiempo de actividad, tolerancia a fallos Conceptos de Confiabilidad, Ingeniería de Confiabilidad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad *d*) Salida/actuadores/dispositivos hápticos **II:** Programación mediante Interfaces de Programación de Aplicaciones (API) específicas de la plataforma vs construcción de aplicaciones tradicional **III:** Programación bajo restricciones y requisitos de la plataforma (por ejemplo, herramientas de desarrollo disponibles, desarrollo, consideraciones de seguridad) Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad

Objetivos de Aprendizaje **I:** Enumerar las restricciones de la programación móvil [Listar/Enumerar]

Bibliografía: [Fielding, 2000, Gasca, 2021]

5.2 Aspectos Comunes: Lenguajes de Plataforma y Patrones Web, 3 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Visión general de los Lenguajes de plataforma (por ejemplo, Python, Swift, Lua, Kotlin) **II:** Técnicas para aprender y dominar un lenguaje de programación específico de la plataforma

Objetivos de Aprendizaje **I:** Enumerar las características de los lenguajes de scripting [Listar/Enumerar] **II:** Describir el modelo de tres capas (three-tier) de la programación web [Describir] **III:** Describir cómo se mantiene el estado en la programación web [Describir]

Bibliografía: [Fielding, 2000, Gasca, 2021]

5.3 Frontend, Frameworks y Estándares Web, 12 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Lenguajes de programación web (por ejemplo, HTML5, JavaScript, Typescript, CSS) **II:** Plataformas, frameworks o meta-frameworks web *a*) Servicios en la nube *b*) API, Componentes Web (Web Components) **III:** Software como Servicio (SaaS) **IV:** Estándares web como el modelo de objetos del documento (DOM), accesibilidad Accesibilidad y Diseño Inclusivo **V:** Consideraciones de Seguridad y Privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad

Objetivos de Aprendizaje **I:** Comparar y contrastar la programación web con la programación de propósito general [Comparar] **II:** Describir las diferencias entre Software-como-Servicio (SaaS) y productos de software tradicionales [Describir] **III:** Discutir cómo los estándares web impactan el desarrollo de software [Debatir] **IV:** Evaluar una aplicación web existente contra los estándares web actuales [Evaluar]

Bibliografía: [Flanagan, 2020, Duckett, 2014, Fielding, 2000]

5.4 Plataformas Móviles: Fundamentos del Desarrollo, 6 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Desarrollo con *a*) Lenguajes de programación móvil *b*) Entornos de programación móvil **II:** Restricciones de la plataforma móvil *a*) Diseño de interfaz de usuario Comprender al Usuario: Objetivos individuales e interacciones con otros *b*) Seguridad **III:** Acceso *a*) Acceso a datos a través de APIs Construcción de Consultas *b*) Diseño de puntos finales de API (endpoints) para aplicaciones móviles: trampas y consideraciones de diseño *c*) Interfaces de red y web Modelado de Datos, Fundamentos de Redes y Comunicaciones

Objetivos de Aprendizaje **I:** Comparar la programación móvil con la programación de propósito general [Comparar] **II:** Evaluar los pros y los contras del desarrollo de aplicaciones móviles nativas y multiplataforma [Evaluar]

Bibliografía: [Deitel et al., 2017, Hellman, 2013]

5.5 Plataformas Móviles: Arquitectura y Restricciones, 3 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Desarrollo *a*) Desarrollo nativo (native) versus multiplataforma (cross-platform) *b*) Patrones de diseño/arquitectura de software para aplicaciones móviles Diseño de API y Modelado, Principios y Arquitectura de Software, Calidad y Evaluación del Diseño **II:** Restricciones de la plataforma móvil *a*) Diseño de interfaz de usuario responsivo (responsive) Accesibilidad y Diseño Inclusivo *b*) Heterogeneidad y movilidad de los dispositivos *c*) Diferencias en las experiencias de usuario (por ejemplo, entre aplicaciones móviles y basadas en web) *d*) Compensación (tradeoff) entre energía y rendimiento **III:** Especificación y pruebas Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas

Objetivos de Aprendizaje **I:** Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] **II:** Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] **III:** Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]

Bibliografía: [Deitel et al., 2017, Hellman, 2013]

5.6 Plataformas Móviles: Capacidades y Computación Asíncrona, 3 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Capacidades (*affordances*) de la computación móvil *a*) Aplicaciones conscientes de la ubicación (*location-aware*) *b*) Computación impulsada por sensores (por ejemplo, giroscopio, acelerómetro, datos de salud de un reloj) *c*) Telefonía y mensajería instantánea *d*) Realidad aumentada Inmersión **II:** Computación asíncrona Algoritmos *a*) Diferencia con la programación síncrona tradicional *b*) Manejo del éxito mediante devoluciones de llamada (*callbacks*) *c*) Manejo de errores de forma asíncrona *d*) Pruebas de código asíncrono y problemas típicos en las pruebas

Objetivos de Aprendizaje **I:** Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] **II:** Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] **III:** Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]

Bibliografía: [Deitel et al., 2017, Hellman, 2013]

5.7 SPD-SEP/Interactiva, 10 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Pautas éticas al usar modelos de IA para ayudar en el periodismo y la creación de contenido **II:** Rendición de cuentas (*accountability*) por las salidas generadas por IA **III:** Comportamiento entre programadores de prompts (*prompt programmers*) y desarrolladores de IA **IV:** Confianza con el público al usar modelos de IA

Objetivos de Aprendizaje **I:** Indicar un marco para la rendición de cuentas (*accountability*) en el despliegue de modelos de IA, incluyendo documentación clara y atribución [Evaluar] **II:** Discutir códigos de conducta éticos y estándares profesionales relevantes para la programación de prompts (*prompt programming*) y el desarrollo de IA [Debatir] **III:** Crear planes y materiales de comunicación para educar al público sobre las capacidades, limitaciones y salvaguardas éticas de la IA [Crear]

Bibliografía: [Duckett, 2014, Deitel et al., 2017]

5.8 Plataformas Interactivas de Datos y Análisis, 6 hr(s)

Competencias: ABET-2,ABET-4

Tópicos: **I:** Plataformas de Cuadernos Computacionales (por ejemplo, Cuadernos Jupyter; Google Colab; R; SPSS; Observable) **II:** Plataformas de Análisis de Datos y SQL en la Nube (por ejemplo, BigQuery) Construcción de Consultas *a*) Apache Spark *b*) Visualizaciones de Datos Visualización **III:** Presentaciones Interactivas Respaldadas por Datos **IV:** Herramientas de Diseño con Bucles de Retroalimentación de Baja Latencia (*Low-Latency Feedback Loops*) *a*) Herramientas de renderizado *b*) Herramientas de diseño gráfico

Objetivos de Aprendizaje **I:** Analizar grandes conjuntos de datos de forma interactiva [Analizar] **II:** Crear una pista de acompañamiento para una interpretación musical, como la codificación en vivo (*live coding*) [Crear] **III:** Crear cuadernos computacionales convincentes que construyan una narrativa para un objetivo/historia periodística dada [Crear] **IV:** Implementar código interactivo que use un conjunto de datos y genere gráficos exploratorios [Implementar] **V:** Contextualizar los atributos de diferentes estilos de análisis de datos, como interactivo versus tubería (*pipeline*) de ingeniería [Contextualizar] **VI:** Escribir un programa usando una plataforma de computación de cuadernos (por ejemplo, búsqueda, ordenación o manipulación de grafos) [Escribir]

Bibliografía: [Flanagan, 2020]

6. Bibliografía

[Deitel et al., 2017] Deitel, P., Deitel, H., and Deitel, A. (2017). *Android How to Program*. Pearson, 3rd edition.

[Duckett, 2014] Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley.

[Fielding, 2000] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine.

[Flanagan, 2020] Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media, 7th edition.

[Gasca, 2021] Gasca, J. (2021). *Web Security for Developers*. No Starch Press.

[Hellman, 2013] Hellman, E. (2013). *Android Programming: Pushing the Limits*. Wiley.

[Deitel et al., 2017] Deitel, P., Deitel, H., and Deitel, A. (2017). *Android How to Program*. Pearson, 3rd edition.

[Duckett, 2014] Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Wiley.

[Fielding, 2000] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine.

[Flanagan, 2020] Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media, 7th edition.

[Gasca, 2021] Gasca, J. (2021). *Web Security for Developers*. No Starch Press.

[Hellman, 2013] Hellman, E. (2013). *Android Programming: Pushing the Limits*. Wiley.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA112 Cálculo II (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - MA111 Cálculo I (2do Sem-Pág 331)

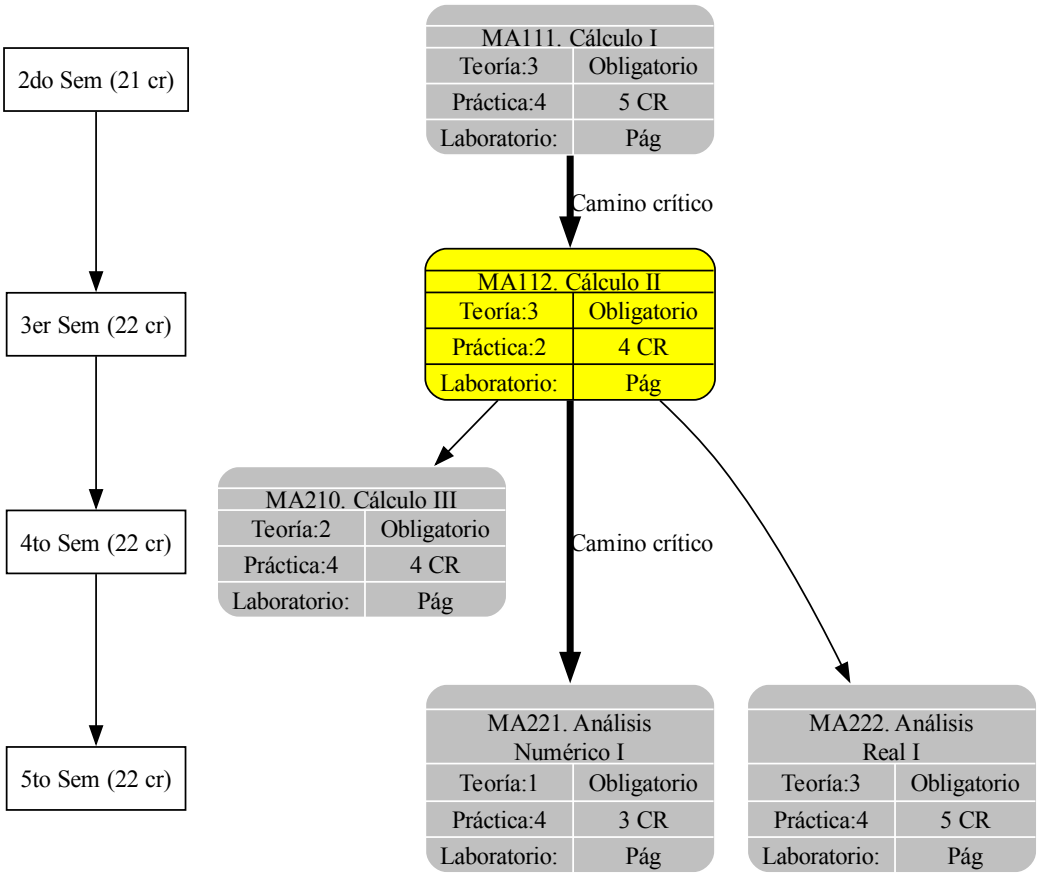


Figura 7.16: Mapa de Conexión. MA112 Cálculo II

2. Fundamentación

El curso incluye conocimientos sobre Cálculo Integral: la Integral Indefinida (antiderivada). Técnicas de Integración. Integral Definida. Integración de funciones hiperbólicas e inversas. Aplicaciones de la Integral Definida (Áreas de regiones planas, Volúmenes, Longitud, Área de Superficie). Integral Impropia. Aplicaciones en problemas que involucren a los seres vivos.

3. Objetivos del curso.

- Comprender la integral definida como el límite de las sumas de Riemann.
- Dominar diversas técnicas de integración (sustitución, partes, fracciones parciales).

- Aplicar el Teorema Fundamental del Cálculo para resolver problemas complejos.
- Utilizar las integrales para calcular áreas entre curvas y volúmenes de sólidos de revolución.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Diferenciación e Integración Numérica, 25 hr(s)

Competencias: ABET-1

Tópicos: I: Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) **II:** Extrapolación de Richardson **III:** Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) **IV:** Cuadratura gaussiana **V:** Análisis de error para integración numérica

Objetivos de Aprendizaje I: Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

5.2 Interpolación y Aproximación de Funciones, 25 hr(s)

Competencias: ABET-1

Tópicos: I: Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje I: Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

6. Bibliografía

[Stewart, 2016] Stewart, J. (2016). *Cálculo de una variable: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de una variable*. Pearson, 13th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de una variable: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de una variable*. Pearson, 13th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA121 Algebra Lineal I (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA100 Matemática Básica (1er Sem-Pág 292)

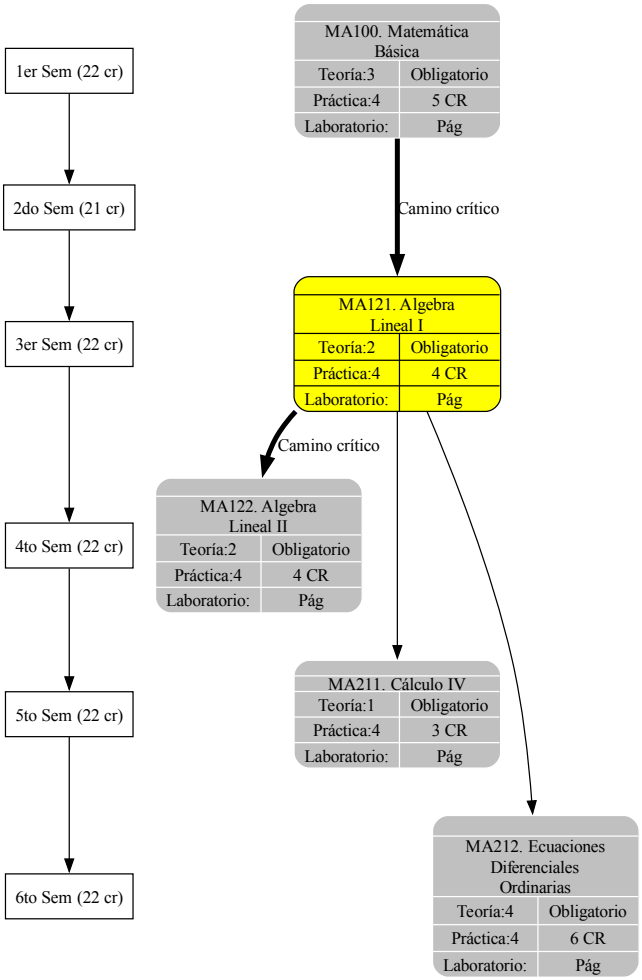


Figura 7.17: Mapa de Conexión. MA121 Algebra Lineal I

2. Fundamentación

Este curso introduce los conceptos fundamentales del álgebra lineal, proporcionando las bases matemáticas para el estudio de espacios vectoriales, transformaciones lineales y sistemas de ecuaciones lineales. Desarrolla habilidades de pensamiento abstracto y resolución de problemas mediante matrices, determinantes y vectores. El curso es fundamental para aplicaciones en ciencias, ingeniería y computación.

3. Objetivos del curso.

- Comprender los conceptos de espacios vectoriales y subespacios
- Dominar las operaciones con matrices y determinantes
- Resolver sistemas de ecuaciones lineales usando diferentes métodos
- Aplicar transformaciones lineales y calcular valores y vectores propios

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 álgebra Lineal Numérica, 25 hr(s)

Competencias: ABET-1

Tópicos: **I:** Descomposición LU y sus aplicaciones **II:** Descomposición QR y problemas de mínimos cuadrados **III:** Descomposición en Valores Singulares (SVD) y sus aplicaciones **IV:** Número de condición y estabilidad numérica de sistemas lineales **V:** Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)

Objetivos de Aprendizaje **I:** Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar] **II:** Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse] **III:** Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar] **IV:** Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar] **V:** Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar]

Bibliografía: [Lay et al., 2016, Strang, 2016]

5.2 Aritmética Computacional y Análisis de Error, 25 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aritmética de punto flotante y el estándar IEEE **II:** Fuentes de error: truncamiento, redondeo y discretización **III:** Propagación de error y número de condición **IV:** Estabilidad de algoritmos numéricos **V:** Error absoluto, relativo y directo/inverso

Objetivos de Aprendizaje **I:** Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse] **II:** Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse] **III:** Calcular el error absoluto y relativo de una aproximación [Usar] **IV:** Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar] **V:** Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar] **VI:** Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

Bibliografía: [Lay et al., 2016, Strang, 2016]

5.3 Búsqueda de Raíces y Optimización Numérica Básica, 25 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje I: Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Lay et al., 2016, Strang, 2016]

5.4 Interpolación y Aproximación de Funciones, 25 hr(s)

Competencias: ABET-1

Tópicos: I: Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje I: Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Lay et al., 2016, Anton and Rorres, 2014]

6. Bibliografía

[Anton and Rorres, 2014] Anton, H. and Rorres, C. (2014). *Álgebra Lineal con Aplicaciones*. Limusa Wiley, 11th edition.

[Lay et al., 2016] Lay, D. C., Lay, S. R., and McDonald, J. J. (2016). *Álgebra Lineal y sus Aplicaciones*. Pearson, 5th edition.

[Strang, 2016] Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition.

[Anton and Rorres, 2014] Anton, H. and Rorres, C. (2014). *Álgebra Lineal con Aplicaciones*. Limusa Wiley, 11th edition.

[Lay et al., 2016] Lay, D. C., Lay, S. R., and McDonald, J. J. (2016). *Álgebra Lineal y sus Aplicaciones*. Pearson, 5th edition.

[Strang, 2016] Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

ID101 Inglés I (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**: Ninguno

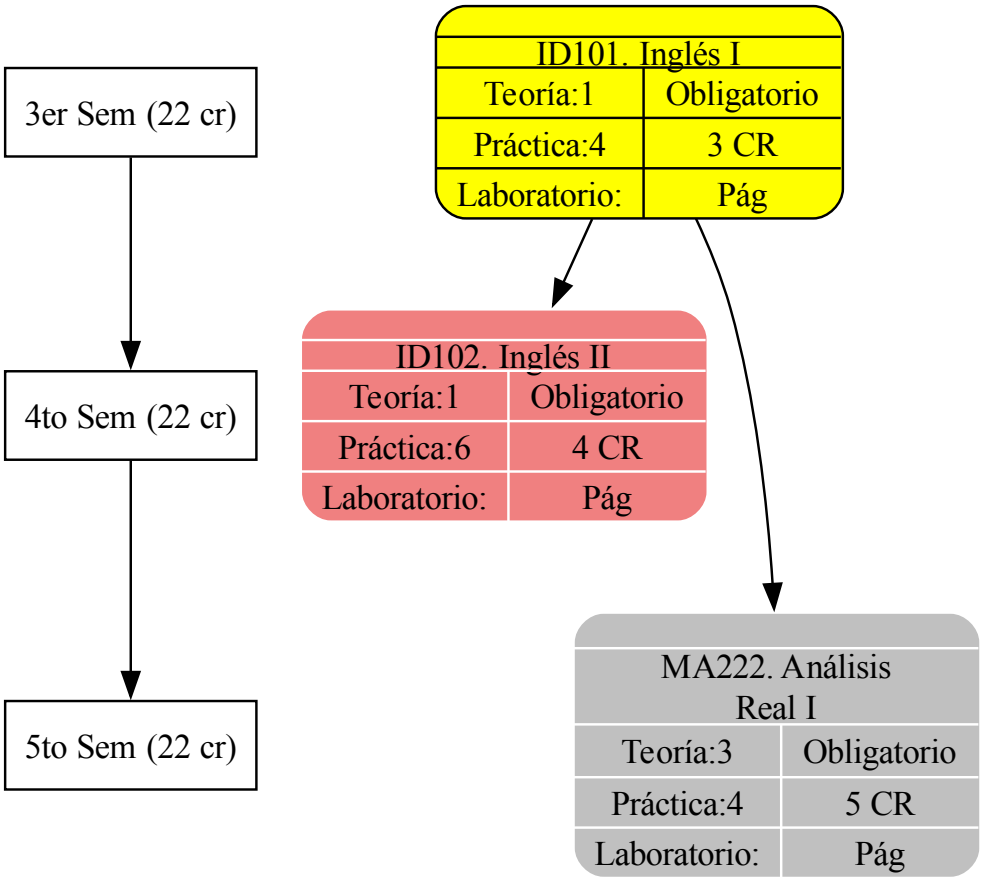


Figura 7.18: Mapa de Conexión. ID101 Inglés I

2. Fundamentación

Este curso aborda aspectos y técnicas fundamentales de la investigación académica y la redacción con el fin de proporcionar al estudiante principiante e intermedio una base sólida para trabajar en trabajos de ensayo, trabajos finales y proyectos de investigación de pregrado. El objetivo es ofrecer una introducción fácilmente apicable, pero teóricamente profunda, en el campo de la investigación académica y la escritura, que se puede entender sin literatura adicional.

3. Objetivos del curso.

- Conocer el idioma Inglés y su estructura gramatical.
- Identificar situaciones y emplear diálogos relacionados a ellas.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 Planificación y preparación, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Analiza los beneficios para usted de publicar su investigación y sugiere varios enfoques

Objetivos de Aprendizaje I: Elegir la revista adecuada y comprender lo que el editor espera de un artículo en términos de contenido estilo y estructura. **II:** Decidir el orden en el que escribir las distintas secciones. **III:** Mantener contentos a los Árbitros.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.2 Estructurar una oración: orden de las palabras, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Proporciona reglas para decidir donde colocar varios tipos de palabras dentro de una oración.

Objetivos de Aprendizaje I: Use oraciones cortas, no tome este concepto demasiado literal. **II:** Comprender el orden de las palabras.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.3 Estructura de Párrafos, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Cubre como estructurar un párrafo uniendo oraciones en el orden más lógico posible. También sugiere formas de dividir párrafos largos.

Objetivos de Aprendizaje I: Comprender ?'Cómo puedo facilitar que la lectura siga lo que estoy diciendo y comprender claramente los beneficios de mis métodos y hallazgos? **II:** Comprender ¿Cómo lograr el mínimo esfuerzo posible de mis lectores?

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.4 Romper oraciones largas, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Explique por qué y cuanto tiempo se crean las oraciones, los pros y contras de usar oraciones cortas para sus lectores y los beneficios de usar oraciones cortas para sus lectores y coautores

Objetivos de Aprendizaje I: Comprender como convertir una oración larga en oraciones cortas.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.5 Ser conciso y eliminar la redundancia, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Comienza dándote buenas razones para evitar la redundancia y luego te muestra como ser conciso

Objetivos de Aprendizaje I: No crea que el uso de términos complejos le hará parecer más inteligente **II:** Escribe usando la terminología más simple y directiva. **III:** Elimine todo lo que no sea esencial: esto permitirá que sus ideas clave se destaquen(se vean) más fácilmente.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.6 Evitar la ambigüedad, la repetición y el lenguaje vago, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Explique que es un Abstract, que tiempos y estilo elegir y cuáles son los distintos tipos de Abstracts.

Objetivos de Aprendizaje I: Entender como comenzar un resumen. **II:** Entender que incluir y como hacer que aumente su impacto. **III:** Identificar las trampas típicas de los resúmenes mal escritos.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.7 Introducción, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Diferenciar la Introducción del Resumen. **II:** Estructura y comienza una Introducción. **III:** Evite las trampas típicas. **IV:** Describe la estructura del resto del artículo.

Objetivos de Aprendizaje I: Uso de los tiempos verbales apropiados. **II:** Comprender ¿Cómo puede evaluar la calidad de mi introducción?

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.8 Revisión de la literatura, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Decidir que literatura mencionar.. **II:** Estructura tu reseña.

Objetivos de Aprendizaje I: Diferenciar su trabajo de la literatura. **II:** Resaltar el nivel de innovación en su trabajo junto con sus limitaciones.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.9 Métodos, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Describe los materiales que usaste en tus experimentos y/o los métodos que usaste para llevar a cabo tu investigación, de una manera que sea lo suficientemente detallada para permitir que otros en tu campo sigan fácilmente tu método y si lo deseas, incluso reproduzcan tu trabajo.

Objetivos de Aprendizaje I: Asegúrese de que las descripciones están completas ,también sean lo más concisas posible. **II:** Permita que sus lectores sigan fácilmente su descripción. **III:** Escribir con extrema claridad, generalmente con no más de dos pasos descritos en una oración y en un orden lógico.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.10 Resultados, 0 hr(s)

Competencias: ABET-4,ABET-6

Objetivos de Aprendizaje I: Organice los resultados en una secuencia que resalte las respuestas a los objetivos, hipótesis o preguntas que se plantea al principio del árticulo. **II:** Informe sus resultados de forma simple y clara.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.11 Discusión, 0 hr(s)

Competencias: ABET-4,ABET-6

Objetivos de Aprendizaje I: Estructura de la discusión. **II:** Suena convincente y creíble al mismo tiempo. **III:** Interprete sus resultados sin repetirlos.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.12 Conclusiones, 0 hr(s)

Competencias: ABET-4,ABET-6

Objetivos de Aprendizaje I: Comprender las habilidades clave para saber lo que los arbitros y lectores esperan encontrar en conclusiones. **II:** Comprender que no debe repetir exactamente las mismas frases e información de su resumen e introducción y al proporcionar un mensaje claro y de gran impacto para los lectores.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

6. Bibliografía

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002d] Soars, L. and John (2002d). *American Headway N 2 Student Book*. Editorial Oxford.

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002d] Soars, L. and John (2002d). *American Headway N 2 Student Book*. Editorial Oxford.

Cuarto Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS1D2 Estructuras Discretas II (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS1D1 Estructuras Discretas I (3er Sem-Pág 342)

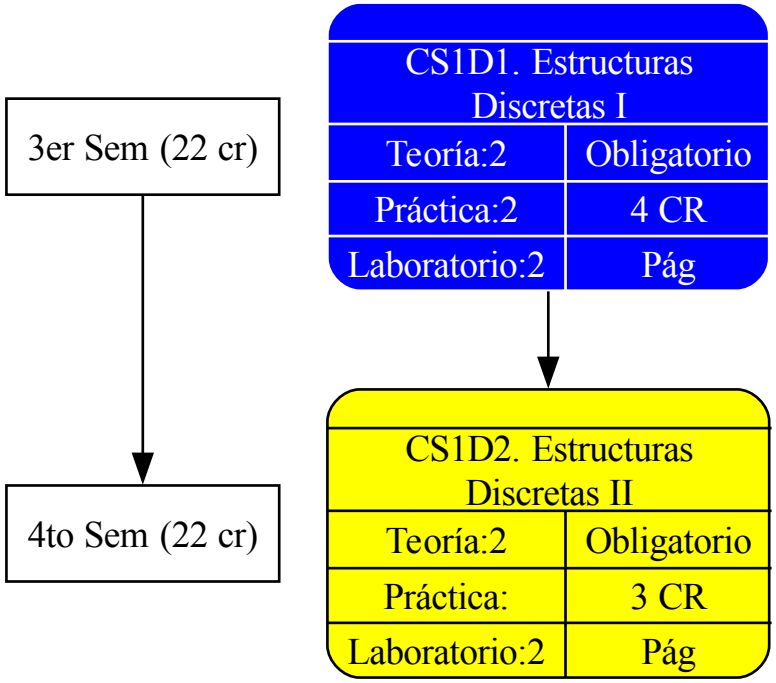


Figura 7.19: Mapa de Conexión. CS1D2 Estructuras Discretas II

2. Fundamentación

Continuando desde Estructuras Discretas I, este curso explora estructuras matemáticas más complejas como conjuntos, relaciones y funciones, las cuales constituyen la base para el modelado de datos en bases de datos y lenguajes de programación. Adicionalmente, los estudiantes estudiarán técnicas de conteo y relaciones de recurrencia, proporcionando las herramientas analíticas necesarias para estimar la complejidad computacional de los algoritmos.

3. Objetivos del curso.

- Realizar operaciones complejas sobre conjuntos, funciones y relaciones.
- Modelar formalmente relaciones de bases de datos y propiedades de funciones.
- Aplicar técnicas de conteo para analizar espacios de búsqueda y probabilidad.
- Resolver relaciones de recurrencia para analizar la eficiencia de algoritmos recursivos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Familiarizarse)

5. Contenido

5.1 Set Theory, Relations and Functions, 24 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Conjuntos: operaciones, cardinalidad, conjuntos potencia, productos cartesianos **II:** Relaciones: propiedades, relaciones de equivalencia, órdenes parciales **III:** Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa

Objetivos de Aprendizaje I: Definir operaciones y relaciones básicas de conjuntos [Evaluar] **II:** Aplicar operaciones de conjuntos para resolver problemas [Usar] **III:** Definir propiedades de las relaciones [Evaluar] **IV:** Determinar si una relación es una relación de equivalencia [Usar] **V:** Definir tipos de funciones [Evaluar] **VI:** Componer funciones y encontrar inversas [Usar]

Bibliografía: [Rosen, 2019, Scheinerman, 2012]

5.2 Counting and Recurrence, 24 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Conteo: permutaciones, combinaciones, teorema del binomio **II:** Conteo avanzado: funciones generadoras, principio de inclusión-exclusión

III: Relaciones de recurrencia: resolver recurrencias simples **IV:** Recurrencia avanzada: funciones generadoras, comportamiento asintótico

Objetivos de Aprendizaje I: Explicar permutaciones y combinaciones [Evaluar] **II:** Aplicar técnicas de conteo para resolver problemas [Usar] **III:** Definir relaciones de recurrencia [Evaluar] **IV:** Resolver relaciones de recurrencia simples [Usar]

V: Desarrollar funciones generadoras para problemas de conteo [Usar] **VI: VII:** Aplicar funciones generatrices para resolver relaciones de recurrencia complejas [Evaluar]

Bibliografía: [Rosen, 2019, Grimaldi, 2003]

6. Bibliografía

[Grimaldi, 2003] Grimaldi, R. P. (2003). *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson, 5th edition.

[Rosen, 2019] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*. McGraw-Hill Education, 8th edition.

[Scheinerman, 2012] Scheinerman, E. R. (2012). *Mathematics: A Discrete Introduction*. Cengage Learning, 3rd edition.

[Grimaldi, 2003] Grimaldi, R. P. (2003). *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson, 5th edition.

[Rosen, 2019] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications*. McGraw-Hill Education, 8th edition.

[Scheinerman, 2012] Scheinerman, E. R. (2012). *Mathematics: A Discrete Introduction*. Cengage Learning, 3rd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS210 Algoritmos y Estructuras de Datos (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS113 Programación Orientada a Objetos II (3er Sem-Pág 336)

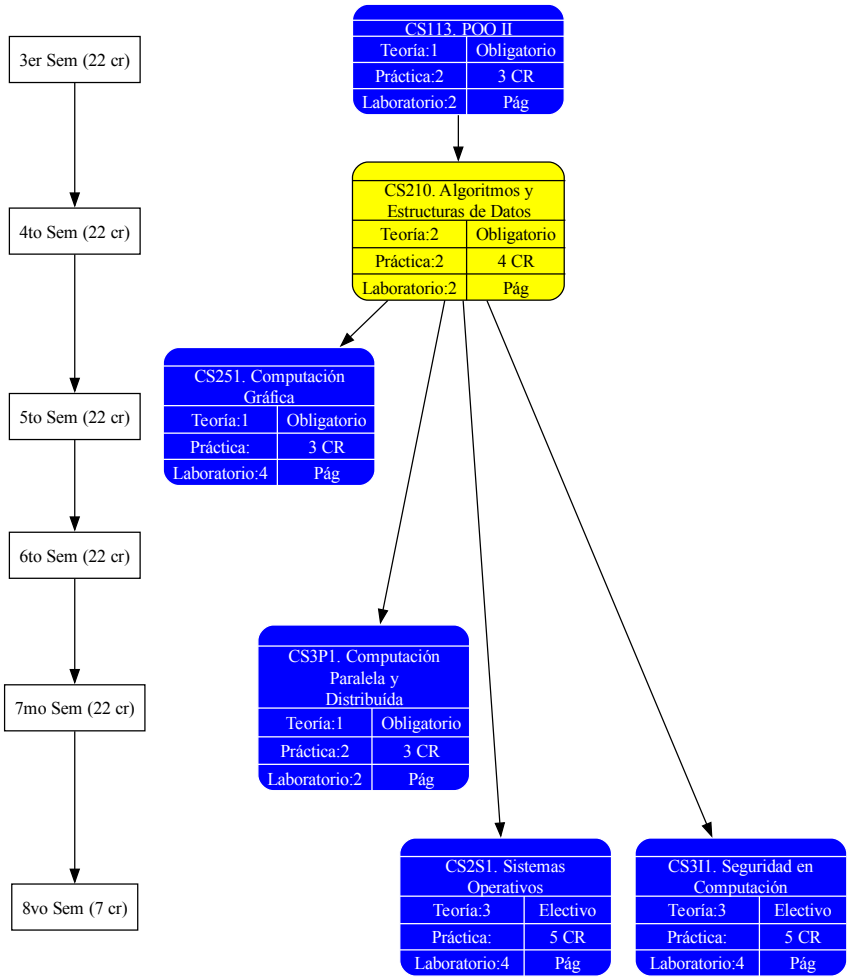


Figura 7.20: Mapa de Conexión. CS210 Algoritmos y Estructuras de Datos

2. Fundamentación

Este curso proporciona un estudio exhaustivo de las estructuras de datos y algoritmos fundamentales con énfasis en su implementación utilizando características avanzadas de C++. Basándose en conocimientos previos de C++, los estudiantes implementarán estructuras de datos genéricas usando plantillas y rasgos, asegurando flexibilidad de tipo y rendimiento. El curso cubre estructuras lineales (vectores, varias listas enlazadas), estructuras de árbol (binario, AVL, B-trees), montículos, tablas hash, tries y estructuras de datos concurrentes. Se pone especial atención en técnicas modernas de

C++ incluyendo plantillas variádicas para operaciones genéricas y programación concurrente para estructuras de datos seguras en hilos.

3. Objetivos del curso.

- Implementar estructuras de datos genéricas usando plantillas y rasgos.
- Dominar estructuras de árbol avanzadas y árboles balanceados.
- Comprender e implementar estructuras de datos concurrentes.
- Aplicar plantillas variádicas para operaciones genéricas en contenedores.
- Analizar la complejidad algorítmica de todas las estructuras implementadas.
- Diseñar estructuras de datos eficientes para restricciones de problemas específicas.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Fundamentos avanzados para estructuras de datos, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Revisión de metaprogramación con plantillas y rasgos avanzados. **II:** Rasgos de tipos y SFINAE para compilación condicional. **III:** Plantillas variádicas y paquetes de parámetros. **IV:** Reenvío perfecto y referencias universales. **V:** Expresiones lambda con capturas genéricas. **VI:** Diseño basado en políticas y CRTP (Patrón de Plantilla Curiosamente Recursivo). **VII:** Restricciones basadas en conceptos (conceptos C++20 si son aplicables).

Objetivos de Aprendizaje **I:** Diseñar rasgos de tipos para personalizar el comportamiento de estructuras de datos [Usar]. **II:** Implementar plantillas variádicas para operaciones genéricas de contenedores [Usar]. **III:** Aplicar diseño basado en políticas para crear estructuras de datos configurables [Evaluar]. **IV:** Usar reenvío perfecto para implementar constructores eficientes [Usar].

Bibliografía: [Stroustrup, 2013, Vandevoorde et al., 2017]

5.2 Estructuras de Datos Fundamentales, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: *a*) Operaciones de diccionario (insertar, eliminar, encontrar) **II:** Arreglos: *a*) Numéricos vs no numéricos, cadenas de caracteres *b*) Unidimensionales (vector) vs multidimensionales (matriz) **III:** Registros/Estructuras/Tuplas y Objetos **IV:** Listas enlazadas (por razones históricas): *a*) Simples vs Dobles y Lineales vs Circulares **V:** Pilas **VI:** Colas y deques: *a*) Cola de prioridad basada en montículo **VII:** Tablas/mapas hash: *a*) Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) **VIII:** árboles: *a*) Binarios, n-arios y árboles de búsqueda *b*) Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) **IX:** Conjuntos

Objetivos de Aprendizaje **I:** Para cada ADT/Estructura de Datos en esta unidad: *a*) Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. *b*) Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Usar] **II:** Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Usar] **III:** Implementar vector con política de asignación personalizable usando plantillas [Usar]. **IV:** Diseñar nodos de lista enlazada genéricos con gestión de memoria basada en rasgos [Usar].

V: Comparar complejidad tiempo/espacio de diferentes estructuras lineales [Evaluar]. **VI:** Aplicar plantillas variádicas para inicializar contenedores con múltiples elementos [Usar].

Bibliografía: [Cormen et al., 2009, Stroustrup, 2013]

5.3 Estructuras de Datos Fundamentales, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: árboles: *a*) Binarios, n-arios y árboles de búsqueda *b*) Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) **II:** Implementación de plantilla de árbol binario con soporte de iteradores. **III:** Árbol de búsqueda binaria con operadores de comparación basados en rasgos. **IV:** Algoritmos de recorrido de árbol (iterativo y recursivo). **V:** Árboles de expresión y sus aplicaciones. **VI:** Estructuras de nodo basadas en plantilla con carga útil configurable.

Objetivos de Aprendizaje I: Dados los requisitos para un problema, desarrollar múltiples soluciones usando varias estructuras de datos y algoritmos. Posteriormente, evaluar la idoneidad, fortalezas y debilidades seleccionando un enfoque que satisfaga mejor los requisitos [Evaluar] **II:** Explicar factores más allá de la eficiencia computacional que influyen en la elección de algoritmos, como el tiempo de programación, la mantenibilidad y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse] **III:** Implementar árbol binario genérico con orden de recorrido configurable [Usar]. **IV:** Diseñar evaluador de árbol de expresión usando patrón visitante [Usar]. **V:** Analizar complejidades de operaciones de árbol para diferentes estrategias de balanceo [Evaluar].

Bibliografía: [Cormen et al., 2009, Knuth, 1997]

5.4 Balanced and Multi-way Trees, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Implementación de árbol AVL con rasgo de balanceo basado en plantillas. **II:** Propiedades y operaciones de árbol rojo-negro. **III:** Implementación de plantilla de B-Tree para almacenamiento basado en disco. **IV:** Estructuras de árbol 2-3 y árbol 2-3-4. **V:** Implementación de trie digital (árbol de prefijos). **VI:** Árboles de sufijos y sus aplicaciones en procesamiento de cadenas. **VII:** Estrategias de división/fusión de nodos para B-trees.

Objetivos de Aprendizaje I: Implementar árboles auto-balanceados usando diseño basado en políticas [Usar]. **II:** Diseñar B-tree para aplicaciones de índice de base de datos [Usar]. **III:** Evaluar compensaciones entre diferentes estructuras de árbol balanceado [Evaluar]. **IV:** Implementar trie para coincidencia eficiente de prefijos de cadenas [Usar].

Bibliografía: [Cormen et al., 2009, Knuth, 1997, Sedgewick and Wayne, 2011]

5.5 Estructuras de Datos Fundamentales, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: árboles: *a*) Binarios, n-arios y árboles de búsqueda *b*) Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) **II:** Implementación de montículo binario usando representación de arreglo. **III:** Variantes de montículo mínimo y montículo máximo. **IV:** Algoritmos heapify y análisis de complejidad. **V:** Implementación de ADT de cola de prioridad. **VI:** Visión general de montículo binomial y montículo de Fibonacci. **VII:** Aplicaciones de montículos: algoritmo de Dijkstra, ordenamiento por montículo.

Objetivos de Aprendizaje I: Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Familiarizarse] **II:** Implementar montículo genérico con rasgo de comparación configurable [Usar]. **III:** Diseñar cola de prioridad que soporte múltiples estrategias de actualización de prioridad [Usar]. **IV:** Analizar complejidad de operaciones de montículo para diferentes tipos de montículo [Evaluar]. **V:** Aplicar montículo en implementaciones de algoritmos de grafos [Usar].

Bibliografía: [Cormen et al., 2009, Sedgewick and Wayne, 2011]

5.6 Estructuras de Datos Fundamentales, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Tablas/mapas hash: *a*) Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) **II:** Diseño de función hash y propiedades. **III:** Implementación de encadenamiento separado con listas enlazadas. **IV:** Direccionamiento abierto: sondeo lineal, sondeo cuadrático, doble hashing. **V:** Hashing perfecto y hashing perfecto mínimo. **VI:** Hashing cuco y hashing de rayuela. **VII:** Estrategias de redimensionamiento de tabla hash y gestión de factor de carga.

Objetivos de Aprendizaje I: Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Familiarizarse] **II:** Diseñar tabla hash genérica con resolución de colisiones configurable [Usar]. **III:** Implementar funciones hash personalizadas para tipos definidos por el usuario [Usar]. **IV:** Evaluar rendimiento de diferentes esquemas de hashing [Evaluar]. **V:** Aplicar tablas hash en la implementación de tabla de símbolos del compilador [Usar].

Bibliografía: [Cormen et al., 2009, Knuth, 1997]

5.7 Concurrent Data Structures, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Principios de diseño de estructuras de datos seguras para hilos. **II:** Listas enlazadas concurrentes basadas en bloqueos. **III:** Algoritmos sin bloqueo y libres de espera. **IV:** Tablas hash concurrentes con bloqueo de grano fino. **V:** Operaciones atómicas y ordenamiento de memoria. **VI:** Colas concurrentes (limitadas e ilimitadas). **VII:** Pilas concurrentes y técnicas de eliminación.

Objetivos de Aprendizaje I: Diseñar estructuras de datos seguras para hilos usando mutex y guardas de bloqueo [Usar]. **II:** Implementar lista enlazada sin bloqueo usando operaciones atómicas [Usar]. **III:** Analizar rendimiento de estructuras de datos concurrentes vs secuenciales [Evaluar]. **IV:** Aplicar colas concurrentes en patrones productor-consumidor [Usar].

Bibliografía: [Herlihy and Shavit, 2012, Williams, 2019]

5.8 Specialized Structures and Applications, 8 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Estructuras de datos persistentes. **II:** Unión-búsqueda de conjuntos disjuntos (union-find) con compresión de trayectoria. **III:** Filtros de Bloom y sus propiedades probabilísticas. **IV:** Listas por niveles para búsqueda balanceada probabilística. **V:** Árboles de segmentos y árboles de Fenwick para consultas de rango. **VI:** Estructuras de datos espaciales: árboles k-d, quadtrees. **VII:** Estructuras de datos inconscientes de la caché.

Objetivos de Aprendizaje I: Implementar estructuras de datos persistentes con compartición estructural [Usar]. **II:** Diseñar union-find con unión por rango y compresión de trayectoria [Usar]. **III:** Aplicar estructuras especializadas para resolver problemas específicos de dominio [Evaluar]. **IV:** Evaluar compensaciones espacio-tiempo en estructuras de datos probabilísticas [Evaluar].

Bibliografía: [Okasaki, 1999, Sedgewick and Wayne, 2011]

6. Bibliografía

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, 3rd edition.

[Herlihy and Shavit, 2012] Herlihy, M. and Shavit, N. (2012). *The Art of Multiprocessor Programming*. Morgan Kaufmann, revised 1st edition.

[Knuth, 1997] Knuth, D. E. (1997). *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 3rd edition.

[Okasaki, 1999] Okasaki, C. (1999). *Purely Functional Data Structures*. Cambridge University Press.

[Sedgewick and Wayne, 2011] Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Addison-Wesley, 4th edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley, 4th edition.

[Vandevoorde et al., 2017] Vandevoorde, D., Josuttis, N. M., and Gregor, D. (2017). *C++ Templates: The Complete Guide*. Addison-Wesley, 2nd edition.

[Williams, 2019] Williams, A. (2019). *C++ Concurrency in Action*. Manning Publications, 2nd edition.

[Cormen et al., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, 3rd edition.

[Herlihy and Shavit, 2012] Herlihy, M. and Shavit, N. (2012). *The Art of Multiprocessor Programming*. Morgan Kaufmann, revised 1st edition.

[Knuth, 1997] Knuth, D. E. (1997). *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 3rd edition.

[Okasaki, 1999] Okasaki, C. (1999). *Purely Functional Data Structures*. Cambridge University Press.

[Sedgewick and Wayne, 2011] Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Addison-Wesley, 4th edition.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley, 4th edition.

[Vandevoorde et al., 2017] Vandevoorde, D., Josuttis, N. M., and Gregor, D. (2017). *C++ Templates: The Complete Guide*. Addison-Wesley, 2nd edition.

[Williams, 2019] Williams, A. (2019). *C++ Concurrency in Action*. Manning Publications, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS271 Bases de Datos I (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 321)
  - CS1D1 Estructuras Discretas I (3er Sem-Pág 342)

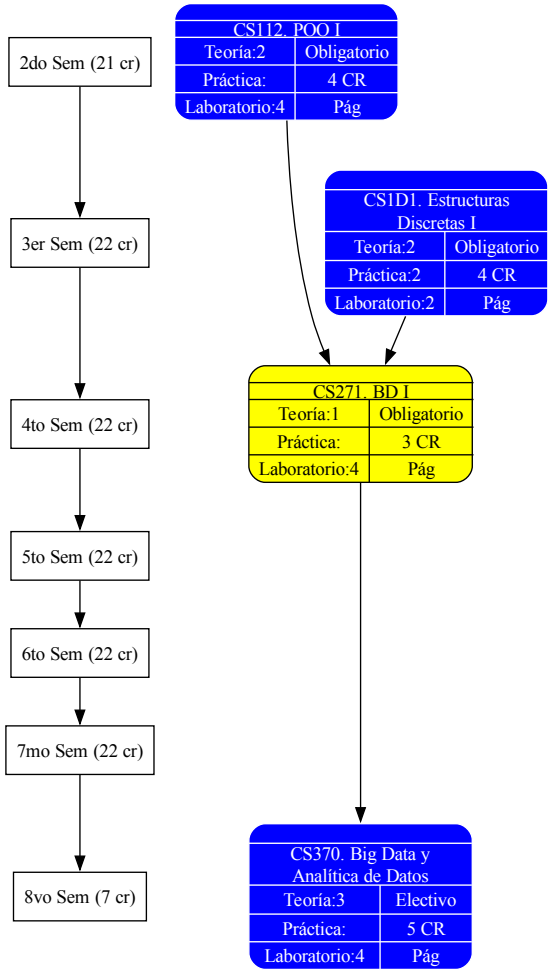


Figura 7.21: Mapa de Conexión. CS271 Bases de Datos I

2. Fundamentación

Este curso introduce conceptos y técnicas fundamentales en gestión de datos, centrándose en sistemas de bases de datos relacionales. Los estudiantes aprenderán modelado de datos usando diagramas entidad-relación, principios de diseño de bases de datos relacionales, teoría de normalización y Lenguaje de Consulta Estructurado (SQL). El curso cubre componentes centrales de sistemas de bases de datos, gestión de transacciones y procesamiento básico de consultas. Se pone énfasis en habilidades

prácticas para diseñar, implementar y consultar bases de datos, así como en comprender el rol de los datos en las organizaciones y las consideraciones éticas asociadas en la gestión de datos.

3. Objetivos del curso.

- Diseñar e implementar bases de datos relacionales normalizadas.
- Formular consultas complejas usando SQL para recuperación y manipulación de datos.
- Aplicar técnicas de modelado de datos usando diagramas entidad-relación.
- Comprender la arquitectura de sistemas de bases de datos y gestión de transacciones.
- Reconocer consideraciones éticas y de seguridad en la gestión de datos.
- Utilizar herramientas y técnicas básicas de administración de bases de datos.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Conceptos Fundamentales de Sistemas de Bases de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Propósito y ventajas de los sistemas de bases de datos **II:** Componentes de los sistemas de bases de datos **III:** Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) **IV:** Arquitectura de base de datos, independencia de datos y abstracción de datos **V:** Gestión de transacciones **VI:** Normalización **VII:** Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) **VIII:** Cómo soportar aplicaciones solo de CRUD **IX:** Bases de datos distribuidas/sistemas basados en la nube **X:** Datos estructurados, semiestructurados y no estructurados **XI:** Uso de un lenguaje de consulta declarativo **XII:** Sistemas que soportan contenido estructurado y/o de flujo

Objetivos de Aprendizaje **I:** Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Familiarizarse] **II:** Enumerar los componentes de un sistema de base de datos (relacional) [Familiarizarse] **III:** Explicar las ventajas de los sistemas de bases de datos sobre los sistemas de archivos [Usar]. **IV:** Describir la arquitectura de tres esquemas y la independencia de datos [Familiarizarse]. **V:** Identificar diferentes tipos de usuarios de bases de datos y sus roles [Familiarizarse].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Silberschatz et al., 2019]

5.2 Modelado de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Modelos conceptuales (por ejemplo, entidad-relación, diagramas UML) **II:** Conjuntos de entidades, conjuntos de relaciones y atributos. **III:** Restricciones de cardinalidad: uno-a-uno, uno-a-muchos, muchos-a-muchos. **IV:** Conjuntos de entidades débiles y relaciones identificadoras. **V:** Características extendidas E-R: especialización, generalización, agregación. **VI:** Diagramas de clases UML para modelado de datos.

Objetivos de Aprendizaje **I:** Describir los componentes del modelo E-R (o algún otro modelo no relacional) [Familiarizarse] **II:** Modelar un entorno dado usando un modelo de datos conceptual [Usar] **III:** Diseñar diagramas E-R para escenarios del mundo real [Usar]. **IV:** Convertir diagramas E-R a esquemas relacionales [Usar]. **V:** Aplicar características E-R extendidas para modelado complejo [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Elmasri and Navathe, 2015]

5.3 Bases de Datos Relacionales, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Integridad de entidad y referencial: Clave candidata, superclaves **II:** Diseño de base de datos relacional **III:** Operaciones de álgebra relacional: selección, proyección, join, unión, etc. **IV:** Restricciones de integridad: dominio, clave, entidad, referencial. **V:** Cálculo relacional: cálculo de tuplas y de dominio. **VI:** Mapeo de diagramas E-R a esquemas relacionales.

Objetivos de Aprendizaje **I:** Describir las características definitorias detrás del modelo de datos relacional [Familiarizarse] **II:** Comentar sobre la diferencia entre una clave foránea y una superclave [Usar] **III:** Expresar consultas usando álgebra relacional [Usar]. **IV:** Diseñar esquemas relacionales con restricciones apropiadas [Usar]. **V:** Mapear diagramas E-R complejos a relaciones normalizadas [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Elmasri and Navathe, 2015]

5.4 Construcción de Consultas, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Formación de Consultas SQL: *a*) Ejecución interactiva de SQL *b*) Ejecución programática de una consulta SQL **II:** Lenguaje de definición de datos: CREATE, ALTER, DROP. **III:** Tipos de datos y restricciones de dominio. **IV:** Creación y gestión de índices. **V:** Definición de vistas y vistas materializadas. **VI:** Diseño e implementación de esquemas de base de datos.

Objetivos de Aprendizaje **I:** Componer consultas SQL que incorporen selección, proyección, unión, intersección, diferencia de conjuntos y división de conjuntos [Usar] **II:** Definir, en SQL, un esquema de relación, incluyendo todas las restricciones de integridad y activadores de eliminación/actualización [Usar] **III:** Crear y modificar esquemas de base de datos usando SQL DDL [Usar]. **IV:** Implementar restricciones de integridad a nivel de esquema [Usar]. **V:** Diseñar y crear índices apropiados para rendimiento [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Beaulieu, 2009]

5.5 Construcción de Consultas, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Consultas SQL complejas: subconsultas anidadas, subconsultas correlacionadas. **II:** Funciones de agregación y operaciones GROUP BY. **III:** Operaciones de conjunto: UNION, INTERSECT, EXCEPT. **IV:** Operaciones JOIN: inner, outer, natural, cross joins. **V:** Manipulación de datos: INSERT, UPDATE, DELETE. **VI:** Control de transacciones: COMMIT, ROLLBACK, SAVEPOINT.

Objetivos de Aprendizaje **I:** Determinar cuándo una consulta SQL anidada está correlacionada o no [Usar] **II:** Componer una consulta SQL para actualizar una tupla en una relación [Usar] **III:** Formular consultas complejas con múltiples joins y subconsultas [Usar]. **IV:** Aplicar agregación para análisis de datos y generación de informes [Usar]. **V:** Implementar operaciones de modificación de datos con control de transacciones [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Beaulieu, 2009]

5.6 Bases de Datos Relacionales, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Introducción a la teoría de dependencias funcionales **II:** Teoría de Normalización: *a*) Descomposición de un esquema; propiedades de unión sin pérdida y preservación de dependencias de una descomposición *b*) Formas normales (BCNF) *c*) Desnormalización (para eficiencia) **III:** Dependencias funcionales y reglas de inferencia. **IV:** Formas normales: 1FN, 2FN, 3FN, FNBC. **V:** Algoritmos de descomposición: sin pérdida de join y que preserva dependencias. **VI:** Consideraciones prácticas en normalización.

Objetivos de Aprendizaje **I:** Describir cómo la teoría de dependencias funcionales generaliza la noción de clave [Familiarizarse] **II:** Detectar qué forma normal produce una descomposición dada [Usar] **III:** Identificar dependencias funcionales en esquemas relacionales [Usar]. **IV:** Normalizar relaciones a FNBC [Evaluar]. **V:** Evaluar descomposiciones para sin pérdida de join y preservación de dependencias [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Elmasri and Navathe, 2015]

5.7 Internos de DBMS, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Gestión de Transacciones: *a*) Niveles de Aislamiento *b*) ACID *c*) Serializabilidad *d*) Transacciones Distribuidas **II:** Control de Concurrencia: *a*) Bloqueo en 2 Fases *b*) Estrategias de manejo de bloqueos mutuos *c*) Modelos de consistencia basados en quórum **III:** Propiedades ACID: Atomicidad, Consistencia, Aislamiento, Durabilidad. **IV:** Serializabilidad y serializabilidad por conflicto. **V:** Control de concurrencia basado en bloqueos: bloqueo en dos fases. **VI:** Detección y prevención de interbloqueos. **VII:** Niveles de aislamiento y sus compensaciones.

Objetivos de Aprendizaje **I:** Describir las cuatro propiedades para un gestor de transacciones correcto [Familiarizarse] **II:** Esbozar el principio de serializabilidad [Familiarizarse] **III:** Explicar las propiedades ACID y su importancia [Usar]. **IV:** Analizar programaciones de transacciones para serializabilidad [Evaluar]. **V:** Aplicar protocolos de bloqueo para control de concurrencia [Usar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Bernstein et al., 1987]

5.8 Seguridad y Privacidad de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Diferencias entre seguridad de datos y privacidad de datos **II:** Información de identificación personal (PII) y su protección **III:** Consideraciones éticas para asegurar la seguridad y privacidad de los datos **IV:** Seguridad de bases de datos: autenticación, autorización, encriptación. **V:** Ataques de inyección SQL y prevención. **VI:** Introducción a bases de datos NoSQL y big data. **VII:** Gobernanza de datos y cumplimiento normativo.

Objetivos de Aprendizaje **I:** Describir las diferencias en los objetivos para seguridad de datos y privacidad de datos [Familiarizarse] **II:** Identificar y mitigar riesgos asociados con diferentes enfoques para proteger datos [Usar] **III:** Implementar medidas básicas de seguridad de bases de datos [Usar]. **IV:** Prevenir vulnerabilidades de inyección SQL [Evaluar]. **V:** Discutir consideraciones éticas en la recolección y uso de datos [Evaluar].

Bibliografía: [Ramakrishnan and Gehrke, 2002, Elmasri and Navathe, 2015]

6. Bibliografía

[Beaulieu, 2009] Beaulieu, A. (2009). *Learning SQL*. O'Reilly Media, 2nd edition.

[Bernstein et al., 1987] Bernstein, P. A., Hadzilacos, V., and Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Addison-Wesley.

[Elmasri and Navathe, 2015] Elmasri, R. and Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson, 7th edition.

[Ramakrishnan and Gehrke, 2002] Ramakrishnan, R. and Gehrke, J. (2002). *Database Management Systems*. McGraw-Hill, 3rd edition.

[Silberschatz et al., 2019] Silberschatz, A., Korth, H. F., and Sudarshan, S. (2019). *Database System Concepts*. McGraw-Hill, 7th edition.

[Beaulieu, 2009] Beaulieu, A. (2009). *Learning SQL*. O'Reilly Media, 2nd edition.

[Bernstein et al., 1987] Bernstein, P. A., Hadzilacos, V., and Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Addison-Wesley.

[Elmasri and Navathe, 2015] Elmasri, R. and Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson, 7th edition.

[Ramakrishnan and Gehrke, 2002] Ramakrishnan, R. and Gehrke, J. (2002). *Database Management Systems*. McGraw-Hill, 3rd edition.

[Silberschatz et al., 2019] Silberschatz, A., Korth, H. F., and Sudarshan, S. (2019). *Database System Concepts*. McGraw-Hill, 7th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA122 Algebra Lineal II (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA121 Algebra Lineal I (3er Sem-Pág 352)

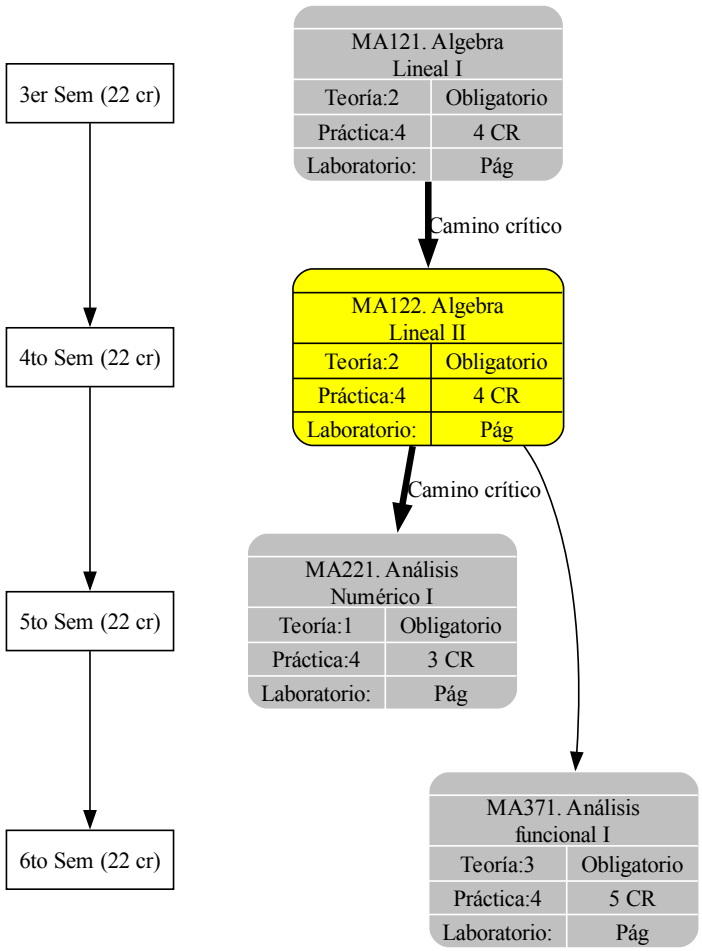


Figura 7.22: Mapa de Conexión. MA122 Algebra Lineal II

2. Fundamentación

Este curso profundiza en los conceptos avanzados del álgebra lineal, enfocándose en espacios con producto interno, transformaciones lineales avanzadas, descomposiciones matriciales y aplicaciones computacionales. Desarrolla habilidades para analizar estructuras algebraicas complejas y aplicar métodos matriciales a problemas de ciencia de datos, procesamiento de señales y optimización.

3. Objetivos del curso.

- Dominar los espacios con producto interno y sus propiedades

- Aplicar descomposiciones matriciales avanzadas (SVD, QR)
- Analizar transformaciones lineales en espacios con producto interno
- Resolver problemas de aproximación y mínimos cuadrados

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 álgebra Lineal Numérica, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Descomposición LU y sus aplicaciones **II:** Descomposición QR y problemas de mínimos cuadrados **III:** Descomposición en Valores Singulares (SVD) y sus aplicaciones **IV:** Número de condición y estabilidad numérica de sistemas lineales **V:** Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)

Objetivos de Aprendizaje **I:** Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar] **II:** Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse] **III:** Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar] **IV:** Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar] **V:** Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

5.2 Aritmética Computacional y Análisis de Error, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aritmética de punto flotante y el estándar IEEE **II:** Fuentes de error: truncamiento, redondeo y discretización **III:** Propagación de error y número de condición **IV:** Estabilidad de algoritmos numéricos **V:** Error absoluto, relativo y directo/inverso

Objetivos de Aprendizaje **I:** Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse] **II:** Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse] **III:** Calcular el error absoluto y relativo de una aproximación [Usar] **IV:** Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar] **V:** Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar] **VI:** Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

5.3 Interpolación y Aproximación de Funciones, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje **I:** Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

5.4 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs), 16 hr(s)

Competencias: ABET-1

Tópicos: I: Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje I: Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Strang, 2016, Trefethen and III, 1997]

6. Bibliografía

[Lay et al., 2016] Lay, D. C., Lay, S. R., and McDonald, J. J. (2016). *Álgebra Lineal y sus Aplicaciones*. Pearson, 5th edition.

[Strang, 2016] Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition.

[Trefethen and III, 1997] Trefethen, L. N. and III, D. B. (1997). *Numerical Linear Algebra*. SIAM.

[Lay et al., 2016] Lay, D. C., Lay, S. R., and McDonald, J. J. (2016). *Álgebra Lineal y sus Aplicaciones*. Pearson, 5th edition.

[Strang, 2016] Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th edition.

[Trefethen and III, 1997] Trefethen, L. N. and III, D. B. (1997). *Numerical Linear Algebra*. SIAM.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA210 Cálculo III (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA112 Cálculo II (3er Sem-Pág 350)

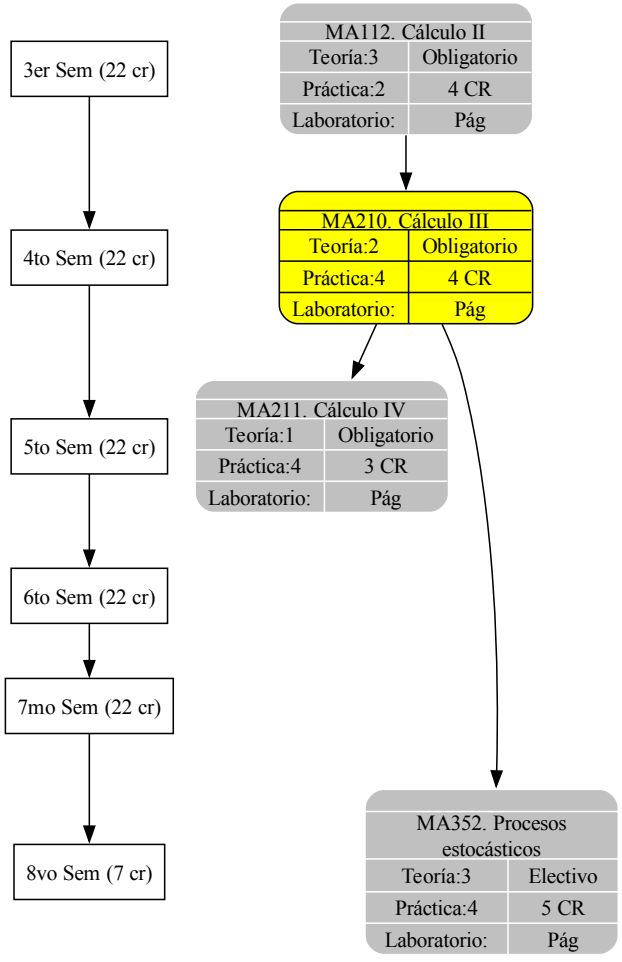


Figura 7.23: Mapa de Conexión. MA210 Cálculo III

2. Fundamentación

La naturaleza del curso es teórica-práctica, su propósito es desarrollar en los estudiantes un marco conceptual y metodológico orientado a proporcionar una sólida formación matemática para la comprensión de todo fenómeno natural. Comprende las siguientes unidades temáticas: Cálculo diferencial en varias variables. Cálculo integral en varias variables. Cálculo vectorial.

3. Objetivos del curso.

- Comprender las funciones de varias variables y sus derivadas parciales

- Aplicar el cálculo diferencial e integral en tres dimensiones
- Resolver problemas de optimización con múltiples variables
- Dominar las integrales múltiples y sus aplicaciones

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Diferenciación e Integración Numérica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) **II:** Extrapolación de Richardson **III:** Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) **IV:** Cuadratura gaussiana **V:** Análisis de error para integración numérica

Objetivos de Aprendizaje **I:** Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

5.2 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs), 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de Euler y sus variantes (explícito, implícito) **II:** Métodos de Runge-Kutta **III:** Métodos multipaso (Adams-Bashforth, Adams-Moulton) **IV:** Análisis de estabilidad y convergencia para solucionadores de EDOs **V:** Sistemas de EDOs y EDOs de orden superior

Objetivos de Aprendizaje **I:** Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar] **II:** Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar] **III:** Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar] **IV:** Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

5.3 Interpolación y Aproximación de Funciones, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje **I:** Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

5.4 Búsqueda de Raíces y Optimización Numérica Básica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje **I:** Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

6. Bibliografía

[Stewart, 2016] Stewart, J. (2016). *Cálculo de varias variables: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de varias variables*. Pearson, 13th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de varias variables: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Thomas et al., 2014] Thomas, G. B., Weir, M. D., and Hass, J. (2014). *Cálculo de varias variables*. Pearson, 13th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

ID102 Inglés II (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 6 horas;
- **Prerrequisitos**:
  - ID101 Inglés I (3er Sem-Pág 355)

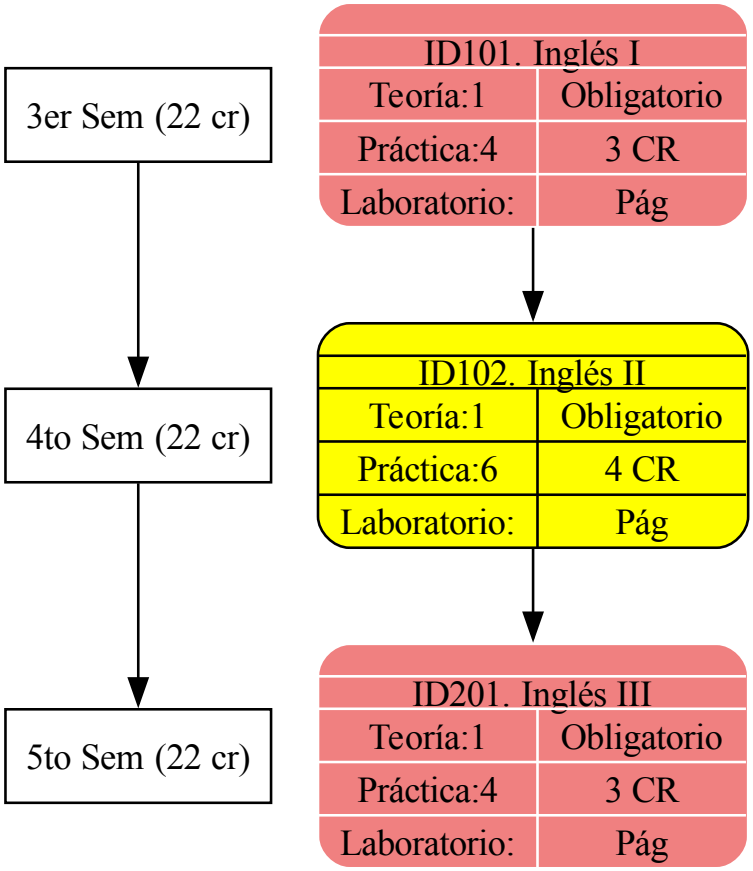


Figura 7.24: Mapa de Conexión. ID102 Inglés II

2. Fundamentación

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.

3. Objetivos del curso.

- Desarrollar la capacidad de hablar fluídamente el idioma.
- Incrementar el vocabulario y el manejo de expresiones simples.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 How long ago?, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Pasado Simple **II:** Oraciones Negativas con ago. **III:** Conjunciones **IV:** Expresiones de Tiempo en pasado **V:** Relaciones y símbolos fonéticos **VI:** Expresiones para dar la fecha

Objetivos de Aprendizaje I: Al terminar la octava unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo pasado es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de fechas y números en orden.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.2 Food you like!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Sustantivos Contables y No Contables **II:** Expresiones con Would like y I'd like **III:** Cuantificadores **IV:** Comidas alrededor del mundo **V:** Pedidos formales **VI:** Cartas formales

Objetivos de Aprendizaje I: Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos, los utilizan en situaciones varias. Expresar situaciones y estados relacionados con cantidades. Explica y aplica vocabulario de comidas y bebidas.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.3 The world of work, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Adjetivos **II:** Oraciones con Adjetivos Comparativos. **III:** Oraciones con Adjetivos Superlativos **IV:** Ciudades y el campo **V:** Indicaciones de dirección

Objetivos de Aprendizaje I: Al terminar la décima unidad, los alumnos habiendo reconocido las características de los adjetivos, utilizan éstos para hacer comparaciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Utilizarán conjunciones para unir ideas tipo.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.4 Looking good!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Presente Continuo **II:** Oraciones Afirmativas, Negativas y Preguntas **III:** Uso de Whose **IV:** Pronombres Posesivos **V:** Ropa y colores **VI:** Expresiones a usar en tiendas de ropa **VII:** Símbolos fonéticos.

Objetivos de Aprendizaje **I:** Al terminar la décimo primera unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente Progresivo. Expresan ideas de posesión con respecto a la ropa y los colores.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.5 Life is an adventure!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Uso de going to **II:** Oraciones en Tiempo Futuro **III:** Expresiones de Cantidad. **IV:** Verbos de acción **V:** Vocabulario del clima **VI:** Expresiones de Sugerencia **VII:** Escribir una postal

Objetivos de Aprendizaje **I:** Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar infinitivos de propósito. Adquirirán vocabulario para describir el clima. Se presentará expresiones para hacer y pedir sugerencias.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.6 You‘re pretty smart!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Formas de Preguntas **II:** Adverbios y Adjetivos **III:** Vocabulario descripción de sentimientos **IV:** Expresiones para viajes en tren **V:** Redacción de historias cortas **VI:** Lecturas.

Objetivos de Aprendizaje **I:** Al finalizar la décimo tercera unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas diversas, realizarán trabajos aplicativos en contextos adecuados. Enfatizan la diferencia entre adjetivos y adverbios. Describen sentimientos. Utilizan expresiones para coger un tren. Asumen la idea se sufijos y prefijos.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

5.7 Have you ever?, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Presente Perfecto **II:** Expresiones con never, ever y yet **III:** Vocabulario verbos en Participio pasado **IV:** Expresiones para viajes en avión **V:** Redacción de cartas de agradecimiento **VI:** Lecturas

Objetivos de Aprendizaje **I:** Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Presente Perfecto experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre pasado simple y presente perfecto. Describen acciones con never, ever y yet. Utilizan expresiones para utilizar en un aeropuerto.

Bibliografía: [Soars and John, 2002a, Cambridge, 2006, MacGrew, 1999]

6. Bibliografía

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002a] Soars, L. and John (2002a). *American Headway N 1 Student Book*. Editorial Oxford.

Quinto Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS251 Computación Gráfica (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 361)

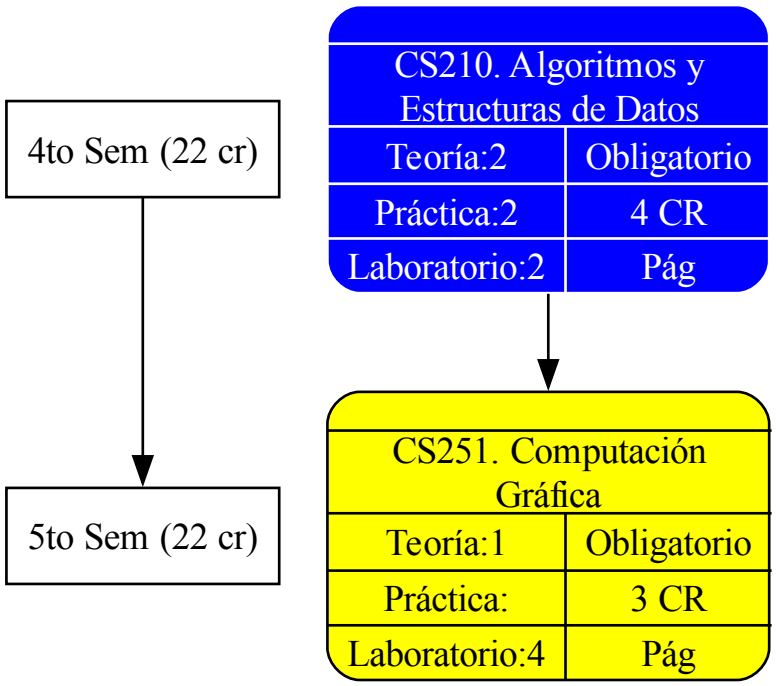


Figura 7.25: Mapa de Conexión. CS251 Computación Gráfica

2. Fundamentación

Los Gráficos por Computadora son un área fundamental de la Ciencia de la Computación que se enfoca en las técnicas matemáticas y computacionales para generar y manipular imágenes. Este curso proporciona a los estudiantes los fundamentos del modelado 2D y 3D, las canalizaciones de renderizado y los principios físicos de la luz y el color, permitiéndoles desarrollar aplicaciones visuales, simulaciones y sistemas interactivos de alto rendimiento.

3. Objetivos del curso.

- Comprender la canalización de gráficos y las transformaciones de sistemas de coordenadas.
- Implementar algoritmos de renderizado para visualización realista y en tiempo real.
- Aplicar modelado matemático para crear y manipular objetos 3D.

4. Competencias Generales (*Outcomes*)

5. Contenido

5.1 Conceptos Fundamentales de Gráficos y Técnicas Interactivas, 16 hr(s)

Competencias:

Tópicos: **I:** Descripción general del pipeline de gráficos por computadora y sistemas interactivos **II:** Conceptos básicos de renderizado: rasterización, trazado de rayos (*ray tracing*), modelos de sombreado **III:** Teoría del color y espacios de color **IV:** Sistemas de coordenadas y transformaciones **V:** Técnicas básicas de interacción: dispositivos de entrada, manejo de eventos **VI:** Conceptos avanzados de renderizado: iluminación global (*global illumination*), renderizado basado en física **VII:** Hardware gráfico y técnicas de aceleración **VIII:** Percepción humana y cognición visual

Objetivos de Aprendizaje **I:** Explicar las etapas básicas del pipeline de gráficos por computadora [Explicar] **II:** Describir técnicas básicas de renderizado como rasterización y trazado de rayos [Describir] **III:** Aplicar teoría del color para diseñar gráficos visualmente efectivos [Aplicar] **IV:** Implementar transformaciones de coordenadas para gráficos 2D y 3D [Implementar] **V:** Diseñar aplicaciones interactivas básicas utilizando dispositivos de entrada comunes y manejo de eventos [Diseñar] **VI:** Comparar técnicas avanzadas de renderizado como iluminación global y renderizado basado en física [Comparar] **VII:** Analizar el rol del hardware gráfico en acelerar tareas de renderizado [Analizar] **VIII:** Explicar cómo la percepción humana influye en el diseño de gráficos e interacción [Explicar]

Bibliografía: [Marschner and Shirley, 2021, Hearn et al., 2014]

5.2 Renderizado Aplicado y Técnicas, 16 hr(s)

Competencias:

Tópicos: **I:** Algoritmos de rasterización: conversión por barrido (*scan conversion*), antialiasing **II:** Trazado de rayos (*ray tracing*): estructuras de aceleración, muestreo (*sampling*), sombreado **III:** Mapeo de texturas (*texture mapping*) y filtrado **IV:** Modelos de sombreado: Lambertiano, Phong, BRDF **V:** Técnicas avanzadas: mapeo de sombras (*shadow mapping*), mapeo ambiental (*environment mapping*), renderizado diferido (*deferred rendering*)

Objetivos de Aprendizaje **I:** Implementar algoritmos de rasterización para renderizado de primitivas [Implementar] **II:** Implementar trazado de rayos básico con estructuras de aceleración [Implementar] **III:** Aplicar técnicas de mapeo de texturas y filtrado para mejorar el realismo visual [Aplicar] **IV:** Comparar diferentes modelos de sombreado y sus aplicaciones [Comparar] **V:** Implementar técnicas avanzadas de renderizado como mapeo de sombras o renderizado diferido [Implementar]

Bibliografía: [Marschner and Shirley, 2021, Hearn et al., 2014, Angel and Shreiner, 2014]

5.3 Modelado Geométrico, 16 hr(s)

Competencias:

Tópicos: **I:** Representaciones geométricas: polígonos, curvas, superficies, formas implícitas **II:** Técnicas de modelado: curvas spline, superficies de subdivisión, geometría sólida constructiva **III:** Modelado procedural y generación **IV:** Algoritmos geométricos: intersección, distancia, descomposición

Objetivos de Aprendizaje **I:** Describir diferentes representaciones geométricas y sus compensaciones (trade-offs) [Describir] **II:** Implementar técnicas de modelado como curvas spline o superficies de subdivisión [Implementar] **III:** Crear modelos procedurales usando algoritmos o reglas [Crear] **IV:** Aplicar algoritmos geométricos para resolver problemas como pruebas de intersección o procesamiento de mallas (mesh) [Aplicar]

Bibliografía: [Marschner and Shirley, 2021, Angel and Shreiner, 2014]

5.4 Visualización, 16 hr(s)

Competencias:

Tópicos: **I:** Principios de visualización de datos: percepción, cognición y codificación efectiva **II:** Tipos de visualizaciones: científicas, de información, geográficas **III:** Técnicas de visualización para datos multivariados, temporales y jerárquicos **IV:** Visualización interactiva y exploración **V:** Evaluación de la efectividad de la visualización

Objetivos de Aprendizaje **I:** Aplicar principios de percepción y cognición para diseñar visualizaciones efectivas [Aplicar] **II:** Seleccionar técnicas de visualización apropiadas para diferentes tipos de datos y tareas [Evaluar] **III:** Implementar visualizaciones interactivas para exploración de datos [Implementar] **IV:** Evaluar la efectividad de una visualización usando métodos empíricos [Evaluar]

Bibliografía: [Marschner and Shirley, 2021]

5.5 Sombreado y Renderizado Avanzado, 12 hr(s)

Competencias:

Tópicos: **I:** Sombreado avanzado: sombreado basado en física, dispersión subsuperficial (subsurface scattering) **II:** Iluminación global (global illumination): radiosidad, mapeo de fotones (photon mapping), trazado de caminos (path tracing) **III:** Técnicas de renderizado en tiempo real: nivel de detalle (level of detail), culling, oclusión **IV:** Programación GPU y sombreadores (shaders)

Objetivos de Aprendizaje **I:** Implementar modelos de sombreado avanzados para materiales realistas [Implementar] **II:** Comparar algoritmos de iluminación global y sus compensaciones [Comparar] **III:** Optimizar el renderizado en tiempo real usando técnicas como nivel de detalle y *culling* [Crear] **IV:** Desarrollar sombreadores GPU para efectos de renderizado personalizados [Crear]

Bibliografía: [Marschner and Shirley, 2021, Angel and Shreiner, 2014]

6. Bibliografía

[Angel and Shreiner, 2014] Angel, E. and Shreiner, D. (2014). *Interactive Computer Graphics: A Top-Down Approach with WebGL*. Pearson, 7th edition.

[Hearn et al., 2014] Hearn, D., Baker, M. P., and Carithers, W. (2014). *Computer Graphics with OpenGL*. Pearson, 4th edition.

[Marschner and Shirley, 2021] Marschner, S. and Shirley, P. (2021). *Fundamentals of Computer Graphics*. A K Peters/CRC Press, 5th edition.

[Angel and Shreiner, 2014] Angel, E. and Shreiner, D. (2014). *Interactive Computer Graphics: A Top-Down Approach with WebGL*. Pearson, 7th edition.

[Hearn et al., 2014] Hearn, D., Baker, M. P., and Carithers, W. (2014). *Computer Graphics with OpenGL*. Pearson, 4th edition.

[Marschner and Shirley, 2021] Marschner, S. and Shirley, P. (2021). *Fundamentals of Computer Graphics*. A K Peters/CRC Press, 5th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA211 Cálculo IV (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA210 Cálculo III (4to Sem-Pág 374)
  - MA121 Algebra Lineal I (3er Sem-Pág 352)

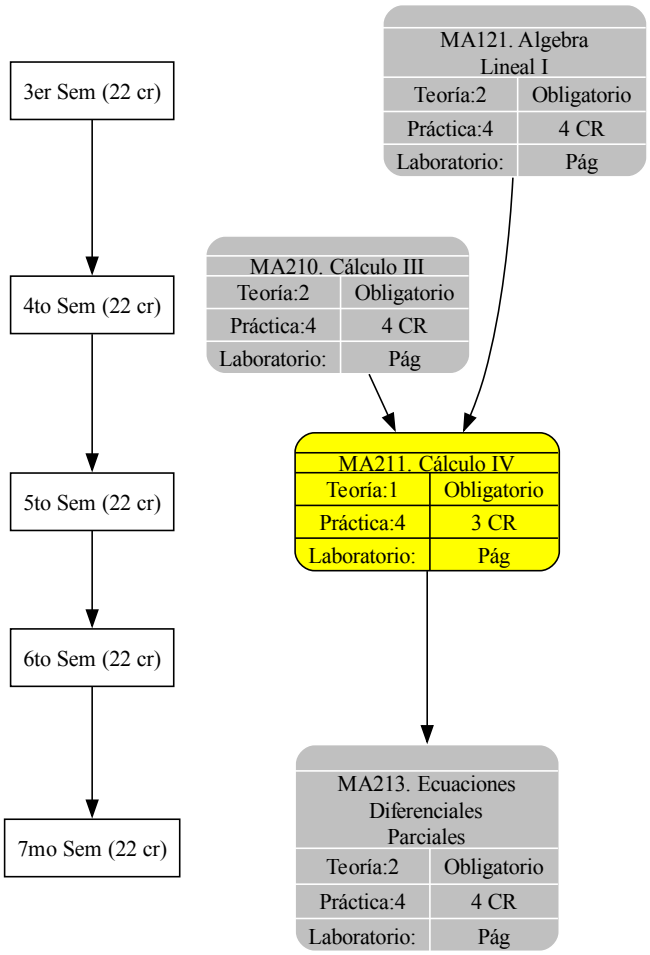


Figura 7.26: Mapa de Conexión. MA211 Cálculo IV

2. Fundamentación

La naturaleza del curso es teórica-práctica, su propósito es desarrollar un marco conceptual y metodológico orientado a proporcionar una sólida formación matemática para la comprensión de todo fenómeno natural. Comprende las siguientes unidades temáticas: Sucesiones en R^n, C, (R^m, R). Derivadas direccionales y Derivación implícita. Isometrías de R^n, cónicas y superficies cuadráticas. Campos vectoriales.

3. Objetivos del curso.

- Comprender los conceptos de campos vectoriales y operadores diferenciales
- Aplicar integrales de línea y de superficie a problemas físicos
- Dominar los teoremas fundamentales del cálculo vectorial
- Resolver problemas de flujo, circulación y potenciales vectoriales

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Diferenciación e Integración Numérica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) **II:** Extrapolación de Richardson **III:** Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) **IV:** Cuadratura gaussiana **V:** Análisis de error para integración numérica

Objetivos de Aprendizaje **I:** Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Marsden and Tromba, 2003, Stewart, 2016]

5.2 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs), 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de Euler y sus variantes (explícito, implícito) **II:** Métodos de Runge-Kutta **III:** Métodos multipaso (Adams-Bashforth, Adams-Moulton) **IV:** Análisis de estabilidad y convergencia para solucionadores de EDOs **V:** Sistemas de EDOs y EDOs de orden superior

Objetivos de Aprendizaje **I:** Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar] **II:** Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar] **III:** Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar] **IV:** Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar]

Bibliografía: [Marsden and Tromba, 2003, Stewart, 2016]

5.3 Interpolación y Aproximación de Funciones, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje **I:** Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Marsden and Tromba, 2003, Stewart, 2016]

5.4 Búsqueda de Raíces y Optimización Numérica Básica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje **I:** Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Marsden and Tromba, 2003, Stewart, 2016]

6. Bibliografía

[Marsden and Tromba, 2003] Marsden, J. E. and Tromba, A. J. (2003). *Cálculo Vectorial*. Pearson, 5th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de varias variables: Trascendentes tempranas*. Cengage Learning, 8th edition.

[Marsden and Tromba, 2003] Marsden, J. E. and Tromba, A. J. (2003). *Cálculo Vectorial*. Pearson, 5th edition.

[Stewart, 2016] Stewart, J. (2016). *Cálculo de varias variables: Trascendentes tempranas*. Cengage Learning, 8th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA221 Análisis Numérico I (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA112 Cálculo II (3er Sem-Pág 350)
  - MA122 Algebra Lineal II (4to Sem-Pág 371)
  - AI161 IA Aplicada (2do Sem-Pág 326)

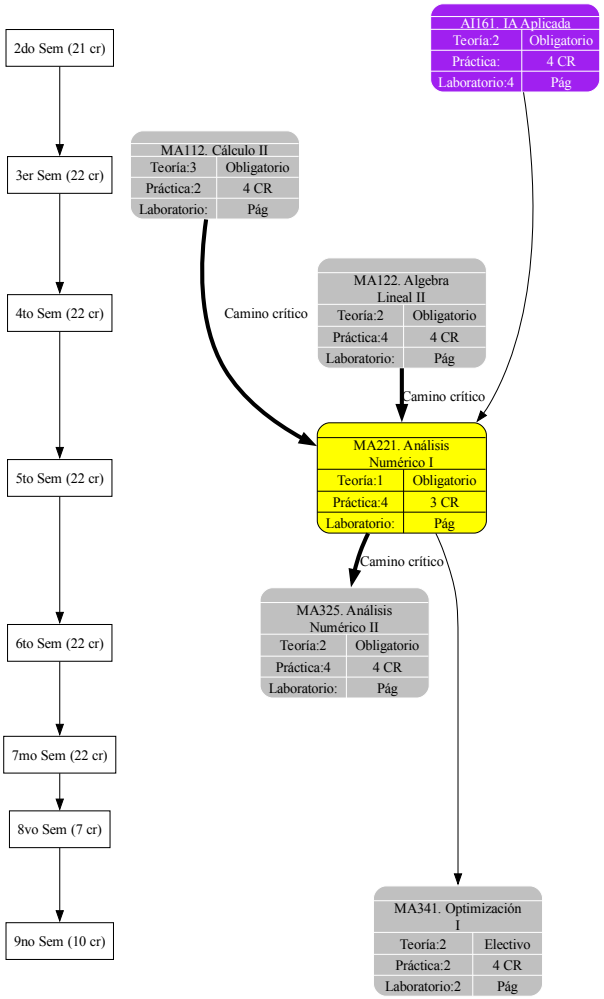


Figura 7.27: Mapa de Conexión. MA221 Análisis Numérico I

2. Fundamentación

Este curso introduce los métodos numéricos fundamentales para resolver problemas matemáticos mediante algoritmos computacionales. Desarrolla habilidades para implementar y analizar métodos numéricos en álgebra lineal, ecuaciones no lineales e interpolación, con énfasis en el análisis de error y la estabilidad numérica.

3. Objetivos del curso.

- Comprender los fundamentos de la aritmética computacional y análisis de error
- Implementar métodos numéricos para sistemas de ecuaciones lineales
- Resolver ecuaciones no lineales usando métodos iterativos
- Aplicar técnicas de interpolación y aproximación de funciones

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Aritmética Computacional y Análisis de Error, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aritmética de punto flotante y el estándar IEEE **II:** Fuentes de error: truncamiento, redondeo y discretización **III:** Propagación de error y número de condición **IV:** Estabilidad de algoritmos numéricos **V:** Error absoluto, relativo y directo/inverso

Objetivos de Aprendizaje **I:** Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse] **II:** Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse] **III:** Calcular el error absoluto y relativo de una aproximación [Usar] **IV:** Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar] **V:** Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar] **VI:** Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.2 álgebra Lineal Numérica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Descomposición LU y sus aplicaciones **II:** Descomposición QR y problemas de mínimos cuadrados **III:** Descomposición en Valores Singulares (SVD) y sus aplicaciones **IV:** Número de condición y estabilidad numérica de sistemas lineales **V:** Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)

Objetivos de Aprendizaje **I:** Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar] **II:** Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse] **III:** Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar] **IV:** Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar] **V:** Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.3 Búsqueda de Raíces y Optimización Numérica Básica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje I: Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.4 Interpolación y Aproximación de Funciones, 20 hr(s)

Competencias: ABET-1

Tópicos: I: Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje I: Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

6. Bibliografía

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Análisis Numérico*. Cengage Learning, 10th edition.

[Chapra and Canale, 2015] Chapra, S. C. and Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill, 7th edition.

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Análisis Numérico*. Cengage Learning, 10th edition.

[Chapra and Canale, 2015] Chapra, S. C. and Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill, 7th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA222 Análisis Real I (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA112 Cálculo II (3er Sem-Pág 350)
  - ID101 Inglés I (3er Sem-Pág 355)

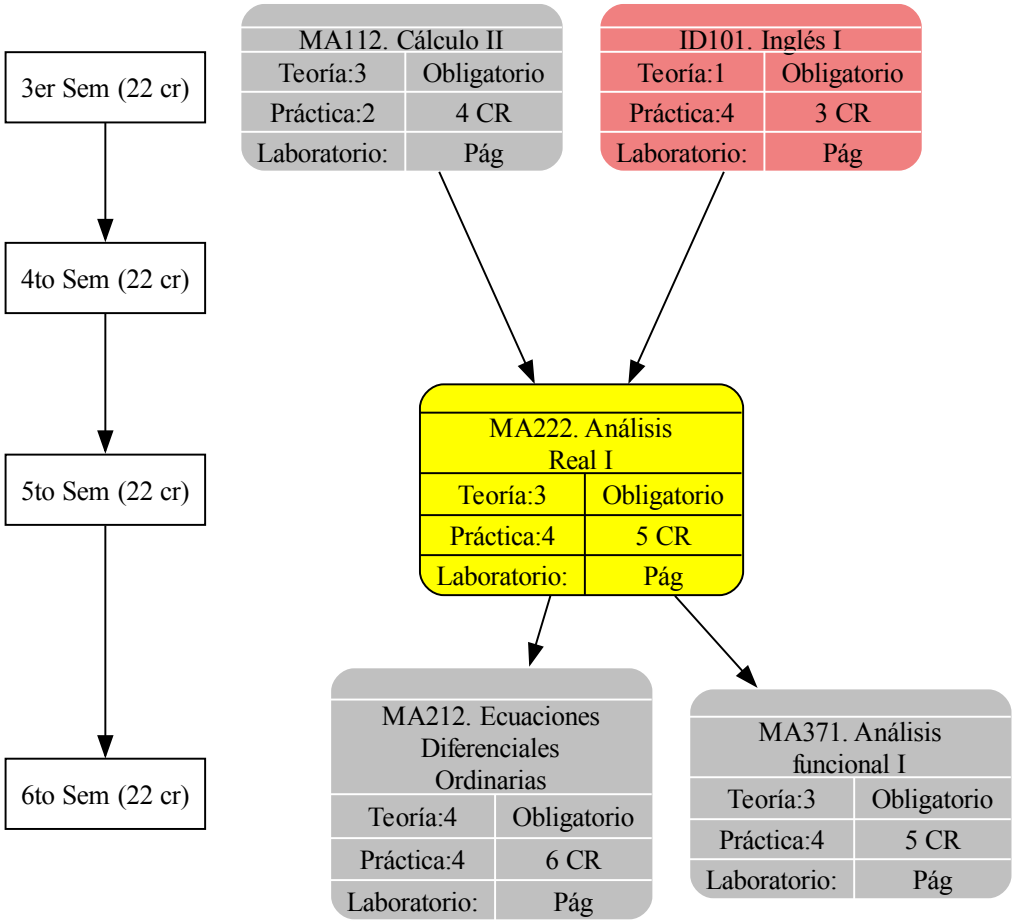


Figura 7.28: Mapa de Conexión. MA222 Análisis Real I

2. Fundamentación

La asignatura corresponde al área de la Estudios Específicos, la naturaleza del curso es teórica-práctica, su propósito está orientado a proporcionar una sólida formación matemática para la comprensión de todo fenómeno natural. Comprende las siguientes unidades temáticas: Numerabilidad. Sucesiones de números reales. Topología de R. Límite y continuidad de funciones reales de una variable real. Derivación. Integral de Riemann-Stieltjes.

3. Objetivos del curso.

- Comprender la estructura de los números reales y sus propiedades
- Analizar convergencia de sucesiones y series numéricas
- Dominar los conceptos de límites y continuidad de funciones
- Desarrollar habilidades en demostración matemática rigurosa

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Aritmética Computacional y Análisis de Error, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aritmética de punto flotante y el estándar IEEE **II:** Fuentes de error: truncamiento, redondeo y discretización **III:** Propagación de error y número de condición **IV:** Estabilidad de algoritmos numéricos **V:** Error absoluto, relativo y directo/inverso

Objetivos de Aprendizaje **I:** Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse] **II:** Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse] **III:** Calcular el error absoluto y relativo de una aproximación [Usar] **IV:** Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar] **V:** Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar] **VI:** Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar]

Bibliografía: [Rudin, 1976, Bartle and Sherbert, 2011]

5.2 Búsqueda de Raíces y Optimización Numérica Básica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Método de bisección e iteración de punto fijo **II:** Método de Newton y método de la secante para búsqueda de raíces **III:** Análisis de convergencia y orden de convergencia **IV:** Descenso de gradiente para optimización univariada **V:** Tasa de convergencia y criterios de parada

Objetivos de Aprendizaje **I:** Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Rudin, 1976, Bartle and Sherbert, 2011]

5.3 Interpolación y Aproximación de Funciones, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje **I:** Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Rudin, 1976, Bartle and Sherbert, 2011]

5.4 Diferenciación e Integración Numérica, 20 hr(s)

Competencias: ABET-1

Tópicos: **I:** Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) **II:** Extrapolación de Richardson **III:** Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) **IV:** Cuadratura gaussiana **V:** Análisis de error para integración numérica

Objetivos de Aprendizaje **I:** Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Rudin, 1976, Bartle and Sherbert, 2011]

6. Bibliografía

[Bartle and Sherbert, 2011] Bartle, R. G. and Sherbert, D. R. (2011). *Introduction to Real Analysis*. Wiley, 4th edition.

[Rudin, 1976] Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition.

[Bartle and Sherbert, 2011] Bartle, R. G. and Sherbert, D. R. (2011). *Introduction to Real Analysis*. Wiley, 4th edition.

[Rudin, 1976] Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA251 Introducción a la Estadística (Obligatorio)

1. Información general

- **Semestre**: 5^{to} Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - AI161 IA Aplicada (2^{do} Sem-Pág 326)

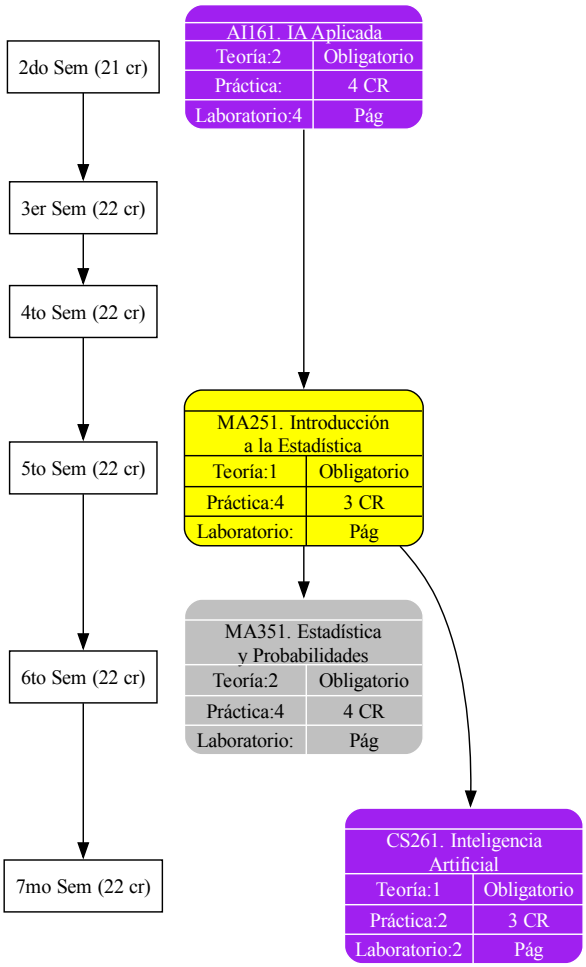


Figura 7.29: Mapa de Conexión. MA251 Introducción a la Estadística

2. Fundamentación

Escribir la justificación del curso aquí ...

3. Objetivos del curso.

- Escribir el 1er objetivo del curso aquí.
- Escribir el 2do objetivo del curso aquí.

4. Competencias Generales (*Outcomes*)

5. Contenido

5.1 Titulo de la unidad, 2 hr(s)

Competencias:

Tópicos: I: Topic1 **II:** Topic2

Objetivos de Aprendizaje I: LearningOutcome1 [Familiarizarse]. **II:** LearningOutcome2 [Usar]. **III:** LearningOutcome3 [Evaluar].

Bibliografía: [Force, 2020, ACM/IEEE-CS/AAAI Joint Task Force, 2023]

6. Bibliografía

[ACM/IEEE-CS/AAAI Joint Task Force, 2023] ACM/IEEE-CS/AAAI Joint Task Force (2023). Cs2023: Acm/ieee-cs/aaai computer science curricula. Technical report, ACM Press and IEEE Computer Society Press and AAAI Press.

[Force, 2020] Force, A.-C. J. T. (2020). Computing curricula 2020. Technical report, ACM Press and IEEE Computer Society Press.

[ACM/IEEE-CS/AAAI Joint Task Force, 2023] ACM/IEEE-CS/AAAI Joint Task Force (2023). Cs2023: Acm/ieee-cs/aaai computer science curricula. Technical report, ACM Press and IEEE Computer Society Press and AAAI Press.

[Force, 2020] Force, A.-C. J. T. (2020). Computing curricula 2020. Technical report, ACM Press and IEEE Computer Society Press.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA281 Teoría de números (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - MA100 Matemática Básica (1er Sem-Pág 292)

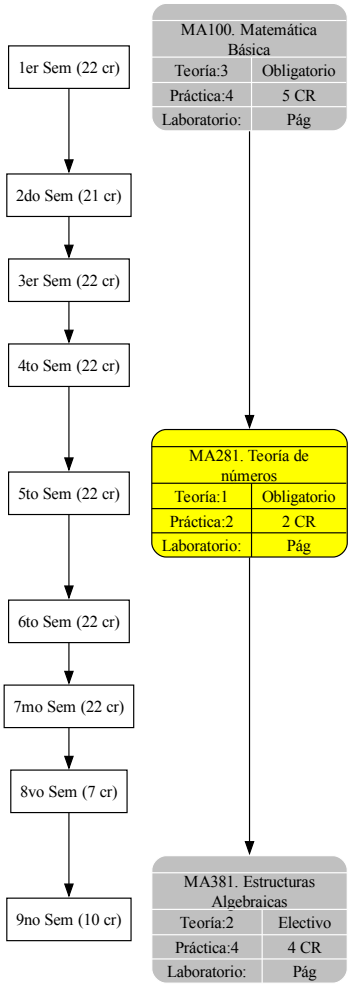


Figura 7.30: Mapa de Conexión. MA281 Teoría de números

2. Fundamentación

La Teoría de Números es una rama fundamental de las matemáticas puras que estudia las propiedades de los números enteros, siendo la base de múltiples aplicaciones en matemática computacional, criptografía y teoría de la información. Este curso proporciona al estudiante las herramientas teóricas y algorítmicas para analizar problemas de divisibilidad, primalidad, congruencias y ecuaciones diofánticas, desarrollando un pensamiento lógico-abstracto riguroso. El enfoque computacional permite implementar y validar los conceptos teóricos, sentando las bases para áreas como la criptografía de clave pública y los algoritmos de factorización.

3. Objetivos del curso.

- Comprender y aplicar los principios fundamentales de la divisibilidad, el algoritmo de Euclides y el teorema fundamental de la aritmética.
- Analizar y resolver problemas de congruencias lineales y sistemas de congruencias utilizando el Teorema Chino del Resto.
- Estudiar las funciones aritméticas básicas y sus aplicaciones en la distribución de números primos.
- Introducir los conceptos de residuos cuadráticos y la ley de reciprocidad cuadrática, comprendiendo su importancia teórica.
- Implementar algoritmos numéricos básicos (Euclides, primalidad, exponenciación modular) para validar los conceptos teóricos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Divisibilidad y Congruencias, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Axiomas de los números enteros: Principio de Buena Ordenación e Inducción Matemática. **II:** Divisibilidad: Algoritmo de la División, Máximo Común Divisor (Algoritmo de Euclides extendido) y lema de Bézout. **III:** Números primos y compuestos. Teorema Fundamental de la Aritmética. Infinidad de primos. **IV:** Congruencias: definición, propiedades algebraicas y clases residuales módulo n. **V:** Aplicaciones computacionales: implementación del algoritmo de Euclides y pruebas de primalidad básicas.

Objetivos de Aprendizaje **I:** Comprender los axiomas fundamentales de los números enteros y la técnica de demostración por inducción [Familiarizarse]. **II:** Utilizar el algoritmo de Euclides extendido para calcular el máximo común divisor y la identidad de Bézout [Usar]. **III:** Demostrar propiedades básicas de números primos y congruencias, aplicándolas en problemas elementales [Evaluar].

Bibliografía: [Rosen, 2011, Apostol, 1976]

5.2 Ecuaciones Diofánticas y Teorema Chino del Resto, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Ecuaciones diofánticas lineales: condiciones de solubilidad y método de solución general. **II:** Teorema Chino del Resto para sistemas de congruencias lineales con módulos coprimos. **III:** Aritmética modular: anillo $\mathbb{Z}_n$, unidades y divisores de cero. **IV:** Teoremas de Fermat y Euler. Función ϕ de Euler y sus propiedades. **V:** Aplicaciones en criptografía básica: cifrado César y fundamentos de RSA.

Objetivos de Aprendizaje **I:** Identificar las condiciones de existencia de soluciones para ecuaciones diofánticas lineales [Familiarizarse]. **II:** Aplicar el Teorema Chino del Resto para resolver sistemas de congruencias y problemas de calendario [Usar]. **III:** Utilizar los teoremas de Fermat y Euler para simplificar cálculos de potencias modulares y analizar estructuras algebraicas básicas [Evaluar].

Bibliografía: [Rosen, 2011, Burton, 2011]

5.3 Funciones Aritméticas y Teoría de Números Multiplicativa, 16 hr(s)

Competencias: ABET-1

Tópicos: I: Funciones aritméticas: Definición y ejemplos (tau, sigma, phi, mu). **II:** Convolución de Dirichlet y la fórmula de inversión de Möbius. **III:** Números perfectos, amigos y abundantes. Conjeturas abiertas. **IV:** Ordenes y raíces primitivas módulo un primo. **V:** Logaritmos discretos: definición y aplicaciones en protocolos de intercambio de clave (Diffie-Hellman).

Objetivos de Aprendizaje I: Reconocer las propiedades de las funciones aritméticas más importantes [Familiarizarse]. **II:** Aplicar la inversión de Möbius para resolver problemas de conteo en teoría de números [Usar]. **III:** Calcular órdenes y raíces primitivas, relacionándolas con la estructura cíclica de $\mathbb{Z}_p^*$ [Evaluar].

Bibliografía: [Apostol, 1976, Nathanson, 2000]

5.4 Residuos Cuadráticos y Aplicaciones Computacionales, 16 hr(s)

Competencias: ABET-1

Tópicos: I: Residuos cuadráticos y no cuadráticos. Símbolo de Legendre y sus propiedades. **II:** Ley de Reciprocidad Cuadrática. Demostración y aplicaciones en el cálculo del símbolo de Legendre. **III:** Símbolo de Jacobi y su uso en tests de primalidad probabilísticos. **IV:** Introducción a los tests de primalidad: Test de Miller-Rabin. **V:** Factorización de enteros: método $p-1$ de Pollard y nociones del algoritmo RSA.

Objetivos de Aprendizaje I: Determinar si un número es residuo cuadrático módulo un primo utilizando el símbolo de Legendre [Familiarizarse]. **II:** Aplicar la Ley de Reciprocidad Cuadrática para calcular símbolos de Legendre y Jacobi de manera eficiente [Usar]. **III:** Implementar algoritmos básicos de prueba de primalidad y factorización, analizando su complejidad y limitaciones [Evaluar].

Bibliografía: [Rosen, 2011, Hoffstein et al., 2008]

6. Bibliografía

[Apostol, 1976] Apostol, T. M. (1976). *Introduction to Analytic Number Theory*. Springer-Verlag, 1st edition. Texto clásico y riguroso para el estudio de funciones aritméticas y teoría analítica de números.

[Burton, 2011] Burton, D. M. (2011). *Elementary Number Theory*. McGraw-Hill Education, 7th edition. Libro introductorio muy didáctico, con abundantes ejemplos y ejercicios.

[Hoffstein et al., 2008] Hoffstein, J., Pipher, J., and Silverman, J. H. (2008). *An Introduction to Mathematical Cryptography*. Springer, 1st edition. Excelente puente entre la teoría de números y la criptografía aplicada.

[Nathanson, 2000] Nathanson, M. B. (2000). *Elementary Methods in Number Theory*, volume 195 of *Graduate Texts in Mathematics*. Springer. Enfoque en métodos combinatorios y aditivos, con demostraciones elegantes.

[Rosen, 2011] Rosen, K. H. (2011). *Elementary Number Theory and Its Applications*. Pearson, 6th edition. Texto ampliamente utilizado, con un balance excelente entre teoría, ejemplos y aplicaciones computacionales.

[Apostol, 1976] Apostol, T. M. (1976). *Introduction to Analytic Number Theory*. Springer-Verlag, 1st edition. Texto clásico y riguroso para el estudio de funciones aritméticas y teoría analítica de números.

[Burton, 2011] Burton, D. M. (2011). *Elementary Number Theory*. McGraw-Hill Education, 7th edition. Libro introductorio muy didáctico, con abundantes ejemplos y ejercicios.

[Hoffstein et al., 2008] Hoffstein, J., Pipher, J., and Silverman, J. H. (2008). *An Introduction to Mathematical Cryptography*. Springer, 1st edition. Excelente puente entre la teoría de números y la criptografía aplicada.

[Nathanson, 2000] Nathanson, M. B. (2000). *Elementary Methods in Number Theory*, volume 195 of *Graduate Texts in Mathematics*. Springer. Enfoque en métodos combinatorios y aditivos, con demostraciones elegantes.

[Rosen, 2011] Rosen, K. H. (2011). *Elementary Number Theory and Its Applications*. Pearson, 6th edition. Texto ampliamente utilizado, con un balance excelente entre teoría, ejemplos y aplicaciones computacionales.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

ID201 Inglés III (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - ID102 Inglés II (4to Sem-Pág 377)

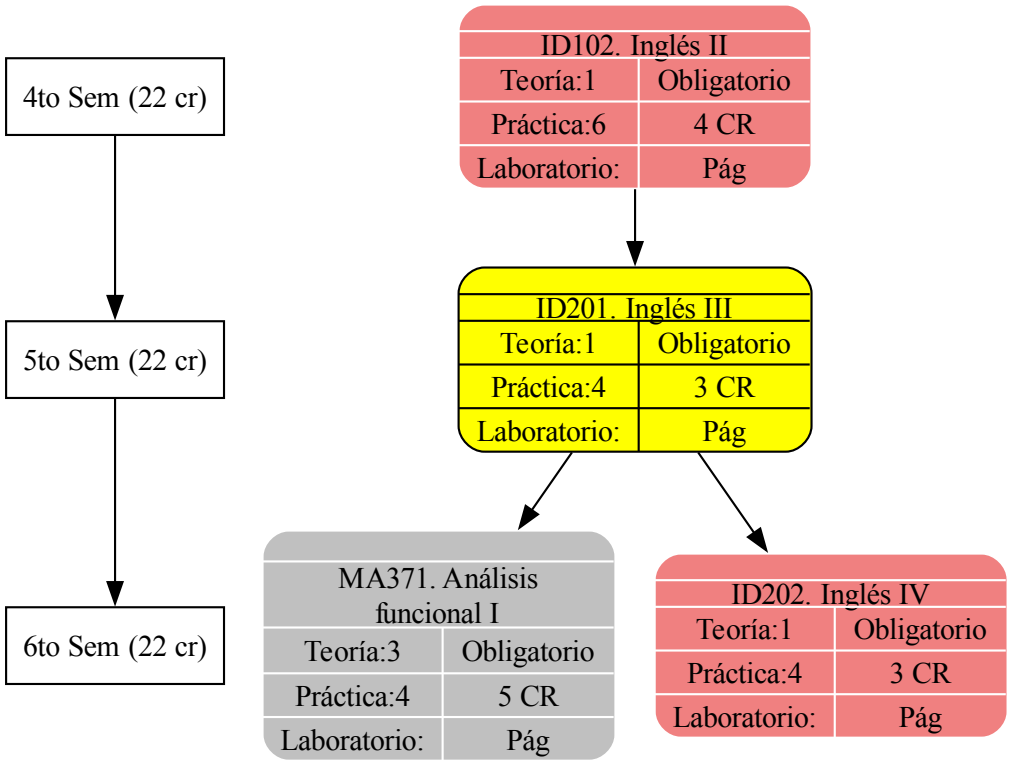


Figura 7.31: Mapa de Conexión. ID201 Inglés III

2. Fundamentación

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

3. Objetivos del curso.

- Formar en el alumno de capacidad de comprender y retener una conversación.
- Brindar técnicas de ilación de ideas.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 Getting to know you!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Tiempos Presente, Pasado y Futuro. **II:** Oraciones Interrogativas con Wh-. **III:** Palabras con más de un significado. **IV:** Partes de la oración. **V:** Expresiones para tiempo libre.

Objetivos de Aprendizaje I: Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los tiempos presente, pasado y futuro es capaz de expresar una mayor cantidad de acciones en forma de oraciones. Además es capaz de expresar ideas en forma de preguntas. Asume la idea de palabras con más de un significado. Utiliza expresiones sociales en situaciones de entretenimiento.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.2 The way we live!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Tiempo Presente Simple. **II:** Tiempo Presente Continuo. **III:** Colocaciones. **IV:** Vocabulario de países del mundo. **V:** Expresiones de enojo. **VI:** Conectores.

Objetivos de Aprendizaje I: Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas del mismo y las aplican adecuadamente. Describen países acuciosamente. Asumen expresiones para demostrar interés. Utilizan conectores para unir ideas varias.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.3 It all went wrong!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Tiempo Pasado Simple. **II:** Tiempo Pasado Continuo. **III:** Verbos Irregulares. **IV:** Expresiones de Tiempo. **V:** Conectores de tiempo.

Objetivos de Aprendizaje I: Al terminar la tercera unidad, los alumnos habiendo reconocido las características de los tiempos en pasado los utilizan adecuadamente. Utilizan prefijos y sufijos para crear y reconocer nuevas palabras. Describen tiempo en forma amplia. Utilizarán conjunciones para unir ideas tipo.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.4 Lets go shopping!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Expresiones de Cantidad Indefinida **II:** Oraciones Afirmativas, Negativas y Preguntas **III:** Uso de Artículos **IV:** Precios de productos **V:** Llenado de formatos y encuestas **VI:** Expresiones para ir de compras

Objetivos de Aprendizaje I: Al terminar la cuarta primera unidad, los alumnos habiendo identificado la idea de cantidad expresan diversas situaciones que la involucran. Reconocen y aplican artículos a sustantivos. Asumen la idea de ir de compras con la ayuda de expresiones. Expresan precios e ideas de dinero. Llenan formatos varios. Expresan actitudes.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.5 What do you want to do?, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Patrones Verbales I. **II:** Intenciones Futuras. **III:** Verbos de Percepción. **IV:** Vocabulario de sentimientos. **V:** Expresiones de Planes y Ambiciones.

Objetivos de Aprendizaje I: Al finalizar la quinta unidad, los alumnos, a partir de la comprensión de la idea de patrones verbales elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar intenciones futuras. Adquirirán vocabulario para describir sentimientos. Se presentará expresiones para describir planes y ambiciones.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.6 The best in the world!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Whats it like?. **II:** Adjetivos. **III:** Comparativos y Superlativos. **IV:** Sinónimos y Antónimos. **V:** Indicaciones de Dirección. **VI:** Lecturas.

Objetivos de Aprendizaje I: Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de adjetivos, estructuran oraciones con diversas formas de los mismos en contextos adecuados. Enfatizan la diferencia entre tipos de ciudades y pueblos y estilos de vida. Utilizan expresiones indicación de direcciones.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.7 Fame!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Presente Perfecto y Pasado Simple **II:** Expresiones for, ever, since **III:** Adverbios **IV:** Expresiones que vienen en pares **V:** Respuestas cortas **VI:** Celebridades

Objetivos de Aprendizaje I: Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo presente perfecto y lo diferencian del pasado simple. Enfatizan la diferencia entre formas de adjetivos. Describen ideas de la música. Utilizan expresiones para dar respuestas cortas. Asumen la idea de dar explicaciones extra de los elementos de una oración.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

6. Bibliografía

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002d] Soars, L. and John (2002d). *American Headway N 2 Student Book*. Editorial Oxford.

Sexto Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA212 Ecuaciones Diferenciales Ordinarias (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 6
- **Horas del curso**: **Teoría**: 4 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 321)
  - MA121 Algebra Lineal I (3er Sem-Pág 352)
  - MA222 Análisis Real I (5to Sem-Pág 390)

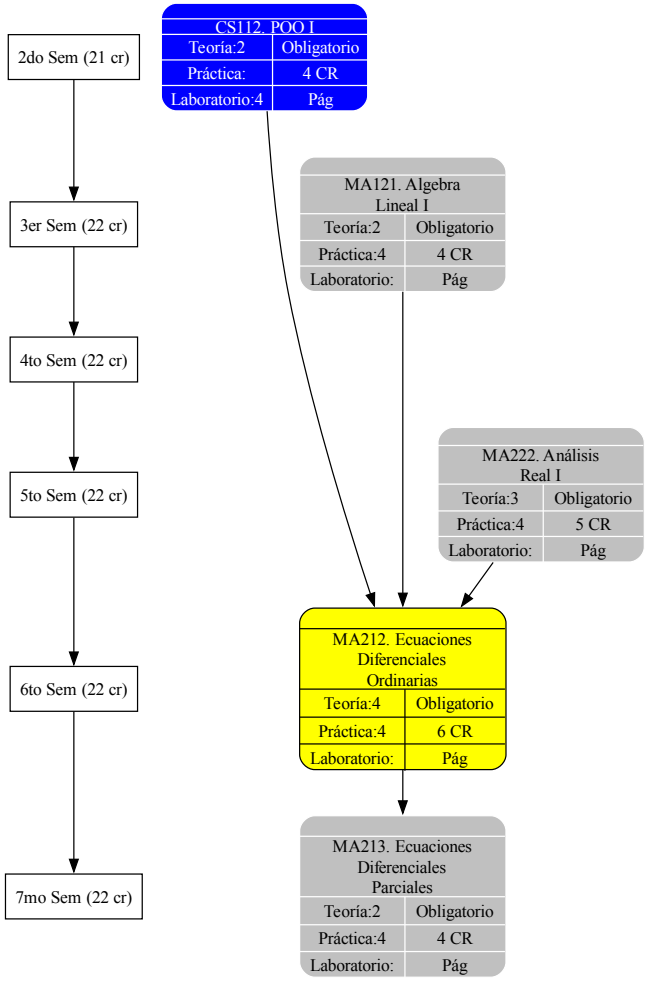


Figura 7.32: Mapa de Conexión. MA212 Ecuaciones Diferenciales Ordinarias

2. Fundamentación

El curso de Ecuaciones Diferenciales Ordinarias proporciona las herramientas matemáticas para analizar sistemas que evolucionan respecto a una sola variable independiente. Se enfoca en la mo-

delación de sistemas dinámicos mediante ecuaciones de primer orden, orden superior y sistemas de ecuaciones, utilizando métodos analíticos y numéricos. Es fundamental para la comprensión de la mecánica, circuitos eléctricos y modelos poblacionales.

3. Objetivos del curso.

- Resolver ecuaciones diferenciales ordinarias de primer orden y orden superior mediante métodos analíticos.
- Aplicar la transformada de Laplace para resolver problemas de valor inicial con funciones discontinuas.
- Analizar la estabilidad de sistemas de ecuaciones diferenciales lineales y no lineales.
- Implementar métodos numéricos para la aproximación de soluciones en modelos complejos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Ecuaciones Diferenciales Ordinarias de Primer Orden, 20 hr(s)

Competencias: ABET-1

Tópicos: I: Modelado con ecuaciones de primer orden: crecimiento, mezclas y enfriamiento

Objetivos de Aprendizaje I: Clasificar EDOs de primer orden y seleccionar el método de solución apropiado (separación de variables, ecuación lineal, ecuación exacta o factor integrante) [Familiarizarse] **II:** Formular y resolver modelos de primer orden para crecimiento y decaimiento, problemas de mezcla, la ley de enfriamiento de Newton y la ley de Torricelli [Usar] **III:** Analizar cualitativamente EDOs autónomas de primer orden mediante campos de direcciones, e identificar puntos de equilibrio y su estabilidad [Evaluar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

5.2 Ecuaciones Diferenciales de Orden Superior y Sistemas, 40 hr(s)

Competencias: ABET-1

Tópicos: I: Definición y propiedades de la Transformada de Laplace; transformadas inversas y teoremas de traslación **II:** Derivadas de una transformada e integrales de convolución

Objetivos de Aprendizaje I: Enunciar el teorema de existencia y unicidad de Picard-Lindelöf e interpretar sus implicancias para la continuación de soluciones de una EDO de orden superior [Familiarizarse] **II:** Resolver sistemas lineales de EDOs usando métodos de valores propios y esbozar sus retratos de fase [Usar] **III:** Aplicar la transformada de Laplace para resolver EDOs con funciones de forzamiento a trozos e impulsivas [Evaluar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

5.3 Métodos Numéricos para Ecuaciones Diferenciales, 36 hr(s)

Competencias: ABET-1

Tópicos: I: Retratos de fase de sistemas 2D: equilibrios, estabilidad y ciclos límite **II:** Estabilidad de puntos críticos y sistemas no lineales

Objetivos de Aprendizaje I: Comparar solucionadores de EDOs explícitos e implícitos y explicar su estabilidad relativa para problemas rígidos [Familiarizarse] **II:** Implementar un método de Runge-Kutta con control de paso adaptativo y aplicarlo a un sistema de EDOs [Usar] **III:** Analizar la estabilidad y convergencia de un esquema de diferencias finitas para una EDP parabólica usando el teorema de equivalencia de Lax [Evaluar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

6. Bibliografía

[Boyce et al., 2017] Boyce, W. E., DiPrima, R. C., and Meade, D. B. (2017). *Ecuaciones Diferenciales y Problemas con Valores en la Frontera*. Limusa Wiley, 11th edition.

[Zill, 2018] Zill, D. G. (2018). *Ecuaciones Diferenciales con Aplicaciones de Modelado*. Cengage Learning, 11th edition.

[Boyce et al., 2017] Boyce, W. E., DiPrima, R. C., and Meade, D. B. (2017). *Ecuaciones Diferenciales y Problemas con Valores en la Frontera*. Limusa Wiley, 11th edition.

[Zill, 2018] Zill, D. G. (2018). *Ecuaciones Diferenciales con Aplicaciones de Modelado*. Cengage Learning, 11th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA325 Análisis Numérico II (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA221 Análisis Numérico I (5to Sem-Pág 387)

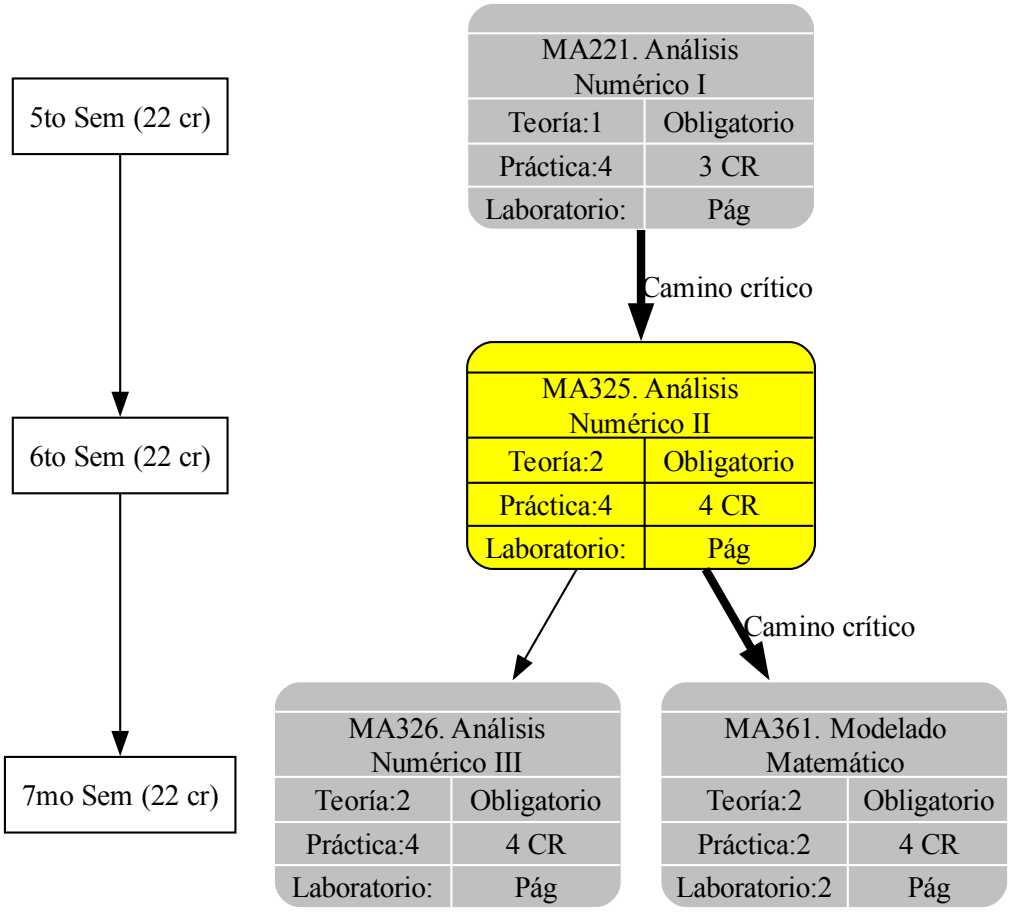


Figura 7.33: Mapa de Conexión. MA325 Análisis Numérico II

2. Fundamentación

Este curso avanza en el estudio de métodos numéricos para problemas matemáticos avanzados, enfocándose en ecuaciones diferenciales, integración numérica y problemas de valores propios. Desarrolla habilidades para implementar y analizar algoritmos numéricos sofisticados con aplicaciones en ingeniería y ciencias computacionales, con énfasis en estabilidad y eficiencia computacional.

3. Objetivos del curso.

- Resolver numéricamente ecuaciones diferenciales ordinarias y parciales

- Aplicar métodos avanzados de integración y diferenciación numérica
- Analizar problemas de valores propios y descomposiciones matriciales
- Evaluar la estabilidad y convergencia de métodos numéricos avanzados

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs), 20 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Método de Euler y sus variantes (explícito, implícito) **II:** Métodos de Runge-Kutta **III:** Métodos multipaso (Adams-Bashforth, Adams-Moulton) **IV:** Análisis de estabilidad y convergencia para solucionadores de EDOs **V:** Sistemas de EDOs y EDOs de orden superior

Objetivos de Aprendizaje **I:** Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar] **II:** Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar] **III:** Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar] **IV:** Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.2 Diferenciación e Integración Numérica, 20 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) **II:** Extrapolación de Richardson **III:** Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) **IV:** Cuadratura gaussiana **V:** Análisis de error para integración numérica

Objetivos de Aprendizaje **I:** Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.3 álgebra Lineal Numérica, 20 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Descomposición LU y sus aplicaciones **II:** Descomposición QR y problemas de mínimos cuadrados **III:** Descomposición en Valores Singulares (SVD) y sus aplicaciones **IV:** Número de condición y estabilidad numérica de sistemas lineales **V:** Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)

Objetivos de Aprendizaje **I:** Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar] **II:** Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse] **III:** Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar] **IV:** Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar] **V:** Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

5.4 Interpolación y Aproximación de Funciones, 20 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Interpolación polinomial (formas de Lagrange, Newton) **II:** Interpolación por splines (splines cúbicos) **III:** Aproximación por mínimos cuadrados **IV:** Fenómeno de Runge y nodos de Chebyshev **V:** Análisis de error para interpolación

Objetivos de Aprendizaje **I:** Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Burden et al., 2016, Chapra and Canale, 2015]

6. Bibliografía

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Análisis Numérico*. Cengage Learning, 10th edition.

[Chapra and Canale, 2015] Chapra, S. C. and Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill, 7th edition.

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Análisis Numérico*. Cengage Learning, 10th edition.

[Chapra and Canale, 2015] Chapra, S. C. and Canale, R. P. (2015). *Métodos Numéricos para Ingenieros*. McGraw-Hill, 7th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA351 Estadística y Probabilidades (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA251 Introducción a la Estadística (5to Sem-Pág 393)
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 321)

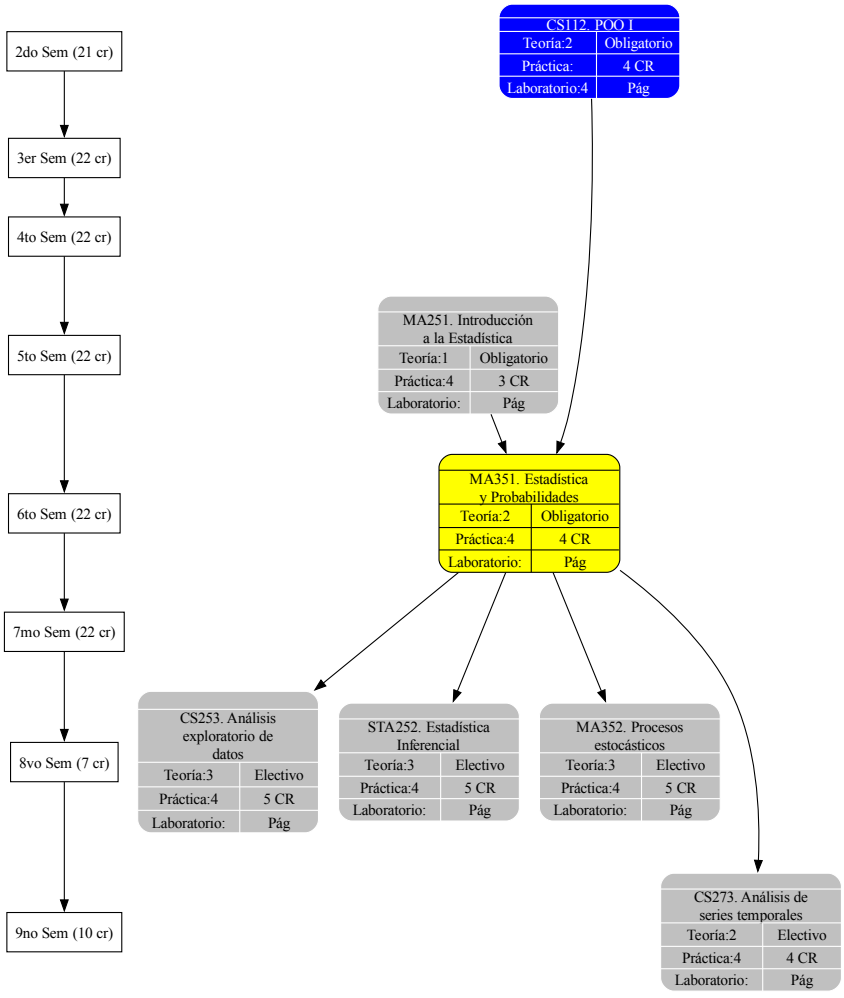


Figura 7.34: Mapa de Conexión. MA351 Estadística y Probabilidades

2. Fundamentación

La probabilidad y la estadística proporcionan la base matemática para razonar bajo incertidumbre, analizar datos y construir modelos predictivos — habilidades esenciales tanto en computación, inteligencia artificial, matemática computacional como en ingeniería. Este curso desarrolla una comprensión rigurosa de la teoría de la probabilidad y su uso para describir fenómenos aleatorios, junto con los fundamentos de la estadística descriptiva, la inferencia estadística (estimación, pruebas de hipótesis y

razonamiento bayesiano) y el modelado de regresión, de modo que los estudiantes puedan resumir y analizar datos, cuantificar la incertidumbre y extraer conclusiones cuantitativas sólidas en su propio campo de aplicación.

3. Objetivos del curso.

- Comprender los fundamentos de la teoría de la probabilidad y la estadística descriptiva.
- Aplicar distribuciones de probabilidad y métodos de inferencia estadística para analizar datos y cuantificar la incertidumbre.
- Desarrollar habilidades para modelar fenómenos aleatorios y construir modelos estadísticos predictivos como la regresión lineal.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

5. Contenido

5.1 Resúmenes Numéricos: Posición, Dispersión y Forma, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Medidas de posición: media, mediana, moda, media recortada y media ponderada **II:** Medidas de dispersión: varianza, desviación estándar, RIQ, MAD y rango **III:** Medidas de forma: asimetría y curtosis; cuantiles y percentiles **IV:** Estadísticas de resumen robustas: resistencia a valores atípicos y distribuciones de colas pesadas

Objetivos de Aprendizaje **I:** Calcular e interpretar medidas de posición, dispersión y forma para un conjunto de datos univariado [Usar] **II:** Seleccionar estadísticas de resumen apropiadas según las propiedades distribucionales de los datos [Usar] **III:** Comparar estadísticas de resumen clásicas y robustas y justificar la elección para un conjunto de datos dado [Evaluar]

Bibliografía: [Devore, 2016]

5.2 Espacios Muestrales y Eventos, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Experimentos aleatorios, resultados y espacios muestrales: finitos, contables e incontables **II:** Operaciones de teoría de conjuntos sobre eventos: unión, intersección, complemento y leyes de De Morgan **III:** Sigma-álgebras y espacios medibles como fundamento de la probabilidad rigurosa **IV:** Conjuntos de Borel y la sigma-álgebra de Borel sobre la recta real

Objetivos de Aprendizaje **I:** Definir espacios muestrales, eventos y sigma-álgebras para un experimento aleatorio dado [Familiarizarse] **II:** Aplicar operaciones de teoría de conjuntos para calcular probabilidades de eventos compuestos [Usar] **III:** Construir sigma-álgebras apropiadas para espacios medibles que surgen en problemas estadísticos [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.3 Axiomas y Propiedades de la Probabilidad, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Los tres axiomas de probabilidad de Kolmogorov y sus consecuencias **II:** Propiedades derivadas: regla del complemento, regla de la adición, monotonicidad y continuidad de la probabilidad **III:** Interpretaciones clásica, frecuentista y subjetiva (bayesiana) de la probabilidad **IV:** Métodos de conteo: permutaciones, combinaciones y el principio de multiplicación

Objetivos de Aprendizaje I: Enunciar los axiomas de Kolmogorov y derivar propiedades fundamentales de la probabilidad [Familiarizarse] **II:** Calcular probabilidades de eventos utilizando la regla de la adición y la regla del complemento [Usar] **III:** Aplicar métodos de conteo para calcular probabilidades en espacios muestrales finitos [Usar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.4 Probabilidad Condicional e Independencia, 4 hr(s)

Competencias: ABET-1

Tópicos: I: Definición e interpretación de la probabilidad condicional P(A|B) y sus propiedades **II:** Teorema de Bayes y ley de la probabilidad total: derivación y aplicaciones **III:** Independencia de eventos: independencia por pares e independencia mutua **IV:** Independencia condicional y su papel en los modelos gráficos probabilísticos

Objetivos de Aprendizaje I: Explicar la probabilidad condicional y la independencia y su relación [Familiarizarse] **II:** Aplicar el teorema de Bayes y la ley de la probabilidad total para resolver problemas de probabilidad [Usar] **III:** Analizar si los eventos son independientes o condicionalmente independientes en un modelo de probabilidad dado [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.5 Variables Aleatorias Discretas y Distribuciones, 4 hr(s)

Competencias: ABET-1

Tópicos: I: Funciones de masa de probabilidad (FMP) y funciones de distribución acumulada (FDA) para variables aleatorias discretas **II:** Distribuciones discretas clave: Bernoulli, Binomial, Geométrica, Binomial Negativa, Poisson, Hipergeométrica **III:** Propiedades de las distribuciones discretas: media, varianza, asimetría y curtosis **IV:** Aproximación de Poisson a la Binomial y argumentos de distribución límite

Objetivos de Aprendizaje I: Identificar y describir las principales distribuciones de probabilidad discretas y sus parámetros [Familiarizarse] **II:** Calcular probabilidades, medias y varianzas utilizando las FMP de distribuciones discretas estándar [Usar] **III:** Seleccionar una distribución discreta apropiada para modelar un fenómeno aleatorio dado y justificar la elección [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.6 Variables Aleatorias Continuas y Distribuciones, 4 hr(s)

Competencias: ABET-1

Tópicos: I: Funciones de densidad de probabilidad (FDP) y FDA para variables aleatorias continuas **II:** Distribuciones continuas clave: Uniforme, Normal, Exponencial, Gamma, Beta, Chi-cuadrado, t, F **III:** La distribución Normal: propiedades, normal estándar y su papel central en la inferencia estadística **IV:** La familia exponencial de distribuciones: forma canónica y estadísticos suficientes

Objetivos de Aprendizaje I: Describir las propiedades y parámetros de las principales distribuciones de probabilidad continuas [Familiarizarse] **II:** Calcular probabilidades y cuantiles para distribuciones continuas estándar [Usar] **III:** Identificar distribuciones pertenecientes a la familia exponencial y derivar sus estadísticos suficientes [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.7 Distribuciones Conjuntas, Marginales y Condicionales, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** FMP y FDP conjuntas para dos o más variables aleatorias **II:** Distribuciones marginales y condicionales derivadas de distribuciones conjuntas **III:** Covarianza, correlación y sus propiedades; la independencia implica covarianza cero (el recíproco no siempre es cierto) **IV:** La distribución normal bivariada y sus distribuciones condicionales

Objetivos de Aprendizaje **I:** Derivar distribuciones marginales y condicionales a partir de una distribución conjunta dada [Usar] **II:** Calcular la covarianza y correlación entre dos variables aleatorias [Usar] **III:** Analizar la estructura condicional de la distribución normal bivariada [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.8 Teoremas Límite: LGN y TCL, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Ley débil de los grandes números: convergencia en probabilidad y sus implicaciones para la estimación **II:** Teorema central del límite: enunciado, condiciones y papel en la justificación de aproximaciones normales **III:** Aplicaciones del TCL: aproximación normal a la binomial y distribuciones muestrales **IV:** Ley fuerte de los grandes números: convergencia casi segura **V:** El método delta: distribución asintótica de funciones suaves de estimadores

Objetivos de Aprendizaje **I:** Enunciar la LGN y el TCL y explicar las condiciones bajo las cuales se cumplen [Familiarizarse] **II:** Aplicar el TCL para derivar aproximaciones normales y distribuciones muestrales [Usar] **III:** Aplicar el método delta para derivar distribuciones asintóticas de estimadores no lineales [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.9 Estimación Puntual, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Propiedades deseables: insesgamiento, consistencia, eficiencia y suficiencia **II:** Estimación por método de momentos: derivación y propiedades asintóticas **III:** Estimación de máxima verosimilitud: derivación, propiedades y principio de invarianza **IV:** Cota inferior de Cramér-Rao y el concepto de información de Fisher

Objetivos de Aprendizaje **I:** Describir las propiedades clave de los estimadores y explicar las concesiones entre ellas [Familiarizarse] **II:** Derivar estimadores por método de momentos y de máxima verosimilitud para distribuciones estándar [Usar] **III:** Evaluar la eficiencia de un estimador usando la cota inferior de Cramér-Rao [Evaluar]

Bibliografía: [Devore, 2016]

5.10 Estimación por Intervalos e Intervalos de Confianza, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Construcción de intervalos de confianza mediante cantidades pivote y distribuciones exactas **II:** Interpretación correcta de los intervalos de confianza: probabilidad de cobertura y errores conceptuales comunes **III:** Intervalos de confianza para muestras grandes basados en normalidad asintótica **IV:** Dualidad entre intervalos de confianza y pruebas de hipótesis

Objetivos de Aprendizaje **I:** Construir intervalos de confianza para medias, proporciones y varianzas usando métodos exactos y de gran muestra [Usar] **II:** Interpretar correctamente los intervalos de confianza e identificar interpretaciones erróneas comunes [Usar] **III:** Explicar la dualidad entre intervalos de confianza y pruebas de hipótesis y utilizarla para extraer inferencias [Evaluar]

Bibliografía: [Devore, 2016]

5.11 Fundamentos de las Pruebas de Hipótesis, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Marco de Neyman-Pearson: hipótesis nula y alternativa, regiones de rechazo, errores tipo I y II **II:** Valores p: definición, interpretación correcta y usos incorrectos comunes; potencia y tamaño de muestra **III:** Pruebas paramétricas comunes: prueba z, prueba t, prueba F y prueba ji-cuadrada **IV:** Lema de Neyman-Pearson y pruebas uniformemente más poderosas

Objetivos de Aprendizaje **I:** Explicar la lógica de las pruebas de hipótesis, los errores tipo I y II y el significado de un valor p [Familiarizarse] **II:** Aplicar pruebas paramétricas estándar e interpretar los resultados en contexto [Usar] **III:** Derivar la prueba más poderosa para una hipótesis simple usando el lema de Neyman-Pearson [Evaluar]

Bibliografía: [Devore, 2016]

5.12 Inferencia Bayesiana, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Marco bayesiano: a priori, verosimilitud, a posteriori y el papel del teorema de Bayes **II:** Distribuciones a priori: informativas, no informativas y conjugadas **III:** Inferencia a posteriori: estimaciones puntuales, intervalos de credibilidad y distribuciones predictivas a posteriori **IV:** Comparación de modelos bayesianos: factores de Bayes y probabilidades a posteriori de modelos

Objetivos de Aprendizaje **I:** Explicar el enfoque bayesiano de la inferencia y contrastarlo con el enfoque frecuentista [Familiarizarse] **II:** Derivar la distribución a posteriori para modelos conjugados estándar [Usar] **III:** Aplicar factores de Bayes para comparar modelos estadísticos e interpretar la evidencia [Evaluar]

Bibliografía: [Ross, 2014, Devore, 2016]

5.13 Regresión Lineal Simple, 4 hr(s)

Competencias: ABET-1

Tópicos: **I:** Estimación por mínimos cuadrados ordinarios (MCO): derivación de coeficientes y sus propiedades **II:** Inferencia sobre coeficientes de regresión: pruebas t, intervalos de confianza y valores p **III:** Coeficiente de determinación R-cuadrado y descomposición de la varianza mediante análisis de varianza **IV:** Teorema de Gauss-Markov: propiedad MELI del MCO bajo supuestos clásicos

Objetivos de Aprendizaje **I:** Ajustar un modelo de regresión lineal simple e interpretar los coeficientes estimados [Usar] **II:** Realizar pruebas de hipótesis y construir intervalos de confianza para parámetros de regresión [Usar] **III:** Verificar los supuestos de Gauss-Markov y evaluar las consecuencias cuando se violan [Evaluar]

Bibliografía: [Devore, 2016]

6. Bibliografía

[Devore, 2016] Devore, J. L. (2016). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning.

[Ross, 2014] Ross, S. M. (2014). *A First Course in Probability*. Pearson.

[Devore, 2016] Devore, J. L. (2016). *Probability and Statistics for Engineering and the Sciences*. Cengage Learning.

[Ross, 2014] Ross, S. M. (2014). *A First Course in Probability*. Pearson.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA371 Análisis funcional I (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA222 Análisis Real I (5to Sem-Pág 390)
  - ID201 Inglés III (5to Sem-Pág 399)
  - MA122 Algebra Lineal II (4to Sem-Pág 371)

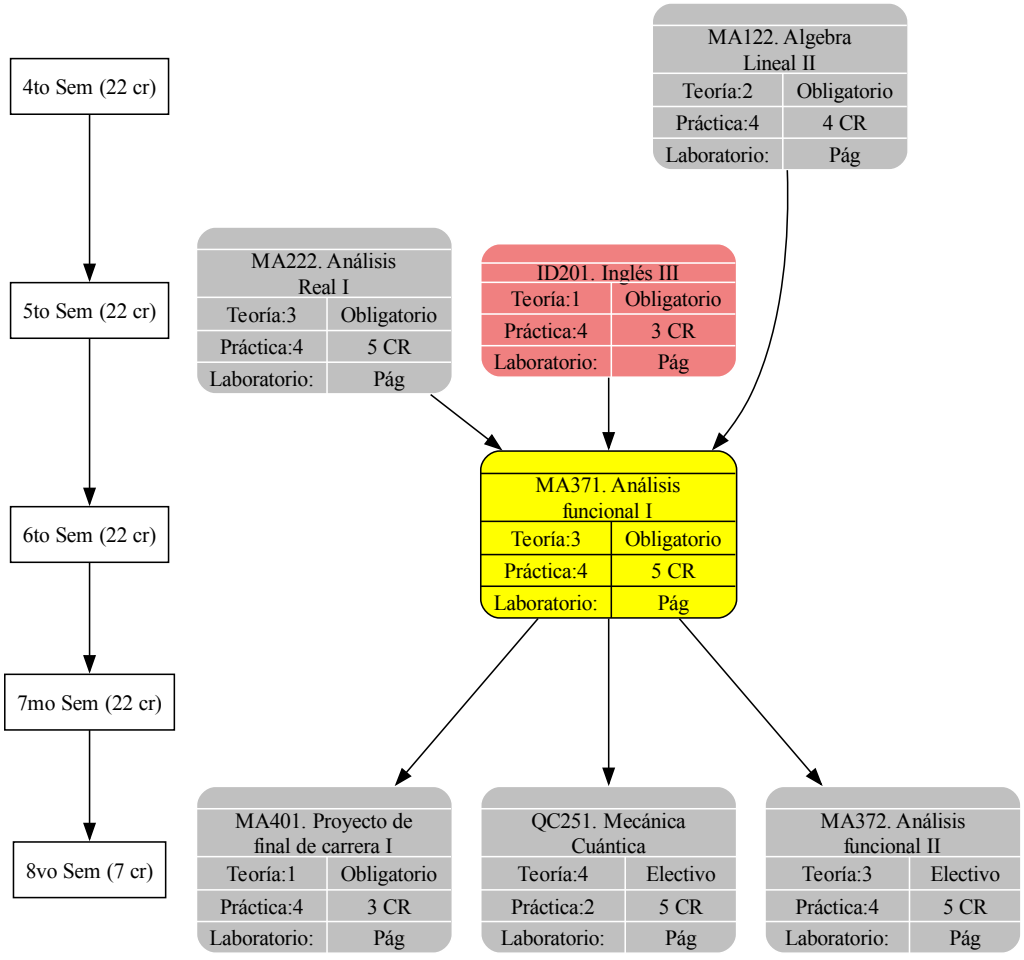


Figura 7.35: Mapa de Conexión. MA371 Análisis funcional I

2. Fundamentación

El análisis funcional es un pilar central de las matemáticas modernas, proporcionando el marco para resolver problemas en ecuaciones diferenciales, mecánica cuántica y análisis numérico. Este curso cierra la brecha entre el álgebra lineal y los espacios de dimensión infinita. Al introducir la Teoría de la Medida como una base rigurosa para la integración, el curso permite el estudio de los espacios

de Banach y Hilbert, los operadores lineales y los teoremas fundamentales que rigen los sistemas funcionales.

3. Objetivos del curso.

- Establecer una comprensión rigurosa de la medida de Lebesgue y la teoría de la integración.
- Dominar las estructuras topológicas y algebraicas de los espacios de Banach y Hilbert.
- Aplicar los teoremas fundamentales del análisis funcional (Hahn-Banach, Aplicación Abierta, Grafo Cerrado).
- Analizar las propiedades de los operadores lineales y su teoría espectral en dimensiones infinitas.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Medida e Integración, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** σ-álgebras y espacios medibles **II:** Medida de Lebesgue en $\mathbb{R}^n$ **III:** Funciones medibles e integración de Lebesgue **IV:** Teoremas de convergencia (Convergencia Monótona, Convergencia Dominada, Lema de Fatou) **V:** Espacios L^p: completitud y dualidad

Objetivos de Aprendizaje **I:** Construir la medida de Lebesgue e identificar conjuntos medibles [Assessment] **II:** Evaluar integrales utilizando los teoremas de convergencia [Usage] **III:** Probar la completitud de los espacios L^p [Familiarity]

Bibliografía: [Folland, 1999, Royden and Fitzpatrick, 2010]

5.2 Espacios Normados y de Banach, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Espacios vectoriales normados y completitud **II:** Bases de Schauder y dimensión infinita **III:** Operadores lineales y funcionales **IV:** Espacios duales y el teorema de Hahn-Banach **V:** Teoremas de categoría: Baire, Aplicación Abierta y Grafo Cerrado

Objetivos de Aprendizaje **I:** Verificar la completitud de diversos espacios funcionales [Assessment] **II:** Aplicar el teorema de Hahn-Banach para extender funcionales lineales [Usage] **III:** Utilizar el teorema de la aplicación abierta para determinar el carácter acotado de un operador [Assessment]

Bibliografía: [Kreyszig, 1989, Brezis, 2011]

5.3 Espacios de Hilbert, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Espacios con producto interno y estructura de Hilbert **II:** Ortogonalidad, teorema de proyección y representación de Riesz **III:** Conjuntos ortonormales y series de Fourier en espacios de Hilbert **IV:** Operadores autoadjuntos, unitarios y normales

Objetivos de Aprendizaje **I:** Construir proyecciones ortogonales en espacios de Hilbert [Usage] **II:** Representar funcionales lineales mediante el teorema de Riesz [Assessment] **III:** Analizar la geometría de espacios de producto interno de dimensión infinita [Familiarity]

Bibliografía: [Kreyszig, 1989, Conway, 1990]

5.4 Teoría de Operadores y Teoría Espectral, 16 hr(s)

Competencias: ABET-1

Tópicos: **I:** Operadores compactos en espacios de Banach y Hilbert **II:** La alternativa de Fredholm **III:** Teoría espectral de operadores lineales acotados **IV:** Teorema espectral para operadores compactos autoadjuntos

Objetivos de Aprendizaje **I:** Identificar operadores compactos y sus propiedades [Assessment] **II:** Determinar el espectro de operadores lineales [Usage] **III:** Aplicar el teorema espectral para descomponer operadores [Assessment]

Bibliografía: [Brezis, 2011, Yosida, 1980]

6. Bibliografía

[Brezis, 2011] Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer Science & Business Media.

[Conway, 1990] Conway, J. B. (1990). *A Course in Functional Analysis*. Springer-Verlag, 2nd edition.

[Folland, 1999] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. Wiley-Interscience, 2nd edition.

[Kreyszig, 1989] Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. Wiley.

[Royden and Fitzpatrick, 2010] Royden, H. and Fitzpatrick, P. (2010). *Real Analysis*. Pearson, 4th edition.

[Yosida, 1980] Yosida, K. (1980). *Functional Analysis*. Springer-Verlag, 6th edition.

[Brezis, 2011] Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer Science & Business Media.

[Conway, 1990] Conway, J. B. (1990). *A Course in Functional Analysis*. Springer-Verlag, 2nd edition.

[Folland, 1999] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. Wiley-Interscience, 2nd edition.

[Kreyszig, 1989] Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. Wiley.

[Royden and Fitzpatrick, 2010] Royden, H. and Fitzpatrick, P. (2010). *Real Analysis*. Pearson, 4th edition.

[Yosida, 1980] Yosida, K. (1980). *Functional Analysis*. Springer-Verlag, 6th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

ID202 Inglés IV (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - ID201 Inglés III (5to Sem-Pág 399)

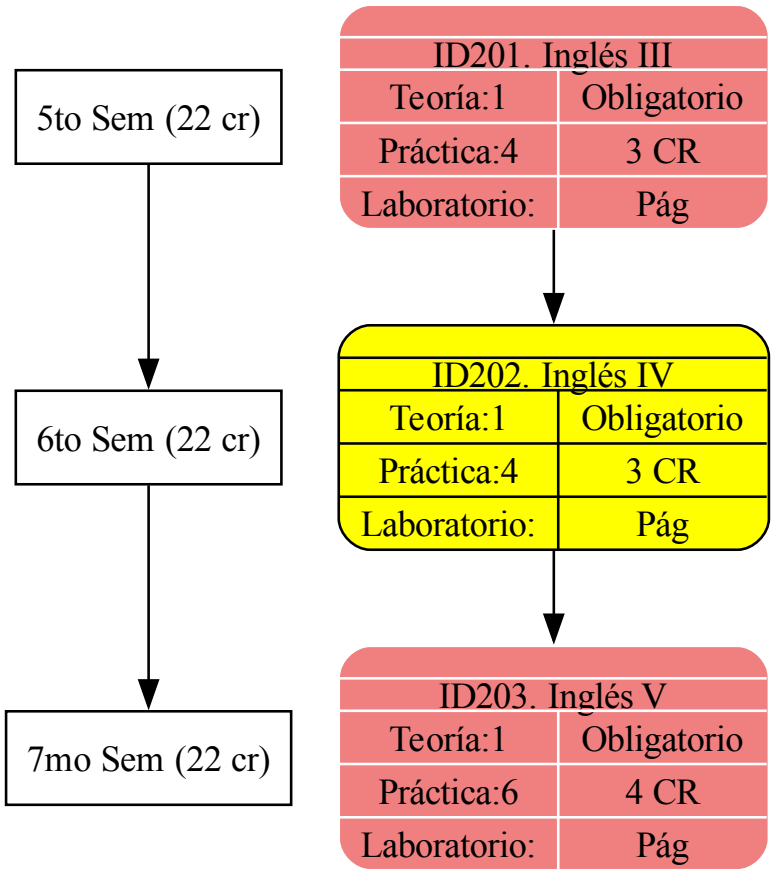


Figura 7.36: Mapa de Conexión. ID202 Inglés IV

2. Fundamentación

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

3. Objetivos del curso.

- Incrementar el nivel de conversación en diferentes temas, en los alumnos. Así como la capacidad de escribir y leer documentación de todo tipo.
- Llevar al alumno a una expresión más intensa en el dominio del idioma.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 Do and don't!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Auxiliares de Modo should, must y have got to. **II:** Oraciones afirmativas, negativas e interrogativas con modals. **III:** Términos para cartas formales. **IV:** Partes de las respuestas cortas. **V:** Expresiones para ocupaciones.

Objetivos de Aprendizaje I: Al terminar la octava unidad, cada una de las alumnas, comprendiendo la gramática de los auxiliares should y must es capaz de expresar una mayor cantidad de acciones en forma obligación y sugerencia. Además es capaz de expresar ideas describiendo ocupaciones. Asume la necesidad de escribir cartas formales.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.2 Going places!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Tiempo Presente Simple y Futuro con Will **II:** Primer Condicional **III:** Colocaciones **IV:** Vocabulario de preposiciones de lugar y de tiempo **V:** Expresiones de conexión de ideas

Objetivos de Aprendizaje I: Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas de futuro y las aplican adecuadamente. Describen condiciones acuciosamente. Asumen expresiones para demostrar ubicación de lugar. Utilizan expresiones de tiempo y conectores para unir ideas varias.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.3 Scared to death!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Patrones de Verbos Infinitivos y gerundios **II:** What + Infinitivo **III:** Something + infinitive **IV:** Expresiones de sentimientos **V:** Exclamaciones de sorpresa

Objetivos de Aprendizaje I: Al terminar la décimo unidad los alumnos serán capaces de reconocer y utilizar patrones tiempos en pasado los utilizan adecuadamente. Utilizan expresiones de exclamación. Y describen sentimientos.Utilizarán conjunciones para unir ideas tipo.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.4 Things that changed the world!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Voz Pasiva **II:** Oraciones Afirmativas, Negativas y Preguntas **III:** Uso de Participios, verbos y sustantivos que van unidos **IV:** Señales. Signos y notas **V:** Resúmenes **VI:** Expresiones para indicar prohibición

Objetivos de Aprendizaje I: Al terminar la décimo primera primera unidad, los alumnos habiendo identificado la idea de acciones pasivas describen acciones adecuadamente en diversas situaciones que la involucran. Reconocen y aplican participios. Asumen la idea de respetar signos y señales públicas. Expresan ideas de hábitos . Hacen resúmenes.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.5 Dreams and reality!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Segundo Condicional **II:** Auxiliar de Modo Might **III:** Verbos de Frase **IV:** Vocabulario de expresiones sociales **V:** Adverbios **VI:** Expresiones para dar consejo

Objetivos de Aprendizaje I: Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión de la idea de Condicionales y de expresar posibilidad elaborarán oraciones utilizando los elementos necesarios.Asimilarán además la necesidad de frases verbales (verbos de 2 palabras).Adquirirán vocabulario para describir expresiones sociales.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.6 Making a living!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Present Perfect Continuous **II:** Present Continuous **III:** Ocupaciones **IV:** Formación de palabras **V:** Adverbios **VI:** Expresiones de uso en el teléfono.

Objetivos de Aprendizaje I: Al finalizar la décimo tercera unidad estructuran oraciones con acciones que incluyen presente y pasado en contextos adecuados. Enfatizan la diferencia entre tipos de ocupaciones. Utilizan expresiones adecuadas para conversaciones telefónicas

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.7 All you need is love!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Pasado Perfecto y Pasado Simple **II:** Expresiones de Reporte **III:** Expresiones de palabras en contextos diferentes **IV:** Despedidas cortas y formales **V:** Historias de amor

Objetivos de Aprendizaje I: Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo pasado perfecto, lo diferencian del pasado simple. Enfatizan la diferencia entre palabras en contextos diferentes. Describen ideas de despedidas. Utilizan expresiones para escribir historias de amor. Asumen la idea de dar y hacer entrevistas.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

6. Bibliografía

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002d] Soars, L. and John (2002d). *American Headway N 2 Student Book*. Editorial Oxford.

Séptimo Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS261 Inteligencia Artificial (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - MA251 Introducción a la Estadística (5to Sem-Pág 393)

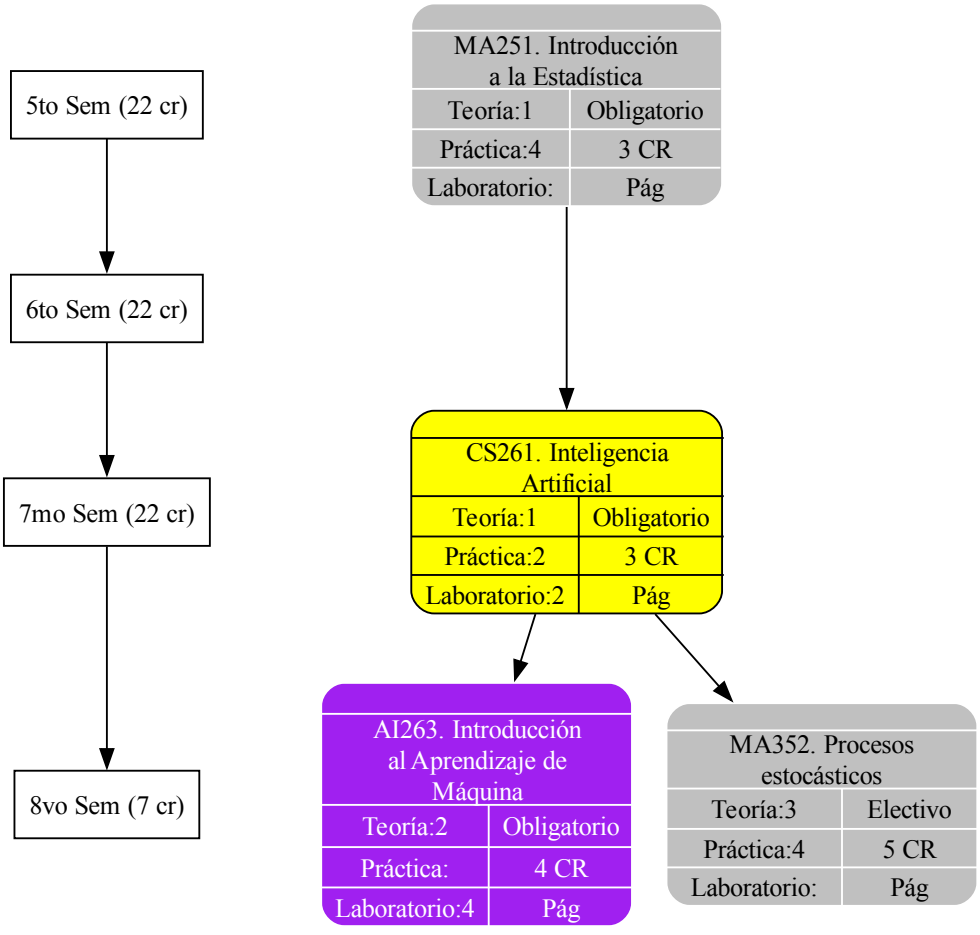


Figura 7.37: Mapa de Conexión. CS261 Inteligencia Artificial

2. Fundamentación

La Inteligencia Artificial (IA) es un campo fundamental de la ciencia de la computación que busca crear sistemas capaces de realizar tareas que normalmente requieren inteligencia humana. Este curso proporciona una introducción a las técnicas utilizadas para construir dichos sistemas, incluyendo algoritmos de búsqueda, razonamiento probabilístico y aprendizaje automático. Los estudiantes

aprenderán a modelar problemas complejos e implementar agentes inteligentes, mientras consideran las implicaciones sociales y éticas de estas tecnologías.

3. Objetivos del curso.

- Comprender los conceptos fundamentales y la historia de la Inteligencia Artificial.
- Implementar y evaluar algoritmos de búsqueda para la resolución de problemas.
- Dominar los conceptos básicos del aprendizaje automático, incluyendo aprendizaje supervisado y no supervisado.
- Aplicar razonamiento probabilístico para manejar la incertidumbre en sistemas inteligentes.
- Analizar los desafíos éticos y el impacto social de la IA.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Introducción a la IA: Conceptos y Comportamiento Inteligente, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Visión general de los problemas de IA, Ejemplos de aplicaciones recientes exitosas de IA **II:** Definiciones de agentes con ejemplos (por ejemplo, reactivos, deliberativos) **III:** ¿Qué es el comportamiento inteligente?: *a*) La prueba de Turing y sus defectos *b*) Entrada y salida multimodal *c*) Simulación del comportamiento inteligente *d*) Razonamiento racional versus no racional **IV:** Visión general de las Aplicaciones de IA, crecimiento e impacto (económico, social, ético) **V:** Problemas filosóficos **VI:** Historia de la IA

Objetivos de Aprendizaje **I:** Describir la prueba de Turing y el experimento mental de la "Habitación China" [Explicar] **II:** Diferenciar entre razonamiento/comportamiento óptimo y razonamiento/comportamiento similar al humano [Evaluar] **III:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Explicar]

Bibliografía: [Russell and Norvig, 2020, Luger, 2008]

5.2 Características del Problema y Naturaleza de los Agentes, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Características del problema: *a*) Totalmente observable versus parcialmente observable *b*) Agente único versus multiagente *c*) Determinista versus estocástico *d*) Estático versus dinámico *e*) Discreto versus continuo **II:** Naturaleza de los agentes: *a*) Autonomía autónoma, semiautónoma, de iniciativa mixta *b*) Reflexivos, basados en metas y basados en utilidad *c*) Toma de decisiones bajo incertidumbre y con información incompleta *d*) La importancia de la percepción y las interacciones ambientales *e*) Agentes basados en aprendizaje *f*) Agentes corporeizados: 1) sensores, dinámicas, efectores **III:** Practicar la identificación de características del problema en entornos de ejemplo **IV:** Profundidad adicional sobre la naturaleza de los agentes con ejemplos **V:** Profundidad adicional sobre Aplicaciones de IA, Crecimiento e Impacto (económico, social, ético, seguridad)

Objetivos de Aprendizaje **I:** Enumerar las características de un problema específico [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Luger, 2008]

5.3 Estado y Búsqueda No Informada, 5 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Representación del espacio de estados de un problema: *a*) Especificación de estados, metas y operadores *b*) Factorización de estados en representaciones (espacios de hipótesis) *c*) Resolución de problemas mediante búsqueda en grafos: 1) por ejemplo, Grafos como un espacio y recorridos de árboles como exploración de ese espacio 2) Construcción dinámica del grafo (no dada de antemano) **II:** Búsqueda en grafo no informada para resolución de problemas: *a*) Búsqueda en amplitud *b*) Búsqueda en profundidad: 1) Con profundización iterativa *c*) Búsqueda de costo uniforme **III:** Complejidades de espacio y tiempo de los algoritmos de búsqueda en grafos **IV:** Búsqueda bidireccional **V:** Comprensión del espacio de búsqueda: *a*) Construcción de árboles de búsqueda *b*) Espacios de búsqueda dinámicos *c*) Explosión combinatoria del espacio de búsqueda *d*) Topología del espacio de búsqueda (por ejemplo, crestas, puntos de silla, mínimos locales)

Objetivos de Aprendizaje **I:** Diseñar la representación del espacio de estados para un rompecabezas (por ejemplo, N-reinas o problema de los 3 jarrones) [Aplicar] **II:** Seleccionar e implementar un algoritmo de búsqueda no informada apropiado para un problema (por ejemplo, tres en raya), y caracterizar sus complejidades de tiempo y espacio [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Luger, 2008]

5.4 Búsqueda Heurística y CSP, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Búsqueda en grafo heurística para resolución de problemas: *a*) Construcción heurística y admisibilidad *b*) Ascenso de colinas *c*) Mínimos locales y el paisaje de búsqueda: 1) Soluciones locales versus globales *d*) Búsqueda primero el mejor voraz *e*) Búsqueda A* **II:** Búsqueda en haz **III:** Implementación de la búsqueda A* **IV:** Satisfacción de restricciones **V:** Variaciones de A* (IDA*, SMA*, RBFS)

Objetivos de Aprendizaje **I:** Seleccionar e implementar un algoritmo de búsqueda informada apropiado para un problema después de diseñar una función heurística útil (por ejemplo, un robot navegando en un mundo de cuadrícula 2D) [Aplicar] **II:** Evaluar si una heurística para un problema dado es admisible/puede garantizar una solución óptima [Evaluar] **III:** Diseñar e implementar búsqueda A*/en haz para resolver un problema, y compararla con otros algoritmos de búsqueda en términos del costo de la solución, número de nodos expandidos, etc [Aplicar] **IV:** Comparar y contrastar varias búsquedas heurísticas con respecto a la aplicabilidad a un problema dado [Evaluar] **V:** Modelar un rompecabezas lógico o Sudoku como un problema de satisfacción de restricciones, resolverlo con búsqueda con retroceso y determinar cuánto la consistencia de arcos puede reducir el espacio de búsqueda [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Luger, 2008]

5.5 Búsqueda Adversarial y Optimización Estocástica, 5 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: **I:** Juegos adversariales de dos jugadores: *a*) Búsqueda minimax *b*) Poda alfa-beta: 1) Corte por profundidad **II:** Búsqueda local **III:** Búsqueda tabú **IV:** Juegos adversariales de dos jugadores: *a*) El efecto horizonte *b*) Libros de apertura/soluciones de finales *c*) Qué significa ”resolver” un juego (por ejemplo, damas) **V:** Implementación de búsqueda minimax, búsqueda en haz **VI:** Búsqueda expectimax (resolución MDP) y nodos de azar **VII:** Búsqueda estocástica: *a*) Recocido simulado *b*) Algoritmos genéticos *c*) Búsqueda en árbol Monte Carlo

Objetivos de Aprendizaje **I:** Aplicar búsqueda minimax en un juego adversarial de dos jugadores (por ejemplo, conecta 4), usando evaluación heurística en una profundidad particular para calcular las puntuaciones a propagar [Aplicar] **II:** Aplicar búsqueda minimax con poda alfa-beta para

podar el espacio de búsqueda en un juego adversarial de dos jugadores (por ejemplo, conecta 4) [Aplicar] **III:** Comparar y contrastar algoritmos genéticos con técnicas de búsqueda clásicas, explicando cuándo es más apropiado usar un algoritmo genético para aprender un modelo versus otras formas de optimización (por ejemplo, descenso de gradiente) [Evaluar] **IV:** Diseñar e implementar una solución de algoritmo genético para un problema [Aplicar] **V:** Diseñar e implementar un programa de recocido simulado para evitar mínimos locales en un problema [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Luger, 2008]

5.6 Fundamentos de Aprendizaje Automático, 7 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: *a*) Aprendizaje supervisado: 1) Clasificación 2) Regresión *b*) Aprendizaje por refuerzo *c*) Aprendizaje no supervisado: 1) Agrupamiento **II:** Ideas fundamentales: *a*) Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. *b*) Fuentes de error e indecidibilidad en el aprendizaje automático **III:** Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: *a*) Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente **IV:** El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): *a*) El compromiso entre sesgo (subajuste) - varianza (sobreajuste)

Objetivos de Aprendizaje I: Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar] **II:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] **III:** Formular una aplicación como un problema de clasificación, incluyendo las características de entrada disponibles y la salida a predecir (por ejemplo, identificar caracteres alfabéticos a partir de entrada de cuadrícula de píxeles) [Aplicar] **IV:** Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] **V:** Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

Bibliografía: [Russell and Norvig, 2020, Alpaydin, 2020]

5.7 Aprendizaje Automático: Datos, Representaciones y Evaluación, 7 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Trabajar con Datos: *a*) Preprocesamiento de datos: 1) Importancia y peligros de las elecciones de preprocesamiento *b*) Manejo de valores faltantes (imputación, marcar como faltante): 1) Implicaciones de imputar versus marcar como faltante *c*) Codificación de variables categóricas, codificación de datos de valor real *d*) Normalización/estandarización *e*) énfasis en datos reales, no ejemplos de libros de texto **II:** Representaciones: *a*) Espacios de hipótesis y complejidad *b*) Expansión de características de base simple, como elevar al cuadrado características univariadas *c*) Representaciones de características aprendidas **III:** Evaluación del aprendizaje automático: *a*) Separación de conjuntos de entrenamiento, validación y prueba *b*) Métricas de rendimiento para clasificadores *c*) Estimación del rendimiento en prueba en datos retenidos *d*) Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación *e*) Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones

Objetivos de Aprendizaje I: Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] **II:** Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] **III:** Explicar los procedimientos

adecuados de evaluación de AA, incluyendo las diferencias entre rendimiento en entrenamiento y prueba, y qué puede salir mal con el proceso de evaluación llevando a informes inexactos del rendimiento de AA [Explicar] **IV:** Comparar dos algoritmos de aprendizaje automático en un conjunto de datos, implementando la metodología de preprocesamiento de datos y evaluación (por ejemplo, métricas y manejo de divisiones entrenamiento/prueba) desde cero [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Alpaydin, 2020]

5.8 Redes Neuronales y Ética, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Redes neuronales básicas: *a*) Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos *b*) Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) **II:** ética para el Aprendizaje Automático: *a*) Enfocarse en datos reales, escenarios reales y estudios de caso *b*) Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas

Objetivos de Aprendizaje I: Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]

Bibliografía: [Russell and Norvig, 2020, Alpaydin, 2020]

5.9 Formulación del Aprendizaje Automático y Evaluación, 3 hr(s)

Competencias: ABET-2

Tópicos: I: Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: *a*) Función objetivo *b*) Descenso de gradiente *c*) Regularización para evitar sobreajuste (formulación matemática) **II:** Conjuntos de modelos: *a*) Combinación de mayoría ponderada simple **III:** Evaluación de rendimiento: *a*) Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) *b*) Métricas de rendimiento para regresores *c*) Matriz de confusión *d*) Validación cruzada: 1) Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)

Objetivos de Aprendizaje I: Comparar y contrastar varias técnicas de aprendizaje (por ejemplo, árboles de decisión, regresión logística, Bayes ingenuo, redes neuronales y redes de creencia), proporcionando ejemplos de cuándo cada estrategia es superior [Comparar] **II:** Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] **III:** Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] **IV:** Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] **V:** Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]

Bibliografía: [Goodfellow et al., 2016]

5.10 Aprendizaje Profundo, RL y Ética, 3 hr(s)

Competencias: ABET-2

Tópicos: I: Aprendizaje profundo: *a*) Redes feed-forward profundas (solo intuición, sin matemáticas) *b*) Redes neuronales convolucionales (solo intuición, sin matemáticas) *c*) Visualización de representaciones de características aprendidas de redes profundas *d*) Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) **II:** Visión general de los métodos de aprendizaje por refuerzo **III:** Dos o más aplicaciones de algoritmos de aprendizaje automático: *a*) Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego **IV:** ética para el Aprendizaje Automático: *a*) Continuar enfocándose en datos reales, escenarios reales y estudios de caso *b*) Privacidad *c*) Equidad *d*) Propiedad intelectual *e*) Explicabilidad

Objetivos de Aprendizaje I: Visualizar el progreso del entrenamiento de una red neuronal a través de curvas de aprendizaje en un kit de herramientas establecido (por ejemplo, TensorBoard) y visualizar las características aprendidas de la red [Aplicar] **II:** Dada una aplicación real de aprendizaje automático, describir problemas éticos respecto a las elecciones de datos, pasos de preprocesamiento, selección de algoritmo y visualización/presentación de resultados [Aplicar]

Bibliografía: [Goodfellow et al., 2016]

5.11 Representación del Conocimiento y Razonamiento Probabilístico Básico, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Tipos de representaciones: *a*) Simbólicas, lógicas: 1) Crear una representación a partir de un enunciado de problema en lenguaje natural *b*) Representaciones subsimbólicas aprendidas *c*) Modelos gráficos (por ejemplo, Bayes ingenuo, red bayesiana) **II:** Repaso del razonamiento probabilístico, teorema de Bayes **III:** Razonamiento bayesiano: *a*) Inferencia bayesiana **IV:** Variables aleatorias y distribuciones de probabilidad: *a*) Axiomas de probabilidad *b*) Inferencia probabilística *c*) Regla de Bayes (derivación) *d*) Inferencia bayesiana (ejemplos más complejos)

Objetivos de Aprendizaje I: Dado un enunciado de problema en lenguaje natural, codificarlo como una representación simbólica o lógica [Aplicar] **II:** Calcular una inferencia probabilística en un problema del mundo real usando el teorema de Bayes para determinar la probabilidad de una hipótesis dada la evidencia [Aplicar] **III:** Aplicar la regla de Bayes para determinar la probabilidad de una hipótesis dada la evidencia [Aplicar] **IV:** Calcular la probabilidad de resultados y probar si los resultados son independientes [Aplicar]

Bibliografía: [Russell and Norvig, 2020]

5.12 Independencia, Modelos de Markov y Toma de Decisiones, 6 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Independencia **II:** Independencia condicional **III:** Cadenas de Markov y modelos de Markov **IV:** Utilidad y toma de decisiones

Objetivos de Aprendizaje I: Explicar cómo podemos tomar decisiones bajo incertidumbre, usando conceptos como el teorema de Bayes y la utilidad [Explicar]

Bibliografía: [Russell and Norvig, 2020]

5.13 Aplicaciones e Impacto Social de la IA, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Al menos una aplicación de IA a un problema y campo específico, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir al menos uno para el Núcleo CS): *a*) Formular y evaluar una aplicación específica como un problema de IA: 1) Cómo lidiar con problemas mal especificados o mal planteados *b*) Disponibilidad/escasez y limpieza de datos: 1) Limpieza y preprocesamiento básico de datos 2) Sesgo del conjunto de datos *c*) Sesgo algorítmico *d*) Sesgo de evaluación *e*) Evaluación de las implicaciones sociales de la aplicación **II:** Modelos generativos profundos desplegados: *a*) Visión general de alto nivel de modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion, etc.), sus usos y sus limitaciones/peligros. *b*) Visión general de alto nivel de modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard, etc.), sus usos y sus limitaciones/peligros. **III:** Visión general del impacto social de la IA: *a*) ética *b*) Equidad *c*) Confianza/explicabilidad *d*) Privacidad y uso de datos de entrenamiento *e*) Autonomía humana y supervisión/regulaciones/requisitos legales *f*) Sostenibilidad

Objetivos de Aprendizaje I: Dado un dominio de aplicación y problema del mundo real, formular una solución de IA para ello, identificando datos/entrada apropiados, preprocesamiento, representaciones, técnicas de IA y métricas/metodología de evaluación [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Alpaydin, 2020]

5.14 Aplicaciones Adicionales e Impacto Social de la IA, 4 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Una o más aplicaciones adicionales de IA a un amplio conjunto de problemas y diversos campos, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir un área diferente de la elegida para el Núcleo CS): *a*) Formular y evaluar una aplicación específica como un problema de IA: 1) Cómo lidiar con problemas mal especificados o mal planteados *b*) Disponibilidad/escasez y limpieza de datos: 1) Limpieza y preprocesamiento básico de datos 2) Sesgo del conjunto de datos *c*) Sesgo algorítmico *d*) Sesgo de evaluación *e*) Evaluación de las implicaciones sociales de la aplicación **II:** Profundidad adicional sobre modelos generativos profundos desplegados: *a*) Introducción a cómo funcionan los modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion) incluyendo discusión de atención *b*) Introducción a cómo funcionan los modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard) incluyendo discusión de atención *c*) Idea de modelos fundamentales, cómo usarlos, y los beneficios/problemas con entrenarlos a partir de grandes datos **III:** Análisis y discusión del impacto social de la IA: *a*) ética *b*) Equidad *c*) Confianza/explicabilidad *d*) Privacidad y uso de datos de entrenamiento *e*) Autonomía humana y supervisión/regulaciones/requisitos legales *f*) Sostenibilidad

Objetivos de Aprendizaje I: Analizar el impacto social de una o más aplicaciones específicas de IA del mundo real, identificando problemas respecto a ética, equidad, sesgo, confianza y explicabilidad [Evaluar] **II:** Describir algunos de los modos de falla de los modelos generativos profundos actuales para lenguaje o imágenes, y cómo esto podría afectar su uso en una aplicación [Describir]

Bibliografía: [Russell and Norvig, 2020, Alpaydin, 2020]

6. Bibliografía

[Alpaydin, 2020] Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press, 4th edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Luger, 2008] Luger, G. F. (2008). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson, 6th edition.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Alpaydin, 2020] Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press, 4th edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Luger, 2008] Luger, G. F. (2008). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson, 6th edition.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS3P1 Computación Paralela y Distribuída (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 361)

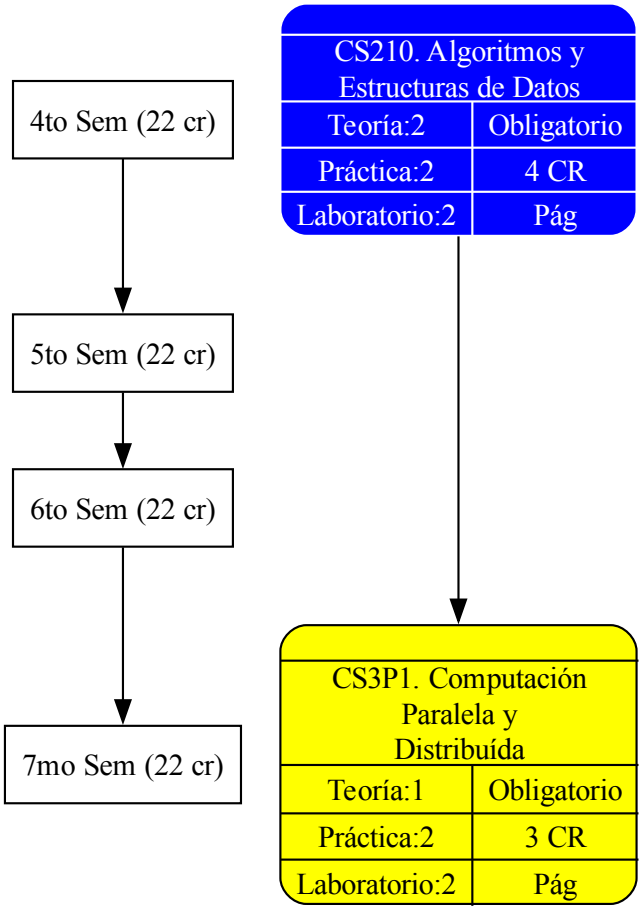


Figura 7.38: Mapa de Conexión. CS3P1 Computación Paralela y Distribuída

2. Fundamentación

Con el fin del escalado de frecuencia en procesadores de un solo núcleo, la computación paralela y distribuida se ha vuelto esencial para el desarrollo de software de alto rendimiento. Este curso introduce a los estudiantes a los modelos de programación, patrones de comunicación y mecanismos de coordinación necesarios para explotar sistemas de múltiples núcleos y clústeres distribuidos. Los estudiantes aprenderán a diseñar algoritmos escalables y evaluar su rendimiento bajo diferentes restricciones computacionales.

3. Objetivos del curso.

- Diseñar e implementar programas paralelos utilizando memoria compartida y distribuida.
- Dominar protocolos de comunicación y sincronización en sistemas distribuidos.
- Evaluar el rendimiento y la escalabilidad mediante métricas formales.
- Aplicar algoritmos paralelos para resolver problemas computacionales complejos.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Programas: Paralelismo Declarativo, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Paralelismo *a*) Paralelismo declarativo: Determinar qué acciones pueden, o no deben, realizarse en paralelo, a nivel de instrucciones, funciones, clausuras, acciones compuestas, sesiones, tareas y servicios es la idea principal que subyace a los algoritmos PDC; no hacerlo es la principal fuente de errores. Ver también: Algoritmos *b*) Definir orden: por ejemplo, usando relaciones de "sucede antes" o grafos acíclicos dirigidos serie/paralelo que representan programas. *c*) Independencia: determinar cuándo el orden no importa, en términos de conmutatividad, dependencias, precondiciones. *d*) Garantizar el orden entre acciones que de otro modo serían paralelas cuando sea necesario, incluyendo bloqueos, publicación segura; e imponer comunicación - enviar un mensaje sucede antes de recibirlo; y relajarlo cuando no sea necesario.

Objetivos de Aprendizaje I: Mostrar gráficamente (como un Grafo Acíclico Dirigido - *Directed Acyclic Graph* (DAG)) cómo paralelizar una expresión numérica compuesta; por ejemplo, $a = (b + c) * (d + e)$ [Diseñar] **II:** Explicar por qué los conceptos de consistencia y tolerancia a fallos no surgen en programas puramente secuenciales [Explicar]

Bibliografía: [Pacheco and Malensek, 2021]

5.2 Programas: Inicio de Actividades, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Iniciar actividades *a*) Las opciones que permiten que las acciones se realicen (eventualmente) en lugares van desde el cableado directo hasta scripts de configuración; también establecer comunicación y gestión de recursos; estos se expresan de manera diferente en los lenguajes y contextos, generalmente confiando en el aprovisionamiento y gestión automatizados por las plataformas. Ver también: Gestión de Recursos *b*) Procedimental: Permitir que múltiples acciones comiencen en un punto de programa dado; por ejemplo, iniciar nuevos hilos, posiblemente delimitando su alcance u organizándolos en grupos jerárquicos. *c*) Reactiva: Habilitar al ocurrir un evento instalando un manejador de eventos, con menos control de cuándo comienzan o terminan las acciones, y puede aplicarse incluso en monoprocesadores. *d*) Dependiente: Habilitar al completarse otras; por ejemplo, secuenciando conjuntos de acciones paralelas. Ver también: Coordinación. *e*) Granularidad: El costo de ejecución de los cuerpos de las acciones debe superar la sobrecarga de organizarlos.

Objetivos de Aprendizaje I: Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]

Bibliografía: [Pacheco and Malensek, 2021]

5.3 Programas: Propiedades de Ejecución, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Propiedades de Ejecución *a*) Ejecución no determinista de acciones sin orden. *b*) Consistencia: Garantizar el acuerdo entre partes sobre valores y predicados cuando sea necesario para evitar carreras, mantener seguridad y atomicidad, o llegar a consenso. *c*) Tolerancia a fallos: Manejar fallos en las partes o la comunicación, incluyendo (Bizantino) mal comportamiento debido a partes y protocolos no confiables, cuando sea necesario para mantener el progreso o la disponibilidad. Ver también: Confiabilidad del Sistema. *d*) Los compromisos son un foco de evaluación. Ver también: Evaluación.

Objetivos de Aprendizaje I: Escribir una función que cuente eventos, como recepciones de paquetes de red, de manera eficiente [Escribir]

Bibliografía: [van Steen and Tanenbaum, 2023]

5.4 Programas: Distribución, 2 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Distribución *a*) Definir lugares, como dispositivos que ejecutan acciones, incluyendo componentes de hardware, hosts remotos; también pueden incluir dispositivos, hosts y usuarios externos no controlados. Ver también: Interconexión y Comunicación *b*) Un dispositivo puede dividir el tiempo o emular múltiples acciones paralelas mediante menos procesadores mediante planificación y virtualización. Ver también: Planificación *c*) Nombrar o identificar lugares (por ejemplo, IDs de dispositivo) y acciones como partes (por ejemplo, IDs de hilo). *d*) Las actividades a través de lugares pueden comunicarse a través de medios. Ver también: Comunicación

Objetivos de Aprendizaje I: Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir]

Bibliografía: [van Steen and Tanenbaum, 2023]

5.5 Programas: Mapeos de Implementación, 2 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Una o más de las siguientes asignaciones y mecanismos a través de sistemas en capas: *a*) Paralelismo a nivel de instrucción y datos en la CPU. Ver también: Organización Funcional. *b*) SIMD y paralelismo de datos heterogéneo. Ver también: Arquitecturas Heterogéneas. *c*) Concurrencia planificada en multinúcleo, tareas, actores. Ver también: Planificación. *d*) Clústeres, nubes; aprovisionamiento elástico. Ver también: Aspectos Comunes: Plataformas, APIs y Restricciones, Aspectos Comunes: Lenguajes de Plataforma y Patrones Web. *e*) Sistemas distribuidos en red. Ver también: Aplicaciones en Red. *f*) Tecnologías emergentes como la computación cuántica y la computación molecular.

Objetivos de Aprendizaje I: Explicar las compensaciones entre diferentes estrategias de mapeo en términos de rendimiento, costo y complejidad de implementación [Explicar]

Bibliografía: [Pacheco and Malensek, 2021, mei W. Hwu et al., 2022]

5.6 Programación GPU, 4 hr(s)

Competencias: ABET-2

Tópicos: I: Computación de Propósito General en GPU (GPGPU) *a*) Arquitectura de GPU: multiprocesadores de flujo (SMs), núcleos CUDA y modelo de ejecución por warp. *b*) Modelo de ejecución SIMT (Single Instruction, Multiple Threads) y sus diferencias respecto al SIMD de CPU. *c*) Interacción host-dispositivo: transferencia de datos entre memoria de CPU (host) y

GPU (dispositivo) vía PCIe. *d*) Casos de uso: computación científica, aprendizaje automático, procesamiento de imágenes y simulación a gran escala. **II:** Modelo de Programación CUDA *a*) Jerarquía de ejecución CUDA: grids, bloques de hilos e hilos individuales; mapeo al hardware GPU. *b*) Definición y lanzamiento de kernels; parámetros de configuración (dimensiones de grid y bloque). *c*) Indexación de hilos: threadIdx, blockIdx, blockDim, gridDim; cálculo de índices globales en 1D, 2D y 3D. *d*) Divergencia de warp: implicaciones de las bifurcaciones dentro de un warp en el rendimiento. *e*) Primitiva de sincronización intra-bloque: __syncthreads(). **III:** Jerarquía de Memoria CUDA *a*) Memoria global: grande, alta latencia, accesible por todos los hilos; patrones de acceso coalescente para el rendimiento. *b*) Memoria compartida: memoria en chip de baja latencia compartida dentro de un bloque de hilos; conflictos de banco. *c*) Registros y memoria local: almacenamiento privado por hilo. *d*) Memoria constante y de textura: cachés de solo lectura optimizadas para patrones de acceso específicos. *e*) Memoria unificada: modelo de programación simplificado con migración automática de datos entre host y dispositivo.

Objetivos de Aprendizaje I: Escribir un kernel CUDA que realice una operación data-paralela (p. ej., suma de vectores, multiplicación de matrices) y configurar correctamente las dimensiones de grid y bloque [Escribir] **II:** Analizar el impacto de los patrones de acceso a memoria global (acceso coalescente vs. acceso entrelazado) y el uso de memoria compartida en el rendimiento de un kernel GPU [Analizar] **III:** Diseñar un algoritmo paralelo usando el modelo de programación CUDA, incluyendo la jerarquía de hilos, estrategia de asignación de memoria y transferencias de datos entre host y dispositivo [Diseñar]

Bibliografía: [Kirk and mei W. Hwu, 2016, Corporation, 2024]

5.7 Comunicación, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Medios *a*) Variedades: canales (paso de mensajes o E/S), memoria compartida, heterogéneos, almacenes de datos. *b*) Dependencia de la disponibilidad y naturaleza del hardware subyacente, conectividad y protocolos; soporte del lenguaje, emulación. Ver también: Interconexión y Comunicación. **II:** Canales *a*) Medios de comunicación de parte a parte explícitos (generalmente nombrados). *b*) APIs: Sockets, constructos arquitectónicos, basados en lenguaje y de kits de herramientas, como Message Passing Interface (MPI), y constructos en capas como Llamada a Procedimiento Remoto (RPC). Ver también: Fundamentos de Redes y Comunicaciones. *c*) APIs de canales de E/S. **III:** Memoria *a*) Arquitecturas de memoria compartida en las que las partes se comunican directamente solo con la memoria en direcciones dadas, con extensiones a memoria heterogénea que admite múltiples almacenes de memoria con transferencia de datos explícita entre ellos; por ejemplo, memoria local y compartida de GPU, Acceso Directo a Memoria (DMA). *b*) Jerarquías de memoria: Múltiples capas de dominios, alcances y cachés compartidos; localidad: latencia, falso uso compartido. *c*) Propiedades de consistencia: Límites de atomicidad a nivel de bits, coherencia, orden local. **IV:** Almacenes de Datos *a*) Estructuras de datos mantenidas cooperativamente que implementan mapas y TADs relacionados. *b*) Variedades: Propiedad exclusiva, compartida, fragmentada, replicada, inmutable, versionada.

Objetivos de Aprendizaje I: Explicar las similitudes y diferencias entre: (1) La parte A envía un mensaje en el canal X con contenido 1 recibido por la parte B (2) A establece la variable compartida X a 1, leída por B (3) A establece "X=1" en un mapa compartido distribuido al que accede B [Explicar] **II:** Escribir un programa que distribuya diferentes segmentos de un conjunto de datos a múltiples trabajadores y recoja los resultados (por ejemplo, sumar segmentos de un arreglo) [Escribir] **III:** Escribir un programa paralelo que solicite datos de múltiples sitios y los resuma usando alguna forma de reducción [Escribir] **IV:** Comparar el rendimiento de versiones con y sin búfer de un programa productor-consumidor [Comparar]

Bibliografía: [van Steen and Tanenbaum, 2023, Kleppmann, 2017]

5.8 Comunicación: Propiedades y Extensiones, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Una o más de las siguientes propiedades y extensiones: *a*) Topologías: Unicast, Multicast, Buzones, Conmutadores; Enrutamiento a través de redes de interconexión de hardware y software. *b*) Propiedades de concurrencia de medios: Orden, consistencia, idempotencia, superposición de comunicación con computación. *c*) Rendimiento del medio: Latencia, ancho de banda (rendimiento), contención (congestión), capacidad de respuesta (vivacidad), confiabilidad (tasas de error y pérdida), progreso basado en protocolo (acks, tiempos de espera, mediación). *d*) Propiedades de seguridad del medio: integridad, privacidad, autenticación, autorización. Ver también: Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada. *e*) Formatos de datos: Serialización, validación, cifrado, compresión. *f*) Políticas de canal: Puntos finales, sesiones, almacenamiento en búfer, respuesta a saturación (esperar vs. descartar), control de velocidad. *g*) Multiplexación y demultiplexación de muchos dispositivos o partes de E/S relativamente lentos; técnicas basadas en finalización y basadas en planificador; APIs async-await, select y polling. *h*) Formalización y análisis de comunicación por canales; por ejemplo, CSP. *i*) Aplicaciones de la teoría de colas para modelar y predecir el rendimiento.

Objetivos de Aprendizaje I: Determinar si un esquema de comunicación dado proporciona propiedades de seguridad suficientes para un uso dado [Determinar] **II:** Dar un ejemplo de un escenario en el que los envíos de mensajes bloqueantes pueden causar un punto muerto (deadlock) [Crear] **III:** Describir al menos una técnica de diseño para evitar fallos de vivacidad en programas que usan múltiples bloqueos [Describir]

Bibliografía: [van Steen and Tanenbaum, 2023, Kleppmann, 2017]

5.9 Memoria y Consistencia, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Modelos de memoria, consistencia de datos y tolerancia a fallos: *a*) Modelos de memoria: consistencia secuencial y de liberación/adquisición. *b*) Gestión de memoria; incluyendo la reclamación de datos compartidos; conteo de referencias y alternativas. *c*) Colocación y transferencia masiva de datos; reducción del tráfico de mensajes y mejora de la localidad; superposición de transferencia de datos y computación; impacto del diseño de datos como array de estructuras vs. estructura de arrays. *d*) Emular memoria compartida: memoria compartida distribuida, Acceso Directo a Memoria Remota (RDMA). *e*) Consistencia de almacenes de datos: Atomicidad, linealizabilidad, transaccionalidad, coherencia, orden causal, resolución de conflictos, consistencia eventual, cadenas de bloques. *f*) Fallos, particionamiento y fallos parciales; votación; protocolos como Paxos y Raft. *g*) Compromisos de diseño entre consistencia, disponibilidad, tolerancia a partición (fallos); imposibilidad de cumplir con todos a la vez. *h*) Seguridad y confianza: Fallos bizantinos, prueba de trabajo y alternativas.

Objetivos de Aprendizaje I: Dar un ejemplo de un orden de accesos entre actividades concurrentes (por ejemplo, un programa con una carrera de datos) que no sea secuencialmente consistente [Crear] **II:** Escribir un programa que ilustre el reordenamiento de acceso a memoria o de mensajes [Escribir] **III:** Describir los méritos relativos del control de concurrencia optimista versus conservador bajo diferentes tasas de contención entre actualizaciones [Describir] **IV:** Dar un ejemplo de un escenario en el que un intento de actualización optimista podría nunca completarse [Crear] **V:** Modificar un sistema concurrente para usar un almacén de datos más escalable, confiable o disponible [Crear] **VI:** Usando una plataforma existente que admita almacenes de datos replicados, escribir un programa que mantenga un mapeo clave-valor incluso cuando uno o más hosts fallen [Escribir]

Bibliografía: [van Steen and Tanenbaum, 2023, Kleppmann, 2017]

5.10 Coordinación, 7 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Dependencias *a*) La iniciación o progreso de una actividad puede depender de otras actividades, para evitar condiciones de carrera, garantizar la terminación o cumplir otros requisitos. *b*) Garantizar el progreso evitando ciclos de dependencia, usando condiciones monótonas, eliminando dependencias no esenciales. **II:** Constructos de control y patrones de diseño: *a*) Basados en finalización: Barreras, reuniones (joins), incluido el control de terminación. *b*) Habilitados por datos: Colas, diseños productor-consumidor. *c*) Basados en condición: Polling, reintentos, retrocesos, ayuda, suspensión, señalización, tiempos de espera. *d*) Reactivos: Habilitar y activar continuaciones.

Objetivos de Aprendizaje **I:** Mostrar cómo garantizar que un programa termine correctamente cuando todas las tareas concurrentes de un conjunto hayan finalizado [Diseñar] **II:** Escribir una función que cuente eventos, como entradas de sensores o recepciones de paquetes de red, de manera eficiente [Escribir] **III:** Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir] **IV:** Escribir un programa en el que la terminación de un conjunto de acciones paralelas sea seguida por otra [Escribir] **V:** Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]

Bibliografía: [Herlihy et al., 2020, van Steen and Tanenbaum, 2023]

5.11 Coordinación: Sincronización y Atomicidad, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Atomicidad *a*) Instrucciones atómicas, ordenamientos de acceso local impuestos. *b*) Bloqueos y exclusión mutua; granularidad de bloqueo. *c*) Uso de bloqueos en un lenguaje específico; mantener la vivacidad sin introducir carreras. *d*) Evitar punto muerto (*deadlock*): Ordenamiento, mayor granularidad, reintentos aleatorios; retrocesos, encapsulación mediante administradores de bloqueos. *e*) Errores comunes: No bloquear o desbloquear cuando es necesario, mantener bloqueos mientras se invocan operaciones desconocidas. *f*) Evitar bloqueos: replicación, solo lectura, propiedad, y construcciones no bloqueantes.

Objetivos de Aprendizaje **I:** Mostrar cómo evitar o reparar un error de carrera en un programa dado [Analizar]

Bibliografía: [Herlihy et al., 2020]

5.12 Coordinación: Propiedades Avanzadas, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Una o más de las siguientes propiedades y extensiones: *a*) Propiedades de progreso incluyendo libre de bloqueo (*lock-free*), libre de espera (*wait-free*), equidad (*fairness*), planificación por prioridad, interacciones con consistencia, confiabilidad. *b*) Rendimiento con respecto a contención, granularidad, convoy, escalabilidad. *c*) Estructuras de datos y algoritmos no bloqueantes. *d*) Propiedad y control de recursos. *e*) Variantes y alternativas de bloqueos: bloqueos de secuencia, bloqueos de lectura-escritura; Actualizar-Copiando-Lectura (RCU), reentrada; tickets; control del ciclo activo (*spinning*) versus bloqueo. *f*) Control basado en transacciones: Optimista y conservador. *g*) Bloqueo distribuido: confiabilidad. *h*) Alternativas a barreras: Relojes; contadores, relojes virtuales; flujo de datos y continuaciones; futuros y RPC; basadas en consenso, recolección de resultados con reductores y colectores. *i*) Especulación, selección, cancelación; consecuencias en observabilidad y seguridad. *j*) Control de recursos usando semáforos y variables de condición. *k*) Flujo de control: Planificación de computaciones, bucles serie-paralelo con líderes (posiblemente elegidos), tuberías y flujos, paralelismo anidado. *l*) Excepciones y fallos. Manejadores, detección, tiempos de espera, tolerancia a fallos, votación.

Objetivos de Aprendizaje I: Escribir un programa que busque especulativamente una solución mediante múltiples actividades, terminando las demás cuando se encuentre una [Escribir] **II:** Escribir un programa en el que una excepción numérica (como división por cero) en una actividad cause la terminación de las demás [Escribir] **III:** Escribir un programa para que múltiples partes acuerden la hora actual del día; discutir sus limitaciones en comparación con protocolos como el protocolo de transferencia de red (NTP) [Escribir]

Bibliografía: [Herlihy et al., 2020]

5.13 Evaluación, 7 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Requisitos de seguridad (safety) y vivacidad (*liveness*) en términos de constructos de lógica temporal para expresar "siempre" y "eventualmente" Ver también: Computación Paralela y Distribuida. **II:** Identificar, probar y reparar violaciones, incluyendo formas comunes de errores como no garantizar el orden necesario (errores de carrera), atomicidad (incluyendo errores de "verificar luego actuar") y terminación (bloqueo activo). **III:** Métricas de requisitos de rendimiento para rendimiento (*throughput*), capacidad de respuesta, latencia, disponibilidad, consumo de energía, escalabilidad, uso de recursos, costos de comunicación, espera y control de velocidad, equidad; acuerdos de nivel de servicio. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento. **IV:** Impacto en el rendimiento de las opciones de diseño e implementación, incluyendo granularidad, sobrecarga, costos de consenso y consumo de energía. Ver también: Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social. **V:** Estimar limitaciones de escalabilidad, por ejemplo usando la Ley de Amdahl o la Ley de Escalabilidad Universal. Ver también: Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento.

Objetivos de Aprendizaje I: Revisar una especificación para habilitar el paralelismo y la distribución sin violar otras propiedades o características esenciales [Rediseñar] **II:** Explicar cómo las nociones concurrentes de seguridad (safety) y vivacidad (liveness) extienden sus contrapartes secuenciales [Explicar] **III:** Especificar un conjunto de invariantes que deben mantenerse en cada paso de cómputo de paralelismo masivo [Analizar] **IV:** Escribir un programa de prueba que pueda revelar un error de carrera de datos; por ejemplo, perder una actualización cuando dos actividades intentan incrementar una variable [Escribir] **V:** En un contexto dado, explicar hasta qué punto se esperaría que introducir paralelismo en un programa por lo demás secuencial mejoraría el rendimiento (*throughput*) y/o reduciría la latencia, y cómo podría afectar la eficiencia energética [Explicar] **VI:** Mostrar cómo cambian la escalabilidad y la eficiencia para problemas de muestra con y sin el supuesto de que el tamaño del problema cambia con el número de procesadores; además, explicar si y cómo cambiaría la escalabilidad bajo relajaciones de dependencias secuenciales [Diseñar]

Bibliografía: [Sterling et al., 2024, Pacheco and Malensek, 2021]

5.14 Evaluación: Métodos Formales, 3 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Métodos formales de verificación y análisis: *a*) Extensiones a requisitos formales secuenciales como la linealizabilidad. *b*) Especificaciones de protocolo, sesión y transacciones. *c*) Uso de herramientas como Lenguaje Unificado de Modelado (UML), Lógica Temporal de Acciones (TLA), lógicas de programa. *d*) Análisis de seguridad: seguridad y vivacidad en presencia de comportamientos hostiles o con errores de otras partes; propiedades requeridas de mecanismos de comunicación (por ejemplo, ausencia de fugas entre capas), filtrado de entrada, limitación de velocidad. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad. *e*) Análisis estático aplicado a corrección, rendimiento (*throughput*), latencia, recursos, energía. Ver también: Diseño Sostenible y Computación Ubicua,

Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social. *f*) Análisis del modelo de Grafo Acíclico Dirigido (DAG) de eficiencia algorítmica (trabajo, span, caminos críticos).

Objetivos de Aprendizaje I: Especificar y medir el comportamiento cuando un servicio es solicitado por un número inesperadamente grande de clientes [Analizar] **II:** Identificar y reparar un problema de rendimiento debido a cuellos de botella secuenciales [Analizar] **III:** Comparar empíricamente el rendimiento (*throughput*) de dos implementaciones de un diseño común (quizás usando un marco de pruebas existente) [Comparar]

Bibliografía: [Sterling et al., 2024, Pacheco and Malensek, 2021]

5.15 Evaluación: Pruebas y Medición, 2 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Herramientas de prueba y técnicas de medición: *a*) Pruebas y depuración; herramientas como detectores de carreras, fuzzers, verificadores de dependencia de bloqueos, pruebas de unidad/esfuerzo/tortura, visualizaciones, integración continua (CI), despliegue continuo (CD), y generadores de pruebas. *b*) Medir y comparar rendimiento (*throughput*), sobrecarga, espera, contención, comunicación, movimiento de datos, localidad, uso de recursos, comportamiento en presencia de números excesivos de eventos, clientes o hilos. Ver también: Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento. *c*) Análisis específicos del dominio de aplicación y técnicas de evaluación.

Objetivos de Aprendizaje I: Identificar y reparar un problema de rendimiento debido a latencia de comunicación o datos [Analizar] **II:** Identificar y reparar un problema de rendimiento debido a sobrecarga en la gestión de recursos [Analizar] **III:** Identificar y reparar un problema de confiabilidad o disponibilidad [Analizar]

Bibliografía: [Sterling et al., 2024, Pacheco and Malensek, 2021]

5.16 Algoritmos, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Expresar e implementar algoritmos en lenguajes y marcos de trabajo dados, para iniciar actividades (por ejemplo hilos), usar constructos de memoria compartida, y APIs de canales, sockets y/o llamada a procedimiento remoto (RPC). Ver también: Computación Paralela y Distribuida. *a*) Ejemplos de datos paralelos incluyendo mapeo/reducción. *b*) Uso de APIs de canal, socket y/o RPC en un lenguaje dado, con control del programa para enviar (generalmente procedimental) vs recibir (generalmente reactivo o basado en RPC). *c*) Uso de bloqueos, barreras y/o sincronizadores para mantener la vivacidad sin introducir carreras.

Objetivos de Aprendizaje I: Implementar un componente paralelo/distribuido basado en un algoritmo conocido [Implementar] **II:** Escribir un programa de datos paralelos que, por ejemplo, calcule el promedio de un arreglo de números [Escribir] **III:** Escribir un programa productor-consumidor en el que un componente genere números y otro calcule su promedio. Medir las aceleraciones cuando los números son escalares pequeños versus valores de multiprecisión grandes [Escribir]

Bibliografía: [Pacheco and Malensek, 2021, Herlihy et al., 2020]

5.17 Algoritmos: Dominios de Aplicación, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Panorama de dominios de aplicación comunes en multinúcleo, reactivo, datos paralelos, clúster, nube, sistemas distribuidos abiertos y marcos de trabajo (con referencia a la siguiente tabla). *a*) Multinúcleo: Agentes de ejecución típicos: Hilos. Mecanismos de comunicación típicos: Memoria compartida, Atómicos, bloqueos. Dominios algorítmicos típicos: Gestión de recursos,

procesamiento de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*), latencia, energía. *b*) Reactivo: Agentes de ejecución típicos: Manejadores, hilos. Mecanismos de comunicación típicos: Canales de E/S. Dominios algorítmicos típicos: Servicios, tiempo real. Objetivos de ingeniería típicos: Latencia. *c*) Datos paralelos: Agentes de ejecución típicos: GPU, SIMD, aceleradores, híbridos. Mecanismos de comunicación típicos: Memoria heterogénea. Dominios algorítmicos típicos: álgebra lineal, gráficos, análisis de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*), energía. *d*) Clúster: Agentes de ejecución típicos: Hosts gestionados. Mecanismos de comunicación típicos: Sockets, canales. Dominios algorítmicos típicos: Simulación, análisis de datos. Objetivos de ingeniería típicos: Rendimiento (*throughput*). *e*) Nube: Agentes de ejecución típicos: Hosts aprovisionados. Mecanismos de comunicación típicos: APIs de servicio. Dominios algorítmicos típicos: Aplicaciones web. Objetivos de ingeniería típicos: Escalabilidad. *f*) Distribuido abierto: Agentes de ejecución típicos: Hosts autónomos. Mecanismos de comunicación típicos: Sockets, Almacenes de datos. Dominios algorítmicos típicos: Almacenes de datos y servicios tolerantes a fallos. Objetivos de ingeniería típicos: Confiabilidad.

Objetivos de Aprendizaje I: Extender un programa secuencial dirigido por eventos estableciendo una nueva actividad en un manejador de eventos (por ejemplo, un nuevo hilo en un manejador de acciones de GUI) [Diseñar] **II:** Mejorar el rendimiento de un componente secuencial introduciendo paralelismo y/o distribución [Crear] **III:** Elegir entre diferentes diseños paralelos/distribuidos para componentes de un sistema dado [Evaluar (valorar)]

Bibliografía: [Pacheco and Malensek, 2021, Herlihy et al., 2020]

5.18 Algoritmos: Dominios Algorítmicos, 4 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Uno o más de los siguientes dominios algorítmicos. Ver también: Estrategias Algorítmicas: *a*) álgebra lineal: Operaciones con vectores y matrices, precisión/estabilidad numérica, aplicaciones en análisis de datos y aprendizaje automático. *b*) Procesamiento de datos: ordenación, búsqueda y recuperación, estructuras de datos concurrentes. *c*) Grafos, búsqueda y combinatoria: Marcado, paralelización de aristas, acotamiento, especulación, análisis basado en redes. *d*) Modelado y simulación: ecuaciones diferenciales; aleatorización, problemas de N-cuerpos, algoritmos genéticos. *e*) Lógica computacional: satisfactibilidad (SAT), programación lógica concurrente. *f*) Gráficos y geometría computacional: Transformaciones, renderizado, trazado de rayos. *g*) Gestión de recursos: Asignar, colocar, reciclar y planificar procesadores, memoria, canales y hosts; recursos exclusivos vs compartidos; algoritmos estáticos, dinámicos y elásticos; Restricciones de tiempo real; Lotes, priorización, partición; descentralización mediante robo de trabajo y técnicas relacionadas. *h*) Servicios: Implementar APIs web, moneda electrónica, sistemas de transacción, juegos multijugador.

Objetivos de Aprendizaje I: Diseñar, implementar, analizar y evaluar un componente o aplicación para X que opere en un contexto dado, donde X esté en uno de los dominios listados, por ejemplo, un algoritmo genético para el diseño de una planta de fábrica [Diseñar] **II:** Criticar el diseño e implementación de un componente o aplicación existente, o uno desarrollado por compañeros de clase [Críticar (análisis crítico)] **III:** Comparar el rendimiento y la eficiencia energética de múltiples implementaciones de un diseño similar, por ejemplo, multinúcleo versus clúster versus GPU [Comparar]

Bibliografía: [Pacheco and Malensek, 2021, Herlihy et al., 2020]

6. Bibliografía

[Corporation, 2024] Corporation, N. (2024). Cuda c++ programming guide. `https://docs.nvidia.com/cuda/cuda-c-programming-guide/`. Documentación oficial de NVIDIA.

[Herlihy et al., 2020] Herlihy, M., Shavit, N., Luchangco, V., and Spear, M. (2020). *The Art of Multiprocessor Programming*. Morgan Kaufmann, 2nd edition.

[Kirk and mei W. Hwu, 2016] Kirk, D. B. and mei W. Hwu, W. (2016). *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, Cambridge, MA, 3rd edition.

[Kleppmann, 2017] Kleppmann, M. (2017). *Designing Data-Intensive Applications*. O'Reilly Media, 1st edition.

[mei W. Hwu et al., 2022] mei W. Hwu, W., Kirk, D. B., and Hajj, I. E. (2022). *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 4th edition.

[Pacheco and Malensek, 2021] Pacheco, P. S. and Malensek, M. (2021). *An Introduction to Parallel Programming*. Morgan Kaufmann, 2nd edition.

[Sterling et al., 2024] Sterling, T., Brodowicz, M., and Anderson, M. (2024). *High Performance Computing: Modern Systems and Practices*. Morgan Kaufmann, 2nd edition.

[van Steen and Tanenbaum, 2023] van Steen, M. and Tanenbaum, A. S. (2023). *Distributed Systems*. Maarten van Steen, 4th edition.

[Corporation, 2024] Corporation, N. (2024). Cuda c++ programming guide. `https://docs.nvidia.com/cuda/cuda-c-programming-guide/`. Documentación oficial de NVIDIA.

[Herlihy et al., 2020] Herlihy, M., Shavit, N., Luchangco, V., and Spear, M. (2020). *The Art of Multiprocessor Programming*. Morgan Kaufmann, 2nd edition.

[Kirk and mei W. Hwu, 2016] Kirk, D. B. and mei W. Hwu, W. (2016). *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, Cambridge, MA, 3rd edition.

[Kleppmann, 2017] Kleppmann, M. (2017). *Designing Data-Intensive Applications*. O'Reilly Media, 1st edition.

[mei W. Hwu et al., 2022] mei W. Hwu, W., Kirk, D. B., and Hajj, I. E. (2022). *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 4th edition.

[Pacheco and Malensek, 2021] Pacheco, P. S. and Malensek, M. (2021). *An Introduction to Parallel Programming*. Morgan Kaufmann, 2nd edition.

[Sterling et al., 2024] Sterling, T., Brodowicz, M., and Anderson, M. (2024). *High Performance Computing: Modern Systems and Practices*. Morgan Kaufmann, 2nd edition.

[van Steen and Tanenbaum, 2023] van Steen, M. and Tanenbaum, A. S. (2023). *Distributed Systems*. Maarten van Steen, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA213 Ecuaciones Diferenciales Parciales (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA212 Ecuaciones Diferenciales Ordinarias (6to Sem-Pág 402)
  - MA211 Cálculo IV (5to Sem-Pág 384)

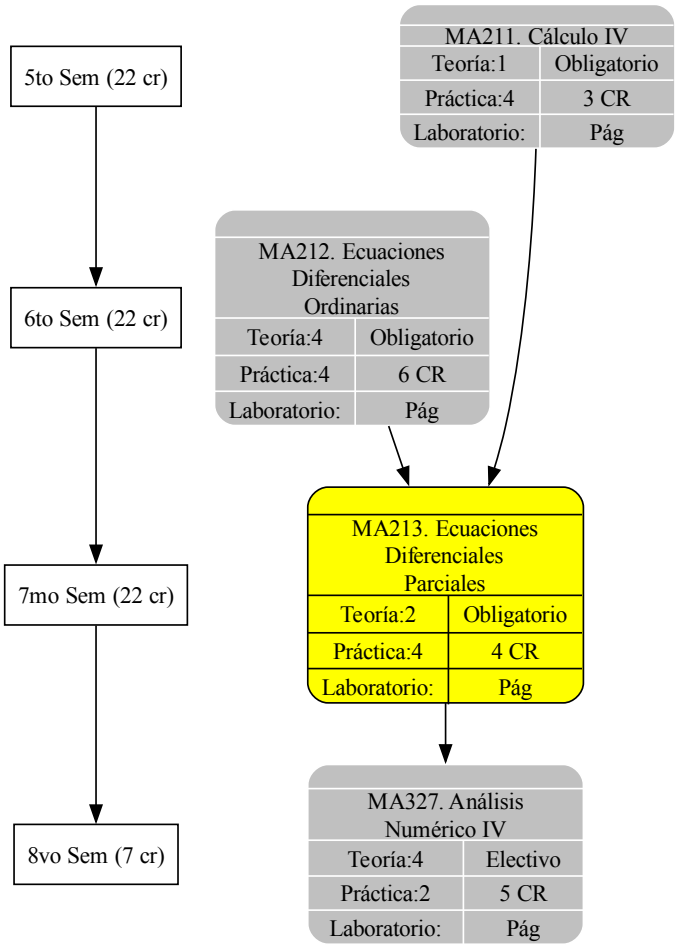


Figura 7.39: Mapa de Conexión. MA213 Ecuaciones Diferenciales Parciales

2. Fundamentación

La asignatura de Ecuaciones Diferenciales Parciales (MA213) estudia fenómenos que dependen de múltiples variables, como la propagación del calor, ondas y potenciales eléctricos. El curso introduce las series de Fourier y los métodos de separación de variables para resolver las ecuaciones clásicas de la física matemática (Calor, Onda y Laplace). Es una pieza clave para la simulación numérica avanzada y el análisis de medios continuos.

3. Objetivos del curso.

- Utilizar series de Fourier para representar funciones periódicas y resolver problemas de frontera.
- Resolver ecuaciones diferenciales parciales lineales mediante el método de separación de variables.
- Clasificar EDPs de segundo orden y comprender su comportamiento físico.
- Aplicar métodos numéricos de diferencias finitas para la resolución de EDPs.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Series de Fourier y Problemas de Sturm-Liouville, 24 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Funciones ortogonales y series de Fourier de senos y cosenos. **II:** Series de Fourier complejas e identidad de Parseval. **III:** Problemas de valores propios y el teorema de Sturm-Liouville. **IV:** Aplicación de series a la resolución de ODEs con condiciones de contorno.

Objetivos de Aprendizaje **I:** Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar] **II:** Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar] **III:** Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar] **IV:** Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

5.2 Ecuaciones de Difusión (Calor), 24 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Deducción de la ecuación de calor en una dimensión. **II:** Método de separación de variables para condiciones de Dirichlet y Neumann. **III:** Problemas de calor en barras con extremos aislados o radiación. **IV:** Ecuación de calor en dos y tres dimensiones (coordenadas cartesianas).

Objetivos de Aprendizaje **I:** Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar] **II:** Clasificar el orden de convergencia de métodos iterativos [Evaluar] **III:** Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar] **IV:** Diseñar criterios de parada apropiados para métodos iterativos [Usar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

5.3 Ecuaciones de Onda y Laplace, 24 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** La ecuación de onda unidimensional: Cuerda vibrante. **II:** Solución de D'Alembert y ondas viajeras. **III:** Ecuación de Laplace en coordenadas cartesianas y polares (Problema de Dirichlet). **IV:** Aplicaciones en estado estacionario y electrostática.

Objetivos de Aprendizaje I: Construir polinomios interpoladores usando diferentes métodos [Usar] **II:** Implementar interpolación por splines cúbicos [Usar] **III:** Aplicar el método de mínimos cuadrados para ajuste de datos [Usar] **IV:** Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse] **V:** Analizar el error de un esquema de interpolación [Evaluar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

5.4 Métodos Numéricos para EDP y Transformadas, 24 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Introducción al método de diferencias finitas para EDPs parabólicas y elípticas. **II:** Uso de la Transformada de Fourier para resolver EDPs en dominios infinitos. **III:** Estabilidad y convergencia de esquemas numéricos básicos. **IV:** Modelamiento de problemas reales mediante software de cálculo científico.

Objetivos de Aprendizaje I: Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar] **II:** Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar] **III:** Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar] **IV:** Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar] **V:** Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar]

Bibliografía: [Boyce et al., 2017, Zill, 2018]

6. Bibliografía

[Boyce et al., 2017] Boyce, W. E., DiPrima, R. C., and Meade, D. B. (2017). *Ecuaciones Diferenciales y Problemas con Valores en la Frontera*. Limusa Wiley, 11th edition.

[Zill, 2018] Zill, D. G. (2018). *Ecuaciones Diferenciales con Aplicaciones de Modelado*. Cengage Learning, 11th edition.

[Boyce et al., 2017] Boyce, W. E., DiPrima, R. C., and Meade, D. B. (2017). *Ecuaciones Diferenciales y Problemas con Valores en la Frontera*. Limusa Wiley, 11th edition.

[Zill, 2018] Zill, D. G. (2018). *Ecuaciones Diferenciales con Aplicaciones de Modelado*. Cengage Learning, 11th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA326 Análisis Numérico III (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA325 Análisis Numérico II (6to Sem-Pág 405)

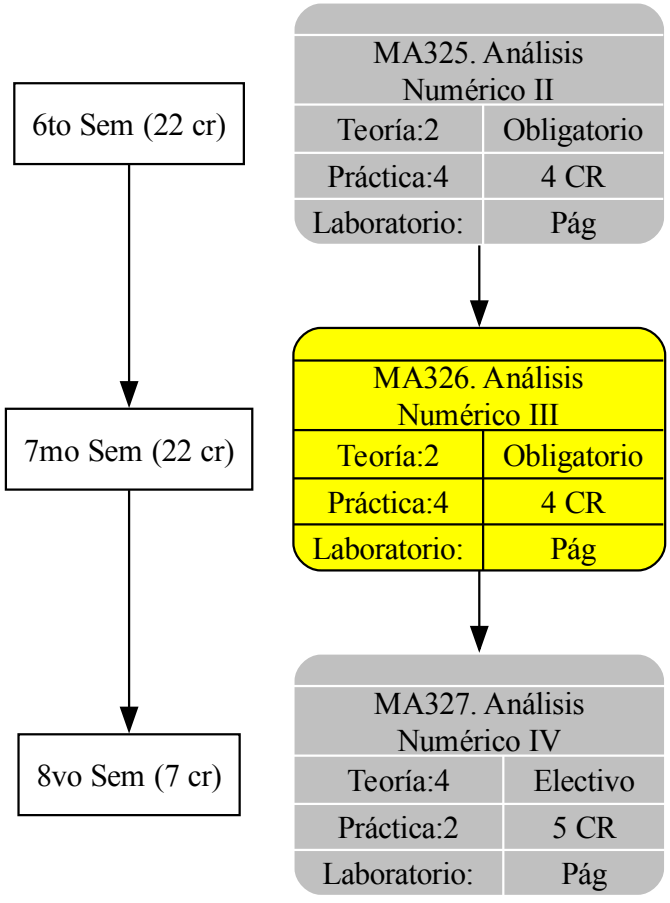


Figura 7.40: Mapa de Conexión. MA326 Análisis Numérico III

2. Fundamentación

El curso de Análisis Numérico 3 se centra en la solución computacional de Ecuaciones Diferenciales Ordinarias (EDO). En la matemática computacional, muchos sistemas no pueden resolverse analíticamente; por lo tanto, comprender la convergencia, estabilidad y eficiencia de los integradores numéricos es vital. Este curso abarca desde los métodos básicos de un paso hasta los esquemas avanzados de Runge-Kutta y el tratamiento especializado de ecuaciones rígidas (stiff), los cuales son críticos para simular reacciones químicas, circuitos eléctricos y mecánica estructural.

3. Objetivos del curso.

- Comprender la teoría fundamental de los métodos numéricos para Problemas de Valor Inicial (PVI).
- Dominar la construcción y el análisis de los métodos de Runge-Kutta de alto orden.
- Evaluar la estabilidad y convergencia de los esquemas numéricos.
- Implementar algoritmos especializados para Ecuaciones Diferenciales Rígidas.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Solución Numérica de EDOs: Fundamentos, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Repaso de Existencia y Unicidad (Continuidad de Lipschitz) **II:** Métodos elementales: Euler, Heun y métodos de series de Taylor **III:** Análisis de error: Error de truncamiento local y error global **IV:** Consistencia, Convergencia y Cero-estabilidad

Objetivos de Aprendizaje I: Formular aproximaciones discretas para sistemas diferenciales continuos [Usage] **II:** Derivar el error de truncamiento local para métodos básicos de un paso [Assessment] **III:** Analizar el impacto del tamaño del paso en la estabilidad numérica [Familiarity]

Bibliografía: [Burden et al., 2016, Heath, 2002]

5.2 Métodos de Runge-Kutta, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Estructura general de los métodos de Runge-Kutta (RK) **II:** Tablas de Butcher y condiciones de orden **III:** Métodos de Runge-Kutta explícitos e implícitos **IV:** Control adaptativo del tamaño del paso: Runge-Kutta-Fehlberg (RK45)

Objetivos de Aprendizaje I: Construir métodos RK de alto orden utilizando Tablas de Butcher [Usage] **II:** Implementar algoritmos adaptativos para optimizar el costo computacional [Assessment] **III:** Probar el orden de convergencia para esquemas RK específicos [Familiarity]

Bibliografía: [Quarteroni et al., 2007, Butcher, 2016]

5.3 Métodos Lineales de Pasos Múltiples, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Fórmulas de Adams-Bashforth y Adams-Moulton **II:** Métodos de Predictores-Correctores **III:** Teoría General de métodos lineales de pasos múltiples (Teoremas de Dahlquist) **IV:** Comparación entre métodos RK y de pasos múltiples

Objetivos de Aprendizaje I: Desarrollar códigos de pasos múltiples para integración eficiente a largo plazo [Usage] **II:** Verificar la condición de las raíces para la cero-estabilidad [Assessment] **III:** Seleccionar métodos adecuados basados en el compromiso entre memoria y precisión [Familiarity]

Bibliografía: [Burden et al., 2016, Quarteroni et al., 2007]

5.4 Ecuaciones Rígidas y Estabilidad Absoluta, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Definición y caracterización de Sistemas Rígidos (Stiff) **II:** Análisis de Estabilidad Lineal: La ecuación de prueba y Regiones de Estabilidad **III:** A-estabilidad y L-estabilidad **IV:** Fórmulas de Diferenciación Regresiva (BDF) y métodos implícitos para rigidez

Objetivos de Aprendizaje **I:** Identificar la rigidez en sistemas físicos mediante el análisis de autovalores [Assessment] **II:** Graficar e interpretar regiones de estabilidad en el plano complejo [Usage] **III:** Aplicar resolvedores implícitos para superar restricciones del tamaño de paso en problemas rígidos [Assessment]

Bibliografía: [Heath, 2002, Butcher, 2016]

6. Bibliografía

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Numerical Analysis*. Cengage Learning, 10th edition.

[Butcher, 2016] Butcher, J. C. (2016). *Numerical Methods for Ordinary Differential Equations*. Wiley, 3rd edition.

[Heath, 2002] Heath, M. T. (2002). *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2nd edition.

[Quarteroni et al., 2007] Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer, 2nd edition.

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Numerical Analysis*. Cengage Learning, 10th edition.

[Butcher, 2016] Butcher, J. C. (2016). *Numerical Methods for Ordinary Differential Equations*. Wiley, 3rd edition.

[Heath, 2002] Heath, M. T. (2002). *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2nd edition.

[Quarteroni et al., 2007] Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA361 Modelado Matemático (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - MA325 Análisis Numérico II (6to Sem-Pág 405)

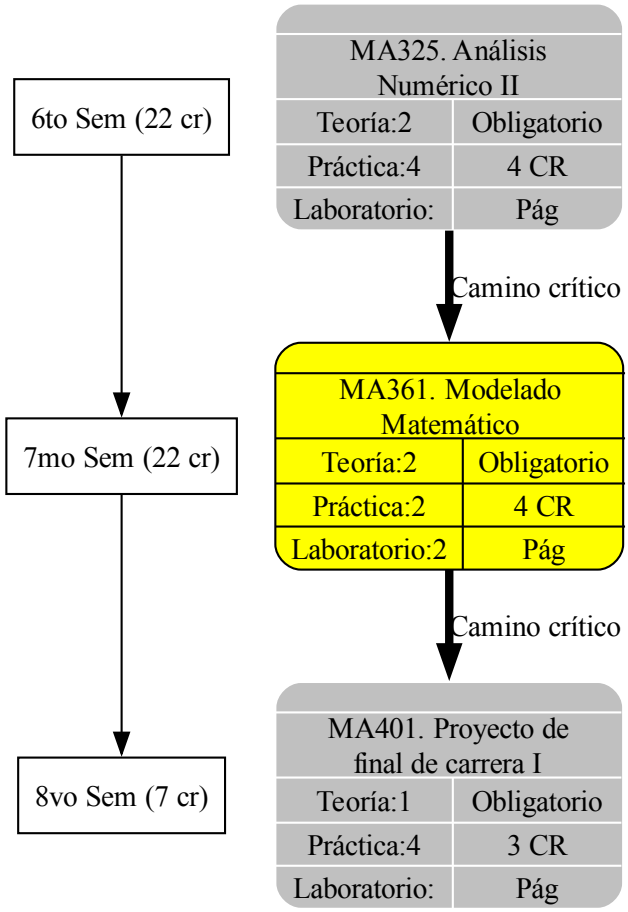


Figura 7.41: Mapa de Conexión. MA361 Modelado Matemático

2. Fundamentación

Este curso introduce los principios fundamentales del modelado matemático para representar, analizar y predecir el comportamiento de sistemas en diversas áreas como ciencias, ingeniería y negocios. Se enfatiza la formulación de modelos a partir de problemas reales, su resolución mediante herramientas matemáticas y computacionales, y la interpretación crítica de los resultados. El curso desarrolla la capacidad de abstracción y la aplicación de matemáticas avanzadas para resolver problemas complejos del mundo real.

3. Objetivos del curso.

- Comprender los principios fundamentales y las etapas del proceso de modelado matemático.
- Formular modelos matemáticos a partir de problemas concretos en diferentes contextos científicos y tecnológicos.
- Aplicar métodos analíticos y numéricos para resolver los modelos planteados.
- Interpretar críticamente los resultados de las simulaciones y validar los modelos con datos reales.
- Desarrollar habilidades para comunicar efectivamente los modelos y sus conclusiones.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Formulación de Modelos Matemáticos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Proceso y ciclo de modelado **II:** Modelos continuos vs. discretos **III:** Modelos deterministas vs. estocásticos **IV:** Identificación de supuestos y técnicas de simplificación **V:** Análisis dimensional y escalado

Objetivos de Aprendizaje **I:** Describir el ciclo completo de modelado desde la definición del problema hasta la interpretación de la solución [Familiarizarse] **II:** Clasificar modelos como continuos/discretos y deterministas/estocásticos [Evaluar] **III:** Identificar y justificar supuestos apropiados para simplificación de modelos [Evaluar] **IV:** Aplicar análisis dimensional para reducir la complejidad de modelos físicos [Usar]

Bibliografía: [Giordano et al., 2013, Lin and Segel, 2009]

5.2 Simulación de Sistemas Continuos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Modelos de Ecuaciones Diferenciales Ordinarias (EDOs) **II:** Modelos de Ecuaciones Diferenciales Parciales (EDPs) **III:** Métodos de diferencias finitas para EDPs **IV:** Selección e implementación de solucionadores **V:** Sistemas stiff y métodos adaptativos

Objetivos de Aprendizaje **I:** Formular modelos de EDOs y EDPs para sistemas físicos [Usar] **II:** Implementar métodos de diferencias finitas para EDPs simples [Usar] **III:** Seleccionar solucionadores numéricos apropiados para diferentes tipos de EDOs/EDPs [Evaluar] **IV:** Manejar sistemas stiff usando métodos numéricos apropiados [Usar]

Bibliografía: [Braun et al., 2013, Giordano et al., 2013]

5.3 Simulación de Sistemas Discretos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Teoría de colas y modelos **II:** Autómatas celulares y modelos basados en agentes **III:** Métodos de simulación Monte Carlo **IV:** Cadenas y procesos de Markov **V:** Simulación de eventos discretos

Objetivos de Aprendizaje I: Formular y analizar modelos simples de colas [Usar] **II:** Implementar autómatas celulares y simulaciones basadas en agentes [Usar] **III:** Aplicar métodos Monte Carlo para simulación probabilística [Usar] **IV:** Analizar cadenas de Markov para transiciones de estado y comportamiento en estado estacionario [Evaluar] **V:** Diseñar simulaciones de eventos discretos para análisis de sistemas [Usar]

Bibliografía: [Giordano et al., 2013, Meerschaert, 2013]

5.4 Análisis de Sensibilidad y Robustez, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Análisis de sensibilidad local (derivadas parciales) **II:** Análisis de sensibilidad global (índices de Sobol) **III:** Método de Morris y técnicas de screening **IV:** Análisis de robustez y escenarios del peor caso **V:** Propagación de incertidumbre

Objetivos de Aprendizaje I: Realizar análisis de sensibilidad local usando métodos basados en derivadas [Usar] **II:** Aplicar técnicas de análisis de sensibilidad global (ej., índices de Sobol) [Usar] **III:** Usar el método de Morris para screening de factores [Usar] **IV:** Evaluar la robustez del modelo bajo variaciones de parámetros [Evaluar] **V:** Cuantificar la propagación de incertidumbre a través de modelos [Evaluar]

Bibliografía: [Banks et al., 2010, Giordano et al., 2013]

5.5 Validación y Verificación de Modelos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Verificación: corrección del código numérico **II:** Validación: comparación con datos experimentales **III:** Cuantificación de Incertidumbre (UQ) en modelos **IV:** Calibración de modelos y estimación de parámetros **V:** Pruebas y métricas de bondad de ajuste

Objetivos de Aprendizaje I: Distinguir entre verificación y validación en modelado [Familiarizarse] **II:** Aplicar técnicas de verificación para garantizar la corrección del código [Usar] **III:** Validar modelos usando datos experimentales y pruebas estadísticas [Usar] **IV:** Calibrar parámetros del modelo usando técnicas de optimización [Usar] **V:** Aplicar métodos de cuantificación de incertidumbre para evaluar la confiabilidad del modelo [Usar]

Bibliografía: [Giordano et al., 2013, Lin and Segel, 2009, Meerschaert, 2013]

6. Bibliografía

[Banks et al., 2010] Banks, J., II, J. S. C., Nelson, B. L., and Nicol, D. M. (2010). *Discrete-Event System Simulation*. Prentice Hall, 5th edition.

[Braun et al., 2013] Braun, M., Coleman, C. S., and Drew, D. A. (2013). *Differential Equation Models*. Springer, 1st edition.

[Giordano et al., 2013] Giordano, F. R., Fox, W. P., and Horton, S. B. (2013). *A First Course in Mathematical Modeling*. Cengage Learning, 5th edition.

[Lin and Segel, 2009] Lin, C. C. and Segel, L. A. (2009). *Mathematics Applied to Deterministic Problems in the Natural Sciences*. Society for Industrial and Applied Mathematics, 1st edition.

[Meerschaert, 2013] Meerschaert, M. M. (2013). *Mathematical Modeling*. Academic Press, 4th edition.

[Banks et al., 2010] Banks, J., II, J. S. C., Nelson, B. L., and Nicol, D. M. (2010). *Discrete-Event System Simulation*. Prentice Hall, 5th edition.

[Braun et al., 2013] Braun, M., Coleman, C. S., and Drew, D. A. (2013). *Differential Equation Models*. Springer, 1st edition.

[Giordano et al., 2013] Giordano, F. R., Fox, W. P., and Horton, S. B. (2013). *A First Course in Mathematical Modeling*. Cengage Learning, 5th edition.

[Lin and Segel, 2009] Lin, C. C. and Segel, L. A. (2009). *Mathematics Applied to Deterministic Problems in the Natural Sciences*. Society for Industrial and Applied Mathematics, 1st edition.

[Meerschaert, 2013] Meerschaert, M. M. (2013). *Mathematical Modeling*. Academic Press, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

ID203 Inglés V (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 6 horas;
- **Prerrequisitos**:
  - ID202 Inglés IV (6to Sem-Pág 417)

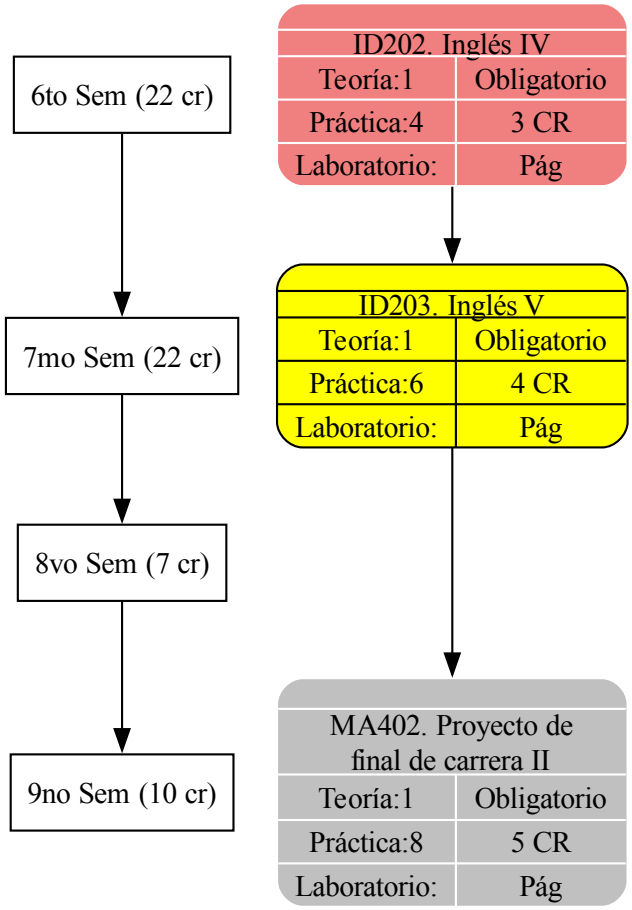


Figura 7.42: Mapa de Conexión. ID203 Inglés V

2. Fundamentación

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

3. Objetivos del curso.

- Incrementar la capacidad y la fluidez de hablar y entender el idioma Inglés.
- Hacer que los alumnos interactuen con mayor énfasis en la creación de diálogos.

4. Competencias Generales (*Outcomes*)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Familiarizarse)

5. Contenido

5.1 It's a wonderful world, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Verbos Auxiliares **II:** Tiempos Verbales **III:** Preguntas y oraciones negativas **IV:** Respuestas cortas **V:** Formación de palabras **VI:** Expresiones coloquiales **VII:** rrección de errores

Objetivos de Aprendizaje I: Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los auxiliares y de los diferentes tipos de oración es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de formación de palabras.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

5.2 Happiness!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Presente Simple **II:** Presente Continuo **III:** Voz Pasiva en Presente **IV:** Verbos para deportes y tiempo libre **V:** Tipos de números y fechas **VI:** Inventos/Mundo Moderno **VII:** Corrección de errores

Objetivos de Aprendizaje I: Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar deportes y actividades de tiempo libre. Utiliza todo tipo de expresiones numéricas. Expresar situaciones y estados relacionados con formas de presente. Explica y aplica vocabulario de actividades al aire libre.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

5.3 Telling tales!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: I: Tiempo Pasado Simple **II:** Pasado Continuo **III:** Voz Pasiva en Pasado **IV:** Vocabulario de Arte y Literatura **V:** Expresiones para dar y pedir opiniones **VI:** Cuentos e historias

Objetivos de Aprendizaje I: Al terminar la tercera unidad, los alumnos habiendo reconocido las características de las formas de pasado pasivo, utilizan éstos para hacer descripciones de diversos tipos. Describen arte y literatura y dan indicaciones de opinión. Utilizarán conjunciones para unir ideas tipo.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

5.4 Doing the right thing!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Verbos Auxiliares de Modo I **II:** Oraciones Afirmativas, Negativas y Preguntas con Modals **III:** Uso de nacionalidades i otros adjetivos **IV:** Expresiones de pedidos y Ofrecimientos **V:** Guía de los Buenos Modales **VI:** Llenado de Formatos **VII:** Símbolos fonéticos

Objetivos de Aprendizaje **I:** Al terminar la cuarta unidad, los alumnos habiendo identificado la idea de expresar ideas de modos de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente. Expresan ideas de nacionalidades y hacen pedidos y ofrecimientos variados.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

5.5 On the move!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Futuro con Hill **II:** Oraciones en Tiempo Futuro con going to **III:** Uso de might para futuro **IV:** Expresiones del clima **V:** Vocabulario del clima **VI:** Expresiones para hoteles y transporte **VII:** E-mails

Objetivos de Aprendizaje **I:** Al finalizar la quinta unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar ideas del clima. Adquirirán vocabulario para describir uso de transporte público. Se presentará expresiones para hacer pedidos en hoteles.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

5.6 I just love it!, 0 hr(s)

Competencias: ABET-4,ABET-6

Tópicos: **I:** Preguntas con formas Like **II:** Patrones Verbales II **III:** Vocabulario de Comida, Lugares y ocupaciones **IV:** Palabras que van unidas en contexto **V:** Expresiones para vistas y sonidos **VI:** Composición de Impresiones personales

Objetivos de Aprendizaje **I:** Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas con like y con patrones verbales trabajos aplicados a contextos adecuados. Enfatizan la diferencia entre comidas, lugares y personas. Describen vistas y sonidos. Utilizan expresiones comparar la vida diaria en diferentes lugares. Asumen la idea de estilos de vida diferentes.

Bibliografía: [Soars and John, 2002g, Soars and John, 2002i, Soars and John, 2002h, Cambridge, 2006, MacGrew, 1999]

6. Bibliografía

[Cambridge, 2006] Cambridge (2006). *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford.

[MacGrew, 1999] MacGrew, J. (1999). *Focus on Grammar Basic*. Editorial Oxford.

[Soars and John, 2002g] Soars, L. and John (2002g). *American Headway N 3 Student Book*. Editorial Oxford.

[Soars and John, 2002h] Soars, L. and John (2002h). *American Headway N 3 Teachers Book*. Editorial Oxford.

[Soars and John, 2002i] Soars, L. and John (2002i). *American Headway N 3 Work Book*. Editorial Oxford.

Octavo Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI263 Introducción al Aprendizaje de Máquina (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS261 Inteligencia Artificial (7mo Sem-Pág 421)

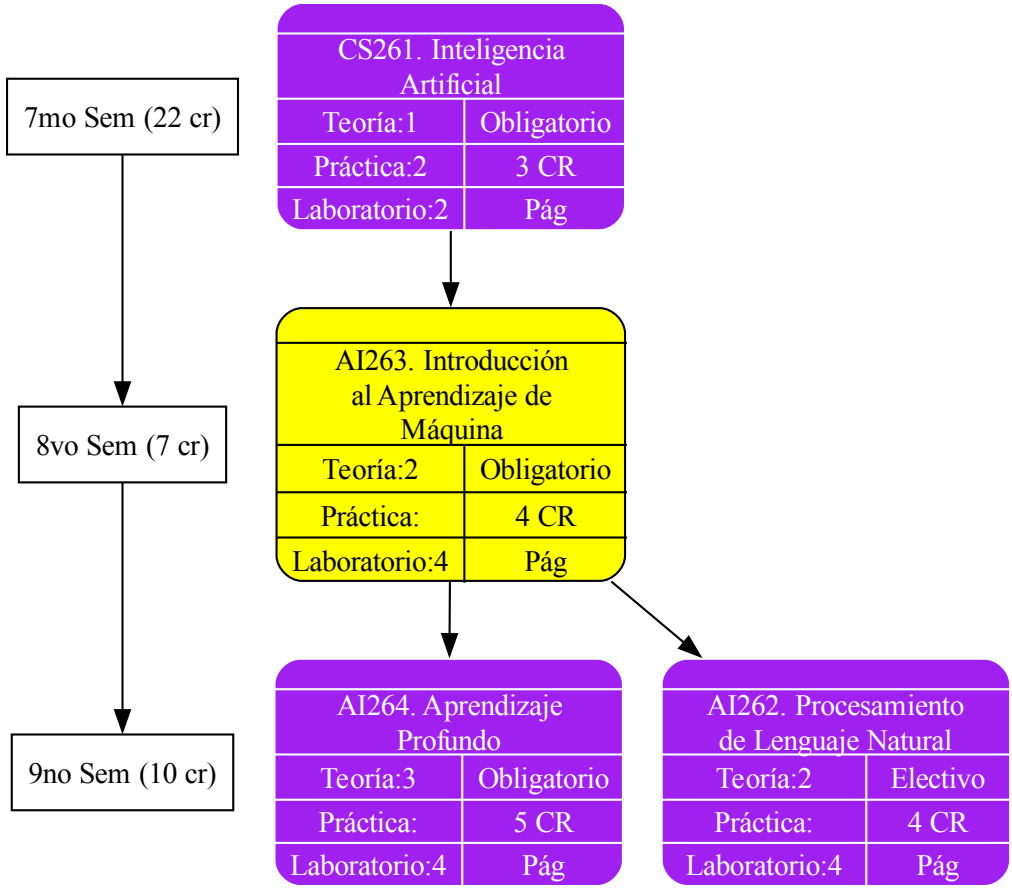


Figura 7.43: Mapa de Conexión. AI263 Introducción al Aprendizaje de Máquina

2. Fundamentación

Este curso introduce los fundamentos del aprendizaje automático, cubriendo algoritmos clásicos y modernos para problemas de clasificación, regresión y agrupamiento. Se enfoca en la implementación práctica usando scikit-learn y TensorFlow, con aplicaciones en visión por computadora y procesamiento de lenguaje natural.

3. Objetivos del curso.

- Comprender los principios matemáticos detrás de algoritmos de ML.
- Implementar pipelines completos de ML con Python.
- Evaluar y optimizar modelos usando métricas estándar.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Fundamentos de ML, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Aprendizaje supervisado vs no supervisado **II:** Overfitting y regularización **III:** Validación cruzada y curvas de aprendizaje

Objetivos de Aprendizaje I: Explicar el bias-variance tradeoff [Familiarizarse] **II:** Implementar k-fold cross-validation [Usar]

Bibliografía: [Bishop, 2006, Goodfellow et al., 2016]

5.2 Modelos Lineales, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Regresión lineal y logística **II:** Máquinas de vectores de soporte (SVM) **III:** Análisis discriminante lineal

Objetivos de Aprendizaje I: Programar modelos lineales con scikit-learn [Usar] **II:** Interpretar coeficientes de regresión [Evaluar]

Bibliografía: [Hastie et al., 2009, Murphy, 2012]

5.3 Ensambles y Árboles, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Árboles de decisión y random forests **II:** Gradient Boosting (XGBoost, LightGBM) **III:** Stacking y votación

Objetivos de Aprendizaje I: Optimizar hiperparámetros con GridSearch [Usar] **II:** Visualizar árboles de decisión [Evaluar]

Bibliografía: [Hastie et al., 2009, Géron, 2022]

5.4 Redes Neuronales Básicas, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Perceptrones multicapa (MLP) **II:** Backpropagation y optimizadores **III:** Introducción a Keras/TensorFlow

Objetivos de Aprendizaje I: Construir redes neuronales simples [Usar] **II:** Monitorizar entrenamiento con TensorBoard [Evaluar]

Bibliografía: [Goodfellow et al., 2016, Chollet, 2021]

6. Bibliografía

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Chollet, 2021] Chollet, F. (2021). *Deep Learning with Python*. Manning, 2nd edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Géron, 2022] Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 3rd edition.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, 2nd edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[Chollet, 2021] Chollet, F. (2021). *Deep Learning with Python*. Manning, 2nd edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Géron, 2022] Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 3rd edition.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, 2nd edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA401 Proyecto de final de carrera I (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA361 Modelado Matemático (7mo Sem-Pág 444)
  - MA371 Análisis funcional I (6to Sem-Pág 414)
  - 140cr

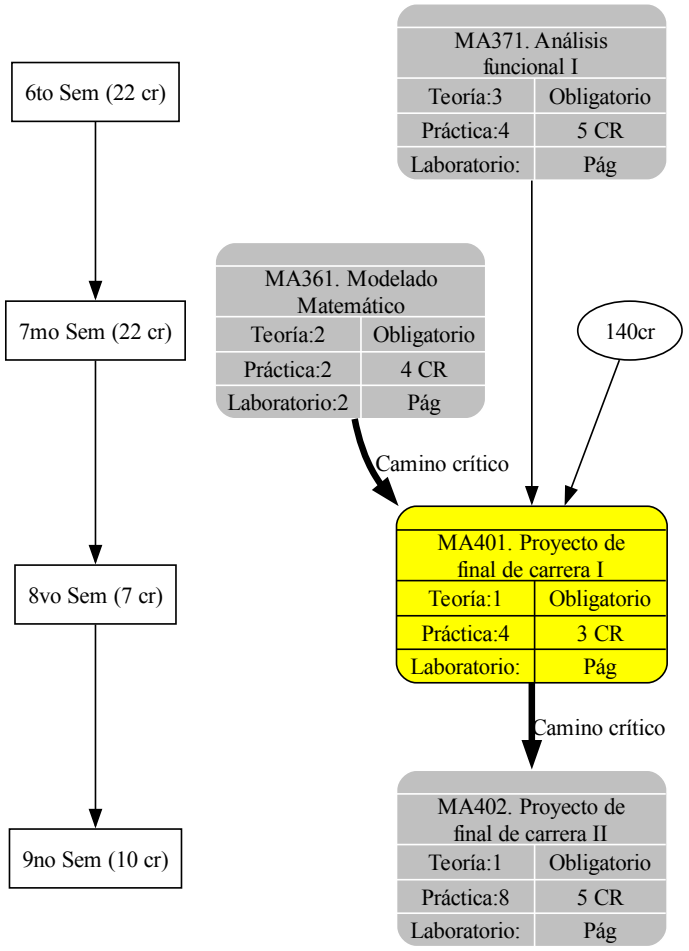


Figura 7.44: Mapa de Conexión. MA401 Proyecto de final de carrera I

2. Fundamentación

Este curso representa el primer semestre del proyecto de final de carrera en matemáticas computacionales. Los estudiantes desarrollan un proyecto de investigación aplicada o desarrollo tecnológico que integra y aplica los conocimientos adquiridos durante la carrera. El curso enfatiza la formulación de

problemas, revisión de literatura, diseño metodológico, y desarrollo inicial de soluciones computacionales bajo supervisión de un profesor tutor. Los proyectos deben demostrar capacidad de investigación, innovación y aplicación de matemáticas computacionales a problemas del mundo real.

3. Objetivos del curso.

- Formular y delimitar un problema de investigación en matemáticas computacionales con relevancia teórica o aplicada.
- Realizar una revisión sistemática de la literatura y estado del arte relacionado con el problema de investigación.
- Diseñar una metodología apropiada para abordar el problema, seleccionando técnicas y herramientas computacionales adecuadas.
- Desarrollar las fases iniciales del proyecto, incluyendo prototipos y análisis preliminares.
- Comunicar efectivamente los avances del proyecto mediante informes técnicos y presentaciones orales.
- Demostrar capacidad de trabajo autónomo y gestión de proyecto bajo supervisión tutorial.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Evaluar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Evaluar)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Evaluar)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Evaluar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Evaluar)

5. Contenido

5.1 Formulación y Planificación del Proyecto, 12 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Identificación y delimitación de problemas de investigación **II:** Revisión sistemática de literatura y estado del arte **III:** Formulación de objetivos e hipótesis de investigación **IV:** Diseño metodológico y plan de trabajo **V:** Elaboración de propuesta de proyecto y cronograma

Objetivos de Aprendizaje **I:** Formular problemas de investigación con impacto social [Evaluar] **II:** Diseñar planes de gestión para proyectos científicos [Usar] **III:** Realizar estudios de problemas usando métodos de investigación [Usar] **IV:** Evaluar la viabilidad técnica y científica de proyectos [Evaluar]

Bibliografía: [Babbie, 2016, Kothari, 2004]

5.2 Desarrollo Metodológico y Herramientas, 12 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Selección y justificación de métodos computacionales **II:** Diseño de experimentos y protocolos de validación **III:** Herramientas computacionales para desarrollo de proyectos **IV:** Gestión de versiones y documentación técnica **V:** Consideraciones éticas en investigación computacional

Objetivos de Aprendizaje **I:** Aplicar principios éticos en investigación profesional [Usar] **II:** Trabajar efectivamente en equipos multidisciplinarios [Usar] **III:** Comunicar avances técnicos de forma efectiva [Usar] **IV:** Seleccionar herramientas apropiadas para el proyecto [Evaluar]

Bibliografía: [Oates, 2006, Pressman, 2014]

5.3 Implementación y Desarrollo Inicial, 12 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Desarrollo de prototipos y pruebas conceptuales **II:** Implementación de algoritmos y métodos computacionales **III:** Análisis preliminar de resultados y validación **IV:** Iteración y refinamiento de soluciones **V:** Documentación técnica del desarrollo

Objetivos de Aprendizaje **I:** Aprender de forma independiente y continua [Usar] **II:** Conducir estudios usando métodos de investigación [Usar] **III:** Implementar procesos matemáticos en computadora [Usar] **IV:** Adaptar soluciones basadas en resultados preliminares [Evaluar]

Bibliografía: [Sommerville, 2015, McConnell, 2004]

6. Bibliografía

[Babbie, 2016] Babbie, E. R. (2016). *The Basics of Social Research*. Cengage Learning, 7th edition.

[Kothari, 2004] Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International, 2nd edition.

[McConnell, 2004] McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

[Oates, 2006] Oates, B. J. (2006). *Researching Information Systems and Computing*. SAGE Publications, 1st edition.

[Pressman, 2014] Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 8th edition.

[Sommerville, 2015] Sommerville, I. (2015). *Software Engineering*. Pearson, 10th edition.

[Babbie, 2016] Babbie, E. R. (2016). *The Basics of Social Research*. Cengage Learning, 7th edition.

[Kothari, 2004] Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International, 2nd edition.

[McConnell, 2004] McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

[Oates, 2006] Oates, B. J. (2006). *Researching Information Systems and Computing*. SAGE Publications, 1st edition.

[Pressman, 2014] Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 8th edition.

[Sommerville, 2015] Sommerville, I. (2015). *Software Engineering*. Pearson, 10th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS2S1 Sistemas Operativos (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 361)

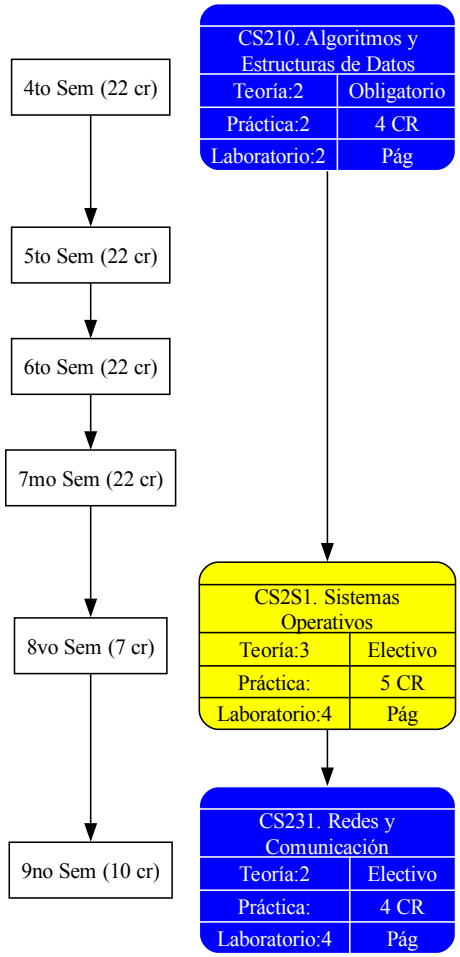


Figura 7.45: Mapa de Conexión. CS2S1 Sistemas Operativos

2. Fundamentación

El sistema operativo es una colección de servicios necesarios para interconectar de manera segura el hardware con las aplicaciones. Este curso explora los mecanismos y políticas necesarios para virtualizar el cómputo, la memoria y las E/S. Los estudiantes comprenderán conceptos centrales como concurrencia, persistencia y límites de confianza, los cuales son fundamentales para desarrollar sistemas de software robustos y eficientes.

3. Objetivos del curso.

- Comprender el rol y propósito de los sistemas operativos modernos.
- Dominar la planificación de procesos y el ciclo de vida de las unidades de ejecución.
- Analizar la gestión de memoria, incluyendo la virtualización de memoria física y virtual.
- Evaluar sistemas de almacenamiento y técnicas de gestión de archivos.
- Comprender los conceptos básicos de seguridad y protección dentro de un entorno de sistema operativo.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Usar)

5. Contenido

5.1 Rol y Propósito de los Sistemas Operativos, 8 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación. **II:** Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos) **III:** Funciones extendidas y/o especializadas del sistema operativo (por ejemplo, sistemas embebidos, tipos de servidor como archivos, web, multimedia, gestores de arranque y seguridad de arranque) **IV:** Problemas de diseño (por ejemplo, eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad, energía, seguridad, compensaciones entre verificación de errores y rendimiento, flexibilidad y rendimiento, y seguridad y rendimiento). Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad **V:** Influencias de seguridad, redes, multimedia, computación paralela y distribuida **VI:** Preocupación general de seguridad/protección: Omitir considerar la seguridad en cada capa crea una oportunidad para acceder de manera inapropiada a los recursos. *a*) Se puede lograr acceso no autorizado a archivos en una unidad no cifrada moviendo el medio a otra computadora. *b*) La seguridad impuesta por los sistemas operativos puede ser derrotada infiltrando la capa de arranque antes de que se cargue el sistema operativo. *c*) El aislamiento de procesos puede ser subvertido por una verificación de autorización inadecuada en los límites de la API. *d*) Las vulnerabilidades en el firmware del sistema pueden proporcionar vectores de ataque que eviten por completo el sistema operativo. *e*) El aislamiento inadecuado de la memoria, cómputo y hardware de la máquina virtual puede exponer el sistema anfitrión a ataques desde sistemas invitados. *f*) El sistema operativo puede necesitar mitigar la explotación de vulnerabilidades de hardware y firmware, lo que lleva a posibles reducciones de rendimiento (por ejemplo, mitigaciones de Spectre y Meltdown). **VII:** Exposición de funciones del sistema operativo en shells y programación de sistemas. Ver también: Scripting de Shell

Objetivos de Aprendizaje **I:** Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar] **II:** Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar] **III:** Comprender las funciones de un sistema operativo contemporáneo con respecto a conveniencia, eficiencia y capacidad de evolucionar [Explicar] **IV:** Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]

Bibliografía: [Silberschatz et al., 2018, Tanenbaum and Bos, 2014]

5.2 Scripting de Shell, 8 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Manejo de errores/excepciones **II:** Tuberías (piping) **III:** Comandos del sistema: *a*) Interfaz con sistemas operativos **IV:** Variables de entorno **V:** Abstracción de archivo y operadores **VI:** Estructuras de datos, como arreglos y listas. **VII:** Expresiones regulares **VIII:** Programas y procesos **IX:** Flujo de trabajo

Objetivos de Aprendizaje **I:** Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Crear] **II:** Resolver varios problemas de procesamiento de texto mediante scripting [Resolver]

Bibliografía: [Robbins and Beebe, 2005, Silberschatz et al., 2018]

5.3 Planificación, 16 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Planificación con expropiación y sin expropiación **II:** Planificadores y políticas (por ejemplo, primero en llegar primero en ser servido, trabajo más corto primero, prioridad, round robin, multinivel). Ver también: Gestión de Recursos **III:** Conceptos de planificación de multiprocesador simétrico (SMP) y coherencia de caché Ver también: Jerarquía de Memoria **IV:** Temporizadores (por ejemplo, construir muchos temporizadores a partir de temporizadores de hardware finitos). Ver también: Organización de Máquina a Nivel de Ensamblador **V:** Justicia e inanición **VI:** Subtemas de sistemas operativos como planificación consciente de energía y planificación en tiempo real. Ver también: Rendimiento y Eficiencia Energética, Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores, Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo **VII:** Planificación cooperativa, como futexes de Linux y planificación en espacio de usuario.

Objetivos de Aprendizaje **I:** Comparar y contrastar los algoritmos comunes utilizados para planificación con y sin expropiación de tareas en sistemas operativos, como prioridad, comparación de rendimiento y esquemas de participación justa [Comparar] **II:** Explicar las relaciones entre algoritmos de planificación y dominios de aplicación [Explicar] **III:** Explicar las distinciones entre tipos de planificadores de procesador como corto plazo, mediano plazo, largo plazo y E/S [Explicar] **IV:** Evaluar un problema o solución para determinar la idoneidad para multiprocesamiento asimétrico y/o simétrico [Evaluar] **V:** Evaluar un problema o solución para determinar la idoneidad como proceso vs hilos [Evaluar] **VI:** Listar algunos contextos que se benefician de la planificación con expropiación y con plazo límite [Listar/Enumerar] **VII:** Explicar las formas en que la lógica incorporada en los algoritmos de planificación es aplicable a otros mecanismos de sistemas operativos, como primero en llegar primero en ser servido o prioridad para E/S de disco, planificación de red, planificación de proyectos y problemas más allá de la computación [Explicar]

Bibliografía: [Silberschatz et al., 2018, Stallings, 2017]

5.4 Gestión de Memoria, 16 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas **II:** Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento **III:** Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas **IV:** Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas

y marcos **V:** Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad *a*) Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes) **VI:** Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento) **VII:** Mecanismos y conceptos de seguridad en gestión de memoria incluyendo sandboxing, protección, aislamiento y vectores relevantes de ataque. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad **VIII:** Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia

Objetivos de Aprendizaje **I:** Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar] **II:** Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir] **III:** Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar] **IV:** Describir la razón y el uso de la memoria caché (rendimiento y proximidad, cómo las cachés complican el aislamiento y la abstracción de máquina virtual) [Describir] **V:** Codificar/Desarrollar programas eficientes que consideren los efectos del reemplazo de páginas y asignación de marcos en el rendimiento de un proceso y el sistema en el que se ejecuta [Crear] **VI:** Explicar cómo se utiliza el hardware para una virtualización eficiente [Explicar]

Bibliografía: [Silberschatz et al., 2018, Arpaci-Dusseau and Arpaci-Dusseau, 2018]

5.5 API de Sistemas de Archivos e Implementación, 12 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Concepto de archivo incluyendo datos, metadatos, operaciones y modo de acceso **II:** Montaje de sistema de archivos **III:** Control de acceso a archivos **IV:** Compartición de archivos **V:** Métodos básicos de asignación de archivos, incluida la tabla de asignación enlazada **VI:** Estructuras del sistema de archivos que comprenden asignación de archivos incluyendo varias estructuras de directorio y métodos para identificar archivos de manera única (por ejemplo, nombre, identificador o ubicación de almacenamiento de metadatos) **VII:** Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) impacto en rendimiento y flexibilidad (es decir, fragmentación interna y externa y compactación) **VIII:** Gestión de espacio libre como usar tablas de bits vs enlazado **IX:** Implementación de directorios para segmentar y rastrear ubicación de archivos

Objetivos de Aprendizaje **I:** Explicar las elecciones que deben tomarse en el diseño de sistemas de archivos [Explicar] **II:** Evaluar diferentes enfoques para organización de archivos, reconociendo las fortalezas y debilidades de cada uno [Evaluar] **III:** Aplicar constructores de software apropiadamente dado el conocimiento de la implementación del sistema de archivos [Aplicar]

Bibliografía: [Silberschatz et al., 2018, Stallings, 2017]

5.6 Protección y Seguridad, 12 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Descripción general de mecanismos de seguridad del sistema operativo. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad **II:** Ataques y antagonismo (planificación, etc.). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad **III:** Revisión de vulnerabilidades principales en sistemas operativos reales. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad **IV:** Estrategias de mitigación del sistema operativo como copias de seguridad. Ver también: Confiabilidad del Sistema **V:** Separación política/mecanismo. Ver también: Gestión de Riesgos y Datos, Gobierno, Gestión y Política de Seguridad **VI:** Métodos y dispositivos de seguridad. Ver también: Seguridad en IA

y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad *a*) Anillos de protección (historia desde Multics hasta x86 virtualizado) *b*) Anillos -1 y -2 de x86_64 (hipervisor y ME/PSP) **VII:** Protección, control de acceso y autenticación. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Protocolos, Aplicaciones y Criptografía Avanzada, Algoritmos Criptográficos, Fundamentos de Criptografía

Objetivos de Aprendizaje I: Comprender el requisito de mecanismos de protección y seguridad en sistemas operativos [Explicar] **II:** Listar y describir los vectores de ataque que aprovechan vulnerabilidades del SO [Listar/Enumerar] **III:** Comprender los mecanismos disponibles en un SO para controlar el acceso a recursos [Explicar] **IV:** Resumir las características y limitaciones de un sistema operativo que impactan la protección y seguridad [Resumir]

Bibliografía: [Silberschatz et al., 2018, Tanenbaum and Bos, 2014]

6. Bibliografía

[Arpaci-Dusseau and Arpaci-Dusseau, 2018] Arpaci-Dusseau, R. H. and Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, 1.0 edition.

[Robbins and Beebe, 2005] Robbins, A. and Beebe, N. H. (2005). *Classic Shell Scripting*. O'Reilly Media.

[Silberschatz et al., 2018] Silberschatz, A., Galvin, P. B., and Gagne, G. (2018). *Operating System Concepts*. Wiley, 10th edition.

[Stallings, 2017] Stallings, W. (2017). *Operating Systems: Internals and Design Principles*. Pearson, 9th edition.

[Tanenbaum and Bos, 2014] Tanenbaum, A. S. and Bos, H. (2014). *Modern Operating Systems*. Pearson, 4th edition.

[Arpaci-Dusseau and Arpaci-Dusseau, 2018] Arpaci-Dusseau, R. H. and Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, 1.0 edition.

[Robbins and Beebe, 2005] Robbins, A. and Beebe, N. H. (2005). *Classic Shell Scripting*. O'Reilly Media.

[Silberschatz et al., 2018] Silberschatz, A., Galvin, P. B., and Gagne, G. (2018). *Operating System Concepts*. Wiley, 10th edition.

[Stallings, 2017] Stallings, W. (2017). *Operating Systems: Internals and Design Principles*. Pearson, 9th edition.

[Tanenbaum and Bos, 2014] Tanenbaum, A. S. and Bos, H. (2014). *Modern Operating Systems*. Pearson, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS3I1 Seguridad en Computación (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 361)

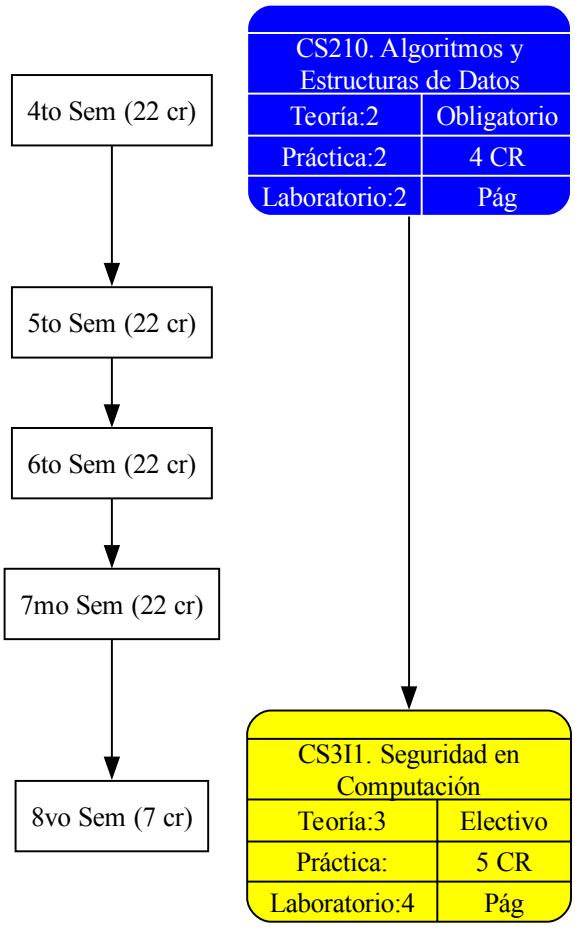


Figura 7.46: Mapa de Conexión. CS3I1 Seguridad en Computación

2. Fundamentación

En un mundo cada vez más digital, la seguridad ya no es una característica opcional sino un requisito fundamental. Este curso proporciona una introducción completa a los principios y prácticas de la ciberseguridad, cubriendo desde los fundamentos teóricos hasta la codificación segura y la informática forense digital. Los estudiantes adquirirán las habilidades necesarias para diseñar, implementar y gobiernar sistemas seguros, comprendiendo las responsabilidades éticas y profesionales involucradas en la protección de la información y la infraestructura.

3. Objetivos del curso.

- Comprender los principios fundamentales y las bases matemáticas de la seguridad.
- Aplicar técnicas de codificación segura y métodos criptográficos al software.
- Analizar vulnerabilidades del sistema y realizar investigaciones forenses.
- Evaluar la gobernanza de seguridad y la ética profesional en ciberseguridad.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Principios y Mentalidad de Seguridad, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Desarrollar una mentalidad de seguridad que incorpore conceptos transversales: confidencialidad, integridad, disponibilidad, evaluación de riesgos, pensamiento sistémico, pensamiento adversarial, pensamiento centrado en el humano. **II:** Vulnerabilidades, amenazas, superficies de ataque y vectores de ataque Protección y Seguridad. **III:** Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS) Protección y Seguridad. **IV:** Principios y prácticas de protección, por ejemplo, privilegio mínimo, diseño abierto, valores predeterminados seguros ante fallos, defensa en profundidad y confianza cero; y cómo pueden implementarse Principios del Sistema Operativo, Protección y Seguridad, Prácticas de Codificación, Construcción a Escala y Proceso, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **V:** Consideraciones de optimización entre seguridad, privacidad, rendimiento y otros objetivos de diseño Diseño de Sistemas, Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **VI:** Conceptos de confianza (trust) y confiabilidad (trustworthiness).

Objetivos de Aprendizaje **I:** Evaluar un sistema para posibles ataques que pueda lanzar un adversario [Evaluar] **II:** Diseñar y desarrollar enfoques para proteger un sistema de un conjunto de amenazas identificadas [Diseñar] **III:** Describir cómo se puede evitar el daño a la privacidad del usuario [Describir] **IV:** Desarrollar un sistema que incorpore varios principios de seguridad y privacidad [Crear]

Bibliografía: [Pfleeger et al., 2015, Stallings, 2017]

5.2 Fundamentos de Criptografía, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Diferencias entre las vistas algorítmica, aplicada y matemática de la criptografía. **II:** Preliminares matemáticos: aritmética modular, algoritmo euclidiano, independencia probabilística, conceptos básicos de álgebra lineal, teoría de números, campos finitos, complejidad, análisis asintótico Matemáticas Discretas, álgebra Lineal. **III:** Criptografía básica: criptografía de clave simétrica y de clave pública Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, Matemáticas Discretas. **IV:** Criptosistemas clásicos, como cifrados de desplazamiento, sustitución, transposición, libros de códigos y máquinas Matemáticas Discretas. **V:** Principio de Kerckhoff y uso de bibliotecas evaluadas (vetted) Prácticas de Codificación, Construcción a Escala y Proceso.

Objetivos de Aprendizaje I: Explicar el papel de la criptografía en el soporte de la seguridad y la privacidad [Explicar] **II:** Discutir los riesgos de inventar los propios métodos criptográficos [Debatir] **III:** Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Debatir] **IV:** Implementar y criptoanalizar cifrados clásicos [Implementar]

Bibliografía: [Stallings, 2017]

5.3 Inyección y Validación de Entrada, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Vulnerabilidades y debilidades comunes. **II:** Inyección SQL y otros ataques de inyección. **III:** Técnicas de cross-site scripting y mitigaciones. **IV:** Validación de entrada y sanitización de datos Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.

Objetivos de Aprendizaje I: Identificar problemas subyacentes en ejemplos dados de una enumeración de debilidades comunes y explicar cómo pueden evitarse [Analizar] **II:** Aplicar técnicas de validación de entrada y sanitización de datos para mejorar la seguridad de un programa [Aplicar] **III:** Evaluar y prevenir ataques de inyección SQL en una aplicación de base de datos [Evaluar] **IV:** Evaluar y prevenir ataques de cross-site scripting contra un sitio web [Evaluar] **V:** Construir un programa que pruebe todos los errores de manejo de entrada [Crear]

Bibliografía: [Pfleeger et al., 2015]

5.4 Análisis de Malware y Seguridad Avanzada, 6 hr(s)

Competencias: ABET-2

Tópicos: I: Técnicas de seguridad ofensiva como defensa. **II:** Técnicas de detección de malware asistidas por IA. **III:** Ransomware: creación, prevención y mitigación. **IV:** Uso seguro de componentes de terceros Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **V:** Malware: variedades, creación, ingeniería inversa y defensa contra ellos Traducción y Ejecución de Lenguajes, Modelo de Ejecución y Memoria de Sistemas. **VI:** Garantía (assurance): pruebas (incluyendo fuzzing y pruebas de penetración), verificación y validación Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **VII:** Análisis estático y dinámico Análisis de Programas y Analizadores, Protección y Seguridad, Evaluación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **VIII:** Compiladores seguros y generación de código seguro Traducción y Ejecución de Lenguajes, Comportamiento en Tiempo de Ejecución y Sistemas.

Objetivos de Aprendizaje I: Describir diferentes tipos de software malicioso [Describir] **II:** Explicar los riesgos del mal uso de interfaces con código de terceros y cómo usar correctamente el código de terceros [Explicar] **III:** Discutir la necesidad de actualizar el software para corregir vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Debatir] **IV:** Aplicar herramientas estáticas y dinámicas para identificar fallos de programación [Aplicar] **V:** Evaluar un sistema para detectar la existencia de malware y eliminarlo [Evaluar] **VI:** Implementar técnicas preventivas para reducir la ocurrencia de ransomware [Implementar]

Bibliografía: [Sikorski and Honig, 2012, Eilam, 2005, National Institute of Standards and Technology, 2020]

5.5 Análisis de Amenazas e Ingeniería de Seguridad, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Objetivos de la ingeniería de seguridad: construir sistemas que permanezcan confiables a pesar de errores, accidentes o adversarios maliciosos Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **II:** Análisis de problemas y análisis situacional para abordar la seguridad del sistema Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **III:** Análisis de seguridad, que cubre análisis de requisitos de seguridad; análisis de controles de seguridad; análisis de amenazas; y análisis de vulnerabilidades Análisis de Programas y Analizadores, Evaluación. **IV:** Dominios de ataque de seguridad y superficies de ataque, por ejemplo, comunicaciones y redes, hardware, físico, ingeniería social, software y cadena de suministro Seguridad de Red. **V:** Modos, técnicas y tácticas de ataque de seguridad, por ejemplo, abuso de autenticación; fuerza bruta; manipulación de búfer; inyección de código; inserción de contenido; denegación de servicio; escucha; omisión de función; suplantación; ataque de integridad; intercepción; phishing; análisis de protocolo; abuso de privilegio; suplantación (spoofing); e inyección de tráfico Seguridad de Red, Protección y Seguridad, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.

Objetivos de Aprendizaje **I:** Crear un modelo de amenazas para un sistema o diseño de sistema [Crear] **II:** Aplicar análisis situacional para desarrollar soluciones seguras bajo un escenario especificado [Aplicar] **III:** Identificar y mitigar vulnerabilidades y debilidades de seguridad en un sistema [Analizar]

Bibliografía: [Anderson, 2020]

5.6 Técnicas de Informática Forense, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Principios y metodologías básicas para la informática forense. **II:** Diseño de sistemas para informática forense. **III:** Informática forense en diferentes situaciones: sistemas operativos, sistemas de archivos, informática forense de aplicaciones, informática forense web, informática forense de red, informática forense de dispositivos móviles, uso de auditoría de bases de datos Seguridad de Red. **IV:** Ataques a la informática forense y prevención de tales ataques.

Objetivos de Aprendizaje **I:** Explicar qué es una investigación digital y cómo se puede implementar Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Explicar] **II:** Diseñar e implementar software para apoyar la informática forense [Diseñar] **III:** Extraer datos de un disco duro para cumplir con la ley Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Extraer] **IV:** Recuperar datos basados en un término de búsqueda dado desde un sistema del que se ha hecho una imagen (imaged) [Analizar] **V:** Reconstruir datos y eventos a partir de un historial de aplicación, o un artefacto web, o una base de datos en la nube, o un dispositivo móvil. Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo, Arquitectura Cloud y Gestión de Datos, Frontend, Frameworks y Estándares Web [Aplicar] **VI:** Capturar y analizar tráfico de red. Seguridad de Red [Analizar] **VII:** Desarrollar enfoques para abordar los desafíos asociados con la informática forense de dispositivos móviles [Crear] **VIII:** Aplicar herramientas de informática forense para investigar violaciones de seguridad [Aplicar] **IX:** Identificar y mitigar métodos anti-forenses [Analizar]

Bibliografía: [Pfleeger et al., 2015]

5.7 Gobierno, Gestión y Política de Seguridad, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Proteger activos críticos de amenazas. **II:** Gobierno de seguridad (security governance): objetivos organizacionales y evaluación general de riesgos. **III:** Gestión de seguridad (security management): lograr y mantener niveles apropiados de confidencialidad, integridad, disponibilidad, responsabilidad (accountability), autenticidad y confiabilidad Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. **IV:** Política de seguridad (security policy): políticas organizacionales, políticas específicas de temas, políticas específicas de sistemas.

Objetivos de Aprendizaje **I:** Describir activos críticos y cómo pueden protegerse [Describir] **II:** Diferenciar entre gobierno de seguridad, gestión y controles, dando ejemplos de cada uno [Diferenciar] **III:** Describir un control técnico e implementarlo para mitigar amenazas específicas [Describir]

Bibliografía: [Anderson, 2020]

5.8 Ética, Ley y Sociedad en Seguridad, 6 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Impactos sociales de las fallas en seguridad y privacidad Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **II:** Aplicabilidad de leyes y regulaciones sobre seguridad y privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **III:** Consideraciones éticas profesionales al diseñar sistemas seguros y mantener la privacidad; hacking ético Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **IV:** Implicaciones de seguridad y privacidad de actores maliciosos de IA/aprendizaje automático, por ejemplo, identificación de deep fakes Formulación del Aprendizaje Automático y Evaluación, Aprendizaje Profundo, RL y Ética, Introducción a la IA: Conceptos y Comportamiento Inteligente, Características del Problema y Naturaleza de los Agentes, Fundamentos de Aprendizaje Automático, Aprendizaje Automático: Datos, Representaciones y Evaluación, Redes Neuronales y Ética, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. **V:** Impactos sociales de los dispositivos del Internet de las Cosas (IoT) y otras tecnologías emergentes en la seguridad y la privacidad Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.

Objetivos de Aprendizaje **I:** Calcular el impacto de una falla de seguridad en un sistema dado [Calcular] **II:** Construir un sistema que se ajuste a las leyes de seguridad [Crear] **III:** Aplicar un conjunto de regulaciones de privacidad para diseñar un sistema que proteja la privacidad [Aplicar] **IV:** Evaluar las ramificaciones legales de un sistema que no se ajusta a las leyes y regulaciones aplicables [Evaluar]

Bibliografía: [Anderson, 2020, Pfleeger et al., 2015]

6. Bibliografía

[Anderson, 2020] Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 3rd edition.

[Eilam, 2005] Eilam, E. (2005). *Reversing: Secrets of Reverse Engineering*. Wiley, Indianapolis, IN.

[National Institute of Standards and Technology, 2020] National Institute of Standards and Technology (2020). Security and privacy controls for information systems and organizations. Technical Report NIST SP 800-53 Rev. 5, National Institute of Standards and Technology, Gaithersburg, MD. Disponible en: `https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final`.

[Pfleeger et al., 2015] Pfleeger, C. P., Pfleeger, S. L., and Margulies, J. (2015). *Security in Computing*. Prentice Hall, 5th edition.

[Sikorski and Honig, 2012] Sikorski, M. and Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, San Francisco, CA.

[Stallings, 2017] Stallings, W. (2017). *Computer Security: Principles and Practice*. Pearson, 4th edition.

[Anderson, 2020] Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 3rd edition.

[Eilam, 2005] Eilam, E. (2005). *Reversing: Secrets of Reverse Engineering*. Wiley, Indianapolis, IN.

[National Institute of Standards and Technology, 2020] National Institute of Standards and Technology (2020). Security and privacy controls for information systems and organizations. Technical Report NIST SP 800-53 Rev. 5, National Institute of Standards and Technology, Gaithersburg, MD. Disponible en: `https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final`.

[Pfleeger et al., 2015] Pfleeger, C. P., Pfleeger, S. L., and Margulies, J. (2015). *Security in Computing*. Prentice Hall, 5th edition.

[Sikorski and Honig, 2012] Sikorski, M. and Honig, A. (2012). *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, San Francisco, CA.

[Stallings, 2017] Stallings, W. (2017). *Computer Security: Principles and Practice*. Pearson, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

QC251 Mecánica Cuántica (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - FI101 Física General (2do Sem-Pág 333)
  - MA371 Análisis funcional I (6to Sem-Pág 414)

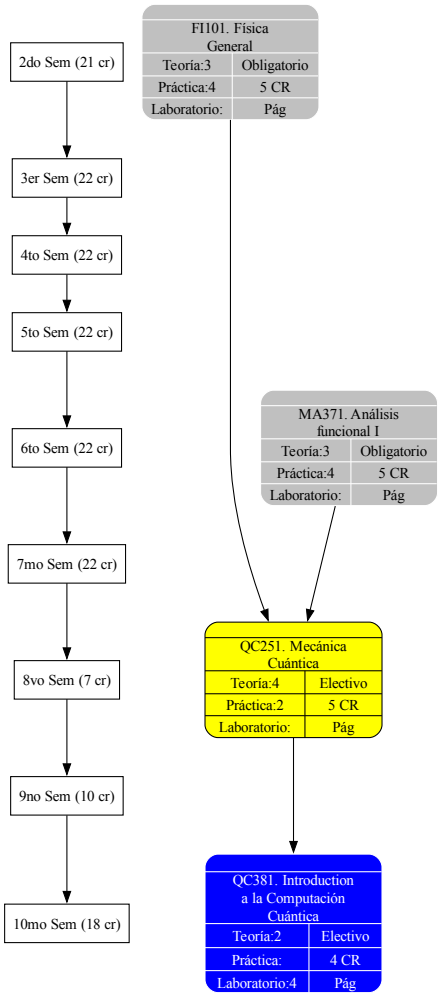


Figura 7.47: Mapa de Conexión. QC251 Mecánica Cuántica

2. Fundamentación

La mecánica cuántica es fundamental para comprender los principios físicos que gobiernan los sistemas a escala atómica y subatómica, los cuales son la base de tecnologías computacionales modernas como la computación cuántica, criptografía cuántica, semiconductores y dispositivos de estado sólido. Este curso proporciona las bases matemáticas y conceptuales de la mecánica cuántica, con especial énfasis en sus aplicaciones en ciencias de la computación, incluyendo algoritmos cuánticos, simulación cuántica y el desarrollo de hardware cuántico.

3. Objetivos del curso.

- Comprender los postulados fundamentales de la mecánica cuántica y su formulación matemática.
- Aplicar la teoría cuántica para resolver problemas básicos en sistemas de pocos niveles cuánticos.
- Relacionar los conceptos cuánticos con aplicaciones en computación cuántica y tecnologías de la información.
- Desarrollar la capacidad de simular sistemas cuánticos simples utilizando herramientas computacionales.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Fundamentos de la Mecánica Cuántica, 12 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Crisis de la física clásica: radiación del cuerpo negro, efecto fotoeléctrico. **II:** Dualidad onda-partícula y principio de complementariedad. **III:** Postulados de la mecánica cuántica: función de onda, operadores y valores esperados. **IV:** Ecuación de Schrödinger dependiente e independiente del tiempo. **V:** Interpretación probabilística de Born.

Objetivos de Aprendizaje **I:** Explicar las limitaciones de la física clásica que motivaron el desarrollo de la mecánica cuántica. [Familiarizarse] **II:** Enunciar los postulados fundamentales de la mecánica cuántica. [Familiarizarse] **III:** Interpretar el significado físico de la función de onda y su relación con probabilidades. [Usar] **IV:** Formular problemas simples usando la ecuación de Schrödinger. [Evaluar]

Bibliografía: [Griffiths and Schroeter, 2018, Nielsen and Chuang, 2010]

5.2 Sistemas Cuánticos Simples, 14 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Partícula en una caja unidimensional y tridimensional. **II:** Oscilador armónico cuántico: solución algebraica y polinomios de Hermite. **III:** Túnel cuántico a través de barreras de potencial. **IV:** Pozo de potencial finito y estados ligados. **V:** Operadores escalera para el oscilador armónico.

Objetivos de Aprendizaje **I:** Resolver analíticamente la ecuación de Schrödinger para sistemas cuánticos simples. [Familiarizarse] **II:** Calcular niveles de energía y funciones de onda para pozos de potencial simétricos. [Usar] **III:** Explicar el fenómeno de tunelamiento cuántico y sus aplicaciones tecnológicas. [Evaluar] **IV:** Utilizar operadores escalera para manipular estados cuánticos. [Evaluar]

Bibliografía: [Griffiths and Schroeter, 2018, McIntyre et al., 2012]

5.3 Álgebra Lineal en Mecánica Cuántica, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Espacios de Hilbert y notación de Dirac. **II:** Operadores lineales, hermitianos y unitarios. **III:** Valores propios y vectores propios en sistemas cuánticos. **IV:** Representaciones matriciales de operadores cuánticos. **V:** Producto tensorial y espacios compuestos.

Objetivos de Aprendizaje I: Utilizar la notación de Dirac para representar estados cuánticos. [Familiarizarse] **II:** Diferenciar entre operadores hermitianos, unitarios y de proyección. [Usar] **III:** Construir representaciones matriciales de operadores cuánticos simples. [Usar] **IV:** Aplicar el producto tensorial para describir sistemas cuánticos compuestos. [Evaluar]

Bibliografía: [Shankar, 2012, Nielsen and Chuang, 2010]

5.4 Momento Angular y Spin, 8 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Momento angular orbital: operadores y autofunciones. **II:** Spin: el grado de libertad intrínseco. **III:** Matrices de Pauli y representación de spin-1/2. **IV:** Acoplamiento de momentos angulares. **V:** Partículas idénticas y principio de exclusión de Pauli.

Objetivos de Aprendizaje I: Calcular valores propios de momento angular para sistemas simples. [Familiarizarse] **II:** Manipular estados de spin usando matrices de Pauli. [Usar] **III:** Aplicar el principio de exclusión de Pauli a sistemas de fermiones. [Evaluar] **IV:** Explicar la importancia del spin en tecnologías como la resonancia magnética. [Evaluar]

Bibliografía: [Griffiths and Schroeter, 2018, McIntyre et al., 2012]

5.5 Teoría de Perturbaciones, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Perturbaciones independientes del tiempo: teoría no degenerada y degenerada. **II:** Correcciones a niveles de energía y funciones de onda. **III:** Perturbaciones dependientes del tiempo: regla de oro de Fermi. **IV:** Transiciones inducidas por radiación. **V:** Método variacional para estados fundamentales.

Objetivos de Aprendizaje I: Aplicar la teoría de perturbaciones no degenerada a sistemas cuánticos simples. [Familiarizarse] **II:** Calcular correcciones a niveles de energía debido a pequeñas perturbaciones. [Usar] **III:** Utilizar la regla de oro de Fermi para calcular tasas de transición. [Evaluar] **IV:** Aplicar el método variacional para estimar energías del estado fundamental. [Evaluar]

Bibliografía: [Griffiths and Schroeter, 2018, Shankar, 2012]

5.6 Aplicaciones en Computación Cuántica, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Qubits: representación y operaciones básicas. **II:** Puertas cuánticas y circuitos cuánticos elementales. **III:** Entrelazamiento cuántico y teorema de no-clonación. **IV:** Algoritmos cuánticos simples: Deutsch-Jozsa y teleportación cuántica. **V:** Simulación de sistemas cuánticos en computadoras clásicas. **VI:** Implementación de algoritmos cuánticos usando frameworks como Qiskit o Cirq.

Objetivos de Aprendizaje I: Representar y manipular qubits usando notación de Dirac. [Familiarizarse] **II:** Explicar el concepto de entrelazamiento cuántico y sus aplicaciones. [Usar] **III:** Implementar circuitos cuánticos simples utilizando frameworks de programación cuántica. [Usar] **IV:** Simular sistemas cuánticos de pocos qubits en ambientes computacionales. [Evaluar] **V:** Describir las ventajas fundamentales de la computación cuántica sobre la clásica. [Evaluar]

Bibliografía: [Nielsen and Chuang, 2010, Rieffel and Polak, 2011]

6. Bibliografía

[Griffiths and Schroeter, 2018] Griffiths, D. J. and Schroeter, D. F. (2018). *Introduction to Quantum Mechanics*. Cambridge University Press, 3rd edition. Texto estándar para cursos introductorios de mecánica cuántica, con explicaciones claras y problemas bien seleccionados.

[McIntyre et al., 2012] McIntyre, D. H., Manogue, C. A., and Tate, J. (2012). *Quantum Mechanics: A Paradigms Approach*. Pearson. Enfoque moderno con aplicaciones experimentales y conexiones con tecnologías actuales.

[Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition. La biblia de la computación cuántica, conecta teoría cuántica con aplicaciones en computación e información.

[Rieffel and Polak, 2011] Rieffel, E. G. and Polak, W. H. (2011). *Quantum Computing: A Gentle Introduction*. MIT Press. Introducción accesible a la computación cuántica, ideal para estudiantes de ciencias de la computación.

[Shankar, 2012] Shankar, R. (2012). *Principles of Quantum Mechanics*. Springer, 2nd edition. Excelente para el formalismo matemático y tratamiento riguroso de los fundamentos.

[Griffiths and Schroeter, 2018] Griffiths, D. J. and Schroeter, D. F. (2018). *Introduction to Quantum Mechanics*. Cambridge University Press, 3rd edition. Texto estándar para cursos introductorios de mecánica cuántica, con explicaciones claras y problemas bien seleccionados.

[McIntyre et al., 2012] McIntyre, D. H., Manogue, C. A., and Tate, J. (2012). *Quantum Mechanics: A Paradigms Approach*. Pearson. Enfoque moderno con aplicaciones experimentales y conexiones con tecnologías actuales.

[Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition. La biblia de la computación cuántica, conecta teoría cuántica con aplicaciones en computación e información.

[Rieffel and Polak, 2011] Rieffel, E. G. and Polak, W. H. (2011). *Quantum Computing: A Gentle Introduction*. MIT Press. Introducción accesible a la computación cuántica, ideal para estudiantes de ciencias de la computación.

[Shankar, 2012] Shankar, R. (2012). *Principles of Quantum Mechanics*. Springer, 2nd edition. Excelente para el formalismo matemático y tratamiento riguroso de los fundamentos.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS253 Análisis exploratorio de datos (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA351 Estadística y Probabilidades (6to Sem-Pág 408)

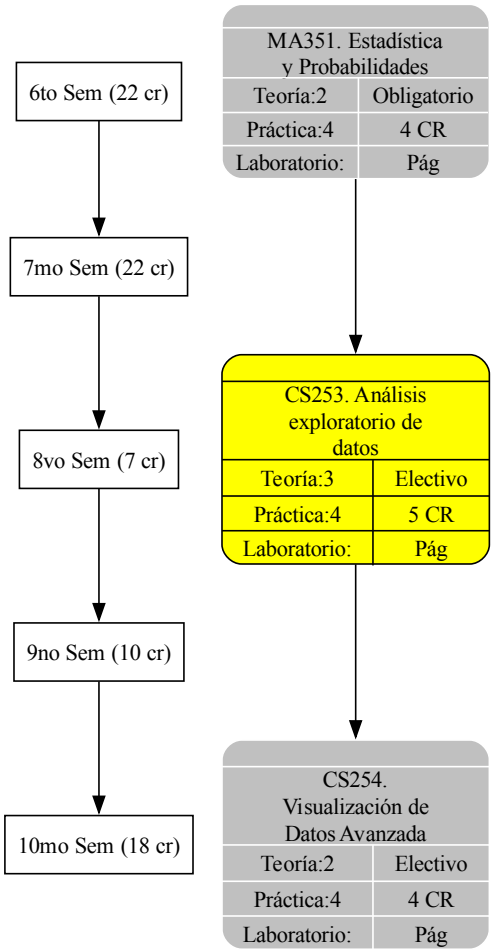


Figura 7.48: Mapa de Conexión. CS253 Análisis exploratorio de datos

2. Fundamentación

El Análisis Exploratorio de Datos (EDA) es una etapa crítica en cualquier proyecto de ciencia de datos. Este curso proporciona las herramientas para entender la estructura, detectar anomalías y extraer patrones iniciales de los datos antes del modelado formal. Se enfoca en el uso de estadísticas descriptivas, limpieza de datos y visualización preliminar para validar supuestos y guiar la selección de modelos complejos, asegurando la integridad y calidad de los hallazgos científicos.

3. Objetivos del curso.

- Aplicar técnicas de limpieza y preprocesamiento de datos para asegurar su calidad.
- Utilizar métodos estadísticos descriptivos para resumir conjuntos de datos complejos.
- Identificar patrones, valores atípicos y relaciones entre variables mediante análisis visual.
- Formular hipótesis basadas en la exploración inicial de los datos.
- Comunicar hallazgos preliminares de manera efectiva a través de reportes técnicos.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Fundamentos del Análisis Exploratorio de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Filosofía y objetivos del análisis exploratorio. **II:** Tipos de datos: estructurados, semi-estructurados y no estructurados. **III:** Escalas de medición: nominal, ordinal, intervalo y razón. **IV:** Flujo de trabajo en la exploración de datos.

Objetivos de Aprendizaje I: Identificar la importancia del EDA en el ciclo de vida de los datos [Familiarizarse]. **II:** Clasificar variables según su escala de medición y tipo [Usar].

Bibliografía: [Tukey, 1977, Peng,]

5.2 Limpieza y Preparación de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Concepto de "Tidy Dataz manipulación de tablas. **II:** Manejo de valores faltantes: imputación versus eliminación. **III:** Detección y tratamiento de valores atípicos (outliers). **IV:** Normalización y estandarización de variables numéricas.

Objetivos de Aprendizaje I: Implementar rutinas de limpieza para conjuntos de datos ruidosos [Usar]. **II:** Evaluar el impacto de diferentes estrategias de imputación [Evaluar].

Bibliografía: [Wickham, 2014, McKinney, 2022]

5.3 Estadística Descriptiva Univariada, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Medidas de tendencia central, dispersión y forma. **II:** Distribuciones de frecuencia y funciones de densidad. **III:** Análisis de asimetría y curtosis en variables reales. **IV:** Visualización univariada: Histogramas, Boxplots y estimación de densidad (KDE).

Objetivos de Aprendizaje I: Resumir numéricamente la distribución de una variable [Usar]. **II:** Interpretar gráficos de densidad para identificar sesgos en los datos [Evaluar].

Bibliografía: [Healy, 2018, Tukey, 1977]

5.4 Exploración Bivariada y Multivariada, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Correlación de Pearson, Spearman y Kendall. **II:** Tablas de contingencia y medidas de asociación categórica. **III:** Diagramas de dispersión, matrices de correlación y mapas de calor. **IV:** Introducción conceptual a la reducción de dimensionalidad (PCA).

Objetivos de Aprendizaje I: Cuantificar la relación entre múltiples variables de un dataset [Usar]. **II:** Utilizar mapas de calor para detectar colinealidad en conjuntos de datos [Evaluar].

Bibliografía: [Healy, 2018, Wickham and Grolemund, 2016]

5.5 Herramientas Computacionales para Exploración, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Ecosistemas computacionales para análisis (Python/Pandas o R/Tidyverse). **II:** Manipulación eficiente de DataFrames de gran volumen. **III:** Generación automática de reportes de perfilado de datos (Data Profiling).

Objetivos de Aprendizaje I: Implementar procesos de EDA automatizados mediante programación [Usar]. **II:** Optimizar el procesamiento de datos para análisis eficiente [Evaluar].

Bibliografía: [McKinney, 2022, Wickham and Grolemund, 2016]

5.6 Pruebas de Hipótesis Exploratorias, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Pruebas de normalidad y supuestos de varianza. **II:** Comparación de grupos: T-test y ANOVA en contexto exploratorio. **III:** Métodos no paramétricos y remuestreo (Bootstraping).

Objetivos de Aprendizaje I: Validar supuestos estadísticos necesarios para el modelado posterior [Usar]. **II:** Diseñar experimentos exploratorios para confirmar hallazgos visuales [Evaluar].

Bibliografía: [Peng, , Healy, 2018]

5.7 Ética de Datos y Responsabilidad, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Identificación de sesgos de selección y confirmación en los datos. **II:** Privacidad de datos: Anonimización y técnicas de enmascaramiento. **III:** Ética en la comunicación de hallazgos preliminares y riesgos asociados.

Objetivos de Aprendizaje I: Identificar posibles sesgos éticos en conjuntos de datos reales [Familiarizarse]. **II:** Aplicar técnicas de anonimización básicas durante el análisis [Usar].

Bibliografía: [Healy, 2018]

5.8 Comunicación de Resultados y Narrativa de Datos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Estructura de un reporte técnico de calidad de datos. **II:** Narrativa (Storytelling) basada en el descubrimiento exploratorio. **III:** Selección de visualizaciones clave para la toma de decisiones.

Objetivos de Aprendizaje I: Redactar informes técnicos efectivos sobre la calidad de los datos [Evaluar]. **II:** Realizar presentaciones de hallazgos exploratorios ante audiencias técnicas [Evaluar].

Bibliografía: [Wickham and Grolemund, 2016, Healy, 2018]

6. Bibliografía

[Healy, 2018] Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press.

[McKinney, 2022] McKinney, W. (2022). *Python for Data Analysis*. O'Reilly Media, 3rd edition.

[Peng,] Peng, R. D. *Exploratory Data Analysis with R*. Leanpub.

[Tukey, 1977] Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

[Wickham, 2014] Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10):1–23.

[Wickham and Grolemund, 2016] Wickham, H. and Grolemund, G. (2016). *R for Data Science*. O'Reilly Media.

[Healy, 2018] Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press.

[McKinney, 2022] McKinney, W. (2022). *Python for Data Analysis*. O'Reilly Media, 3rd edition.

[Peng,] Peng, R. D. *Exploratory Data Analysis with R*. Leanpub.

[Tukey, 1977] Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

[Wickham, 2014] Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10):1–23.

[Wickham and Grolemund, 2016] Wickham, H. and Grolemund, G. (2016). *R for Data Science*. O'Reilly Media.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS370 Big Data y Analítica de Datos (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS271 Bases de Datos I (4to Sem-Pág 366)

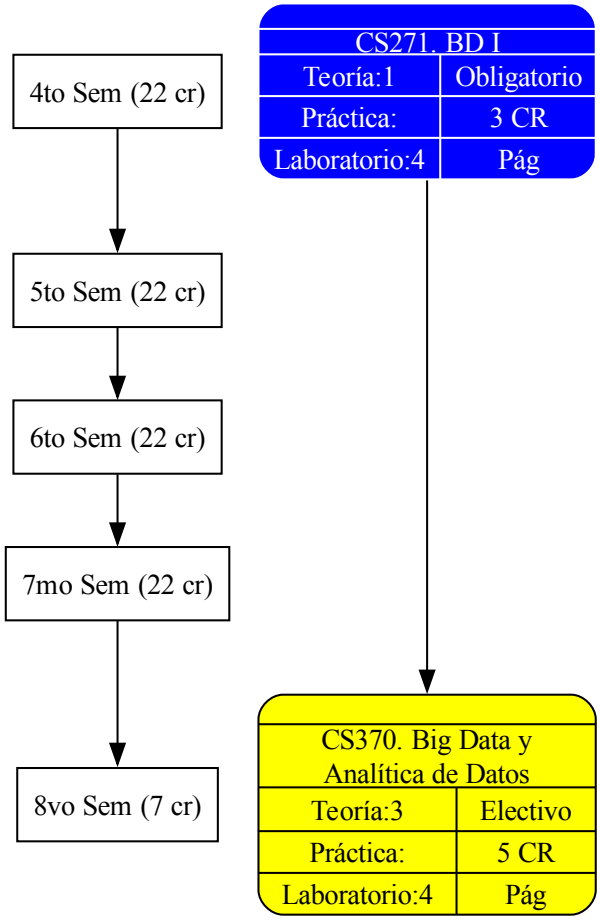


Figura 7.49: Mapa de Conexión. CS370 Big Data y Analítica de Datos

2. Fundamentación

En la era actual, el volumen, velocidad y variedad de los datos superan las capacidades de los sistemas tradicionales de almacenamiento y procesamiento. Este curso introduce los conceptos y tecnologías fundamentales del Big Data. Cubre sistemas de archivos distribuidos, frameworks de procesamiento paralelo, bases de datos NoSQL y flujo de datos en tiempo real. Los estudiantes aprenderán a diseñar arquitecturas escalables para extraer valor de conjuntos de datos masivos.

3. Objetivos del curso.

- Comprender las 5 V's del Big Data y su impacto en el diseño de sistemas.
- Diseñar e implementar soluciones escalables usando el paradigma MapReduce.
- Dominar el uso de sistemas de archivos distribuidos como HDFS.
- Analizar diferentes patrones arquitectónicos para Big Data (Lambda y Kappa).
- Aplicar técnicas de procesamiento de flujos para análisis de datos en tiempo real.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Conceptos Fundamentales de Sistemas de Bases de Datos, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Propósito y ventajas de los sistemas de bases de datos **II:** Componentes de los sistemas de bases de datos **III:** Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) **IV:** Arquitectura de base de datos, independencia de datos y abstracción de datos **V:** Gestión de transacciones **VI:** Normalización **VII:** Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) **VIII:** Cómo soportar aplicaciones solo de CRUD **IX:** Bases de datos distribuidas/sistemas basados en la nube **X:** Datos estructurados, semiestructurados y no estructurados **XI:** Uso de un lenguaje de consulta declarativo **XII:** Sistemas que soportan contenido estructurado y/o de flujo

Objetivos de Aprendizaje **I:** Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Analizar] **II:** Enumerar los componentes de un sistema de base de datos (relacional) [Enumerar] **III:** Seguir una consulta a medida que es procesada por los componentes de un sistema de base de datos (relacional) [Analizar] **IV:** Defender el valor de la independencia de datos [Defender] **V:** Componer una consulta simple de selección-proyección-unión en SQL [Componer] **VI:** Enumerar las cuatro propiedades de un gestor de transacciones correcto [Enumerar] **VII:** Describir las ventajas de eliminar datos duplicados repetidos [Describir] **VIII:** Esbozar cómo MapReduce usa paralelismo para procesar datos eficientemente [Esquematizar/Esbozar] **IX:** Evaluar las diferencias entre bases de datos estructuradas y semiestructuradas/no estructuradas [Evaluar]

Bibliografía: [White, 2015, Karau et al., 2017]

5.2 Bases de Datos Distribuidas/Computación en la Nube, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** DBMS Distribuido: *a*) Almacenamiento de datos distribuido *b*) Procesamiento de consultas distribuido *c*) Modelo de transacción distribuida *d*) Soluciones homogéneas y heterogéneas *e*) Bases de datos distribuidas cliente-servidor **II:** DBMS Paralelo: *a*) Arquitecturas de DBMS paralelo: memoria compartida, disco compartido, nada compartido *b*) Aceleración y escalamiento, por ejemplo, uso del modelo de procesamiento MapReduce *c*) Replicación de datos y modelos de consistencia débil

Objetivos de Aprendizaje **I:** Describir los componentes clave de un DBMS distribuido, incluyendo almacenamiento de datos distribuido, procesamiento de consultas y gestión de transacciones

[Describir] **II:** Analizar las ventajas y desventajas entre arquitecturas de DBMS paralelo: memoria compartida, disco compartido y nada compartido [Analizar] **III:** Describir estrategias de replicación de datos y modelos de consistencia débil en sistemas de bases de datos distribuidas [Describir]

Bibliografía: [White, 2015, Lin and Dyer, 2010]

5.3 Sistemas NoSQL, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: ¿Por qué NoSQL? (por ejemplo, Desajuste de impedancia entre Aplicación [CRUD] y RDBMS) **II:** Modelo de datos Clave-Valor y Documento **III:** Sistemas de almacenamiento (por ejemplo, sistemas Clave-Valor, Lagos de Datos (*Data Lakes*)) **IV:** Modelos de Distribución (Fragmentación y Replicación) **V:** Bases de Datos de Grafos **VI:** Modelos de Consistencia (Actualización y Lectura, consistencia de quórum, teorema CAP) **VII:** Modelo de procesamiento (por ejemplo, Map-Reduce, map-reduce multi-etapa, map-reduce incremental) **VIII:** Estudios de Caso: Sistema de almacenamiento en la nube (por ejemplo, S3); Bases de datos de grafos; "Cuándo no usar NoSQL"

Objetivos de Aprendizaje I: Desarrollar un caso de uso para el uso de NoSQL sobre RDBMS [Crear] **II:** Describir las características definitorias detrás de los modelos de datos basados en Clave-Valor y Documentos [Describir]

Bibliografía: [Sadalage and Fowler, 2012, Kleppmann, 2017]

5.4 Análisis de Datos, 12 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Técnicas exploratorias de datos (motivación, representación, estadísticas descriptivas, visualizaciones) **II:** Ciclo de vida de ciencia de datos: comprensión del negocio, comprensión de datos, preparación de datos, modelado, evaluación, despliegue y aceptación del usuario **III:** Algoritmos de minería de datos y aprendizaje automático: por ejemplo, clasificación, agrupamiento, asociación, regresión **IV:** Adquisición y gobernanza de datos **V:** Consideraciones de seguridad y privacidad de datos **VI:** Equidad y sesgo de datos **VII:** Técnicas de visualización de datos y su uso en análisis de datos **VIII:** Resolución de Entidades

Objetivos de Aprendizaje I: Describir varios enfoques de exploración de datos, incluyendo visualización, para comprender conjuntos de datos no familiares [Describir] **II:** Aplicar varios enfoques de exploración de datos para comprender conjuntos de datos no familiares [Aplicar] **III:** Describir algoritmos básicos de aprendizaje automático/minería de datos y cuándo son apropiados para su uso [Describir] **IV:** Aplicar varios algoritmos de aprendizaje automático/minería de datos [Aplicar] **V:** Describir consideraciones legales y éticas en la adquisición, uso y modificación de conjuntos de datos [Describir] **VI:** Describir problemas de equidad y sesgo en la recolección y uso de datos [Describir]

Bibliografía: [Matei et al., 2012, Bill and Matei, 2018, Kleppmann, 2017, Psaltis, 2017]

6. Bibliografía

[Bill and Matei, 2018] Bill, C. and Matei, Z. (2018). *Spark: The Definitive Guide: Big Data Processing Made Simple*. O'Reilly Media.

[Karau et al., 2017] Karau, H., Konwinski, A., Wendell, P., and Zaharia, M. (2017). *Learning Spark: Lightning-Fast Big Data Analysis*. O'Reilly Media.

[Kleppmann, 2017] Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.

[Lin and Dyer, 2010] Lin, J. and Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers.

[Matei et al., 2012] Matei, Z., Mosharaf, C., Tathagata, D., Ankur, D., Justin, M., Murphy, M., J, F. M., Scott, S., and Ion, S. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association.

[Psaltis, 2017] Psaltis, A. (2017). *Streaming Data: Understanding the real-time pipeline*. Manning Publications.

[Sadalage and Fowler, 2012] Sadalage, P. J. and Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.

[White, 2015] White, T. (2015). *Hadoop: The Definitive Guide*. O'Reilly Media, 4th edition.

[Bill and Matei, 2018] Bill, C. and Matei, Z. (2018). *Spark: The Definitive Guide: Big Data Processing Made Simple*. O'Reilly Media.

[Karau et al., 2017] Karau, H., Konwinski, A., Wendell, P., and Zaharia, M. (2017). *Learning Spark: Lightning-Fast Big Data Analysis*. O'Reilly Media.

[Kleppmann, 2017] Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.

[Lin and Dyer, 2010] Lin, J. and Dyer, C. (2010). *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers.

[Matei et al., 2012] Matei, Z., Mosharaf, C., Tathagata, D., Ankur, D., Justin, M., Murphy, M., J, F. M., Scott, S., and Ion, S. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association.

[Psaltis, 2017] Psaltis, A. (2017). *Streaming Data: Understanding the real-time pipeline*. Manning Publications.

[Sadalage and Fowler, 2012] Sadalage, P. J. and Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.

[White, 2015] White, T. (2015). *Hadoop: The Definitive Guide*. O'Reilly Media, 4th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

STA252 Estadística Inferencial (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA351 Estadística y Probabilidades (6to Sem-Pág 408)

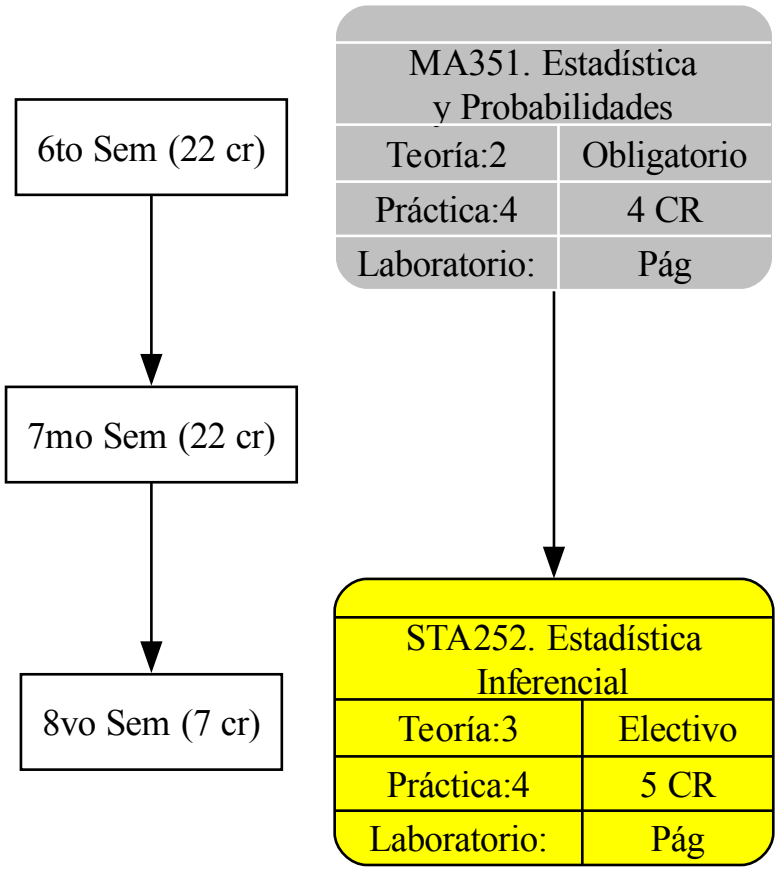


Figura 7.50: Mapa de Conexión. STA252 Estadística Inferencial

2. Fundamentación

La estadística inferencial es crucial en ciencias de la computación para analizar datos, extraer conclusiones y tomar decisiones basadas en evidencia. Este curso cubre los principios de la inferencia estadística, incluyendo estimación puntual, intervalos de confianza y pruebas de hipótesis.

3. Objetivos del curso.

- Comprender los conceptos fundamentales de la estadística inferencial.

- Aplicar técnicas de estimación puntual e intervalos de confianza.
- Realizar pruebas de hipótesis para tomar decisiones basadas en datos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Familiarizarse)

5. Contenido

5.1 Introducción a la Inferencia Estadística, 6 hr(s)

Competencias: ABET-1

Tópicos: I: Muestreo y distribuciones muestrales. **II:** Estimación puntual. **III:** Intervalos de confianza.

Objetivos de Aprendizaje I: Describir los principios del muestreo y las distribuciones muestrales. [Familiarizarse] **II:** Calcular estimaciones puntuales para parámetros poblacionales. [Usar] **III:** Construir e interpretar intervalos de confianza. [Evaluar]

Bibliografía: [Devore, , Montgomery and Runger,]

5.2 Prueba de Hipótesis, 10 hr(s)

Competencias: ABET-1

Tópicos: I: Hipótesis nula y alternativa. **II:** Errores Tipo I y Tipo II. **III:** Valor p. **IV:** Pruebas de hipótesis para medias y proporciones.

Objetivos de Aprendizaje I: Formular hipótesis nula y alternativa. [Familiarizarse] **II:** Calcular e interpretar el valor p. [Usar] **III:** Realizar pruebas de hipótesis para medias y proporciones. [Evaluar]

Bibliografía: [Devore, , Montgomery and Runger,]

5.3 Análisis de Varianza (ANOVA), 8 hr(s)

Competencias: ABET-1

Tópicos: I: ANOVA de una vía. **II:** ANOVA de dos vías.

Objetivos de Aprendizaje I: Aplicar ANOVA de una vía para comparar medias de múltiples grupos. [Familiarizarse] **II:** Realizar ANOVA de dos vías para analizar la influencia de dos variables. [Usar]

Bibliografía: [Devore, , Montgomery and Runger,]

5.4 Regresión Lineal, 8 hr(s)

Competencias: ABET-1

Tópicos: I: Regresión lineal simple. **II:** Regresión lineal múltiple.

Objetivos de Aprendizaje I: Ajustar un modelo de regresión lineal simple. [Familiarizarse] **II:** Interpretar los coeficientes de un modelo de regresión lineal. [Usar] **III:** Aplicar regresión lineal múltiple para modelar la relación entre múltiples variables. [Evaluar]

Bibliografía: [Devore, , Montgomery and Runger,]

5.5 Pruebas No Paramétricas, 8 hr(s)

Competencias: ABET-1

Tópicos: **I:** Prueba de chi-cuadrado. **II:** Prueba U de Mann-Whitney. **III:** Prueba de Kruskal-Wallis.

Objetivos de Aprendizaje **I:** Aplicar pruebas no paramétricas cuando no se cumplen los supuestos de las pruebas paramétricas. [Familiarizarse] **II:** Realizar la prueba de chi-cuadrado para independencia. [Usar] **III:** Usar las pruebas U de Mann-Whitney y Kruskal-Wallis para comparar grupos. [Evaluar]

Bibliografía: [Devore,]

5.6 Aplicaciones en Computación, 8 hr(s)

Competencias: ABET-1

Tópicos: **I:** Análisis de datos en experimentos computacionales. **II:** Validación de modelos. **III:** Aprendizaje automático (ej., evaluación de modelos).

Objetivos de Aprendizaje **I:** Aplicar técnicas de estadística inferencial para analizar datos en experimentos computacionales. [Familiarizarse] **II:** Usar métodos estadísticos para validar modelos computacionales. [Usar] **III:** Aplicar estadística inferencial en la evaluación de modelos de aprendizaje automático. [Evaluar]

Bibliografía: [Devore,]

6. Bibliografía

[Devore,] Devore, J. L. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 10th edition.

[Montgomery and Runger,] Montgomery, D. C. and Runger, G. C. *Applied Statistics and Probability for Engineers*. Wiley, 8th edition.

[Devore,] Devore, J. L. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 10th edition.

[Montgomery and Runger,] Montgomery, D. C. and Runger, G. C. *Applied Statistics and Probability for Engineers*. Wiley, 8th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA327 Análisis Numérico IV (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - MA326 Análisis Numérico III (7mo Sem-Pág 441)
  - MA213 Ecuaciones Diferenciales Parciales (7mo Sem-Pág 438)

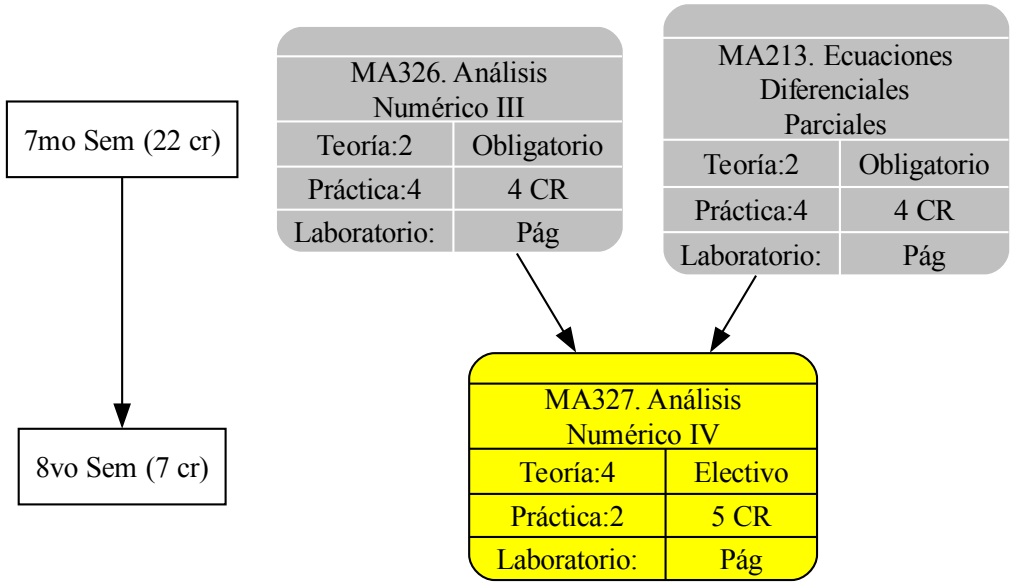


Figura 7.51: Mapa de Conexión. MA327 Análisis Numérico IV

2. Fundamentación

Análisis Numérico 4 proporciona las herramientas computacionales necesarias para resolver Ecuaciones Diferenciales Parciales (EDP), las cuales describen una amplia gama de fenómenos en física, ingeniería y finanzas. El curso se centra en los tres tipos fundamentales de EDP de segundo orden: elípticas, parabólicas e hiperbólicas. Los estudiantes aprenderán a discretizar dominios continuos, analizar la estabilidad y convergencia de esquemas numéricos e implementar algoritmos para problemas complejos con condiciones de contorno.

3. Objetivos del curso.

- Clasificar las ecuaciones diferenciales parciales y comprender sus implicaciones físicas.
- Desarrollar e implementar el Método de Diferencias Finitas (MDF) para diferentes tipos de EDP.
- Analizar la estabilidad (Von Neumann), consistencia y convergencia de esquemas numéricos.
- Introducir técnicas avanzadas de discretización como los métodos de Elementos Finitos y Volúmenes Finitos.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Ecuaciones Elípticas y Problemas de Valor de Contorno, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Clasificación de EDPs de segundo orden **II:** Método de Diferencias Finitas para las ecuaciones de Laplace y Poisson **III:** Condiciones de contorno: Dirichlet, Neumann y Robin **IV:** Solución de sistemas dispersos: Jacobi, Gauss-Seidel y SOR **V:** Introducción al Método de Elementos Finitos (MEF) en 1D

Objetivos de Aprendizaje **I:** Discretizar problemas de estado estacionario usando plantillas (stencils) de cinco puntos [Usage] **II:** Resolver los sistemas lineales resultantes mediante métodos iterativos optimizados [Assessment] **III:** Evaluar el efecto del refinamiento de la malla en la precisión de la solución [Familiarity]

Bibliografía: [Burden et al., 2016, Quarteroni et al., 2007]

5.2 Ecuaciones Parabólicas y Problemas de Evolución, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** La ecuación del calor en 1D y 2D **II:** Esquemas explícitos y restricciones de estabilidad (condición CFL) **III:** Esquemas implícitos: Crank-Nicolson y Euler regresivo **IV:** Análisis de consistencia y estabilidad de Von Neumann

Objetivos de Aprendizaje **I:** Implementar algoritmos de avance temporal para procesos de difusión [Usage] **II:** Determinar los límites de estabilidad de los esquemas numéricos explícitos [Assessment] **III:** Aplicar el algoritmo de Thomas para sistemas tridiagonales en métodos implícitos [Usage]

Bibliografía: [Heath, 2002, Quarteroni et al., 2007]

5.3 Ecuaciones Hiperbólicas y Fenómenos Ondulatorios, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Ecuaciones de advección de primer orden **II:** La ecuación de onda y la solución de D'Alembert **III:** Esquemas Upwind, Lax-Wendroff y Leapfrog **IV:** Errores de dispersión y disipación en la propagación de ondas

Objetivos de Aprendizaje **I:** Diseñar esquemas numéricos para ecuaciones de transporte y ondas [Usage] **II:** Analizar la difusión y oscilación numérica en esquemas de alto orden [Assessment] **III:** Verificar la condición de Courant-Friedrichs-Lewy (CFL) para sistemas hiperbólicos [Usage]

Bibliografía: [LeVeque, 2007, Heath, 2002]

5.4 Temas Avanzados en EDPs Numéricas, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** EDPs no lineales y la ecuación de Burgers **II:** Introducción al Método de Volúmenes Finitos (MVF) para leyes de conservación **III:** Problemas multidimensionales y métodos implícitos de dirección alternante (ADI) **IV:** Implementación computacional usando Python/MATLAB

Objetivos de Aprendizaje **I:** Implementar resolvedores numéricos para leyes de conservación no lineales [Usage] **II:** Generalizar métodos de 1D a geometrías complejas en 2D/3D [Familiarity] **III:** Realizar estudios comparativos de diferentes técnicas de discretización [Assessment]

Bibliografía: [Quarteroni et al., 2007, LeVeque, 2007]

6. Bibliografía

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Numerical Analysis*. Cengage Learning, 10th edition.

[Heath, 2002] Heath, M. T. (2002). *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2nd edition.

[LeVeque, 2007] LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM.

[Quarteroni et al., 2007] Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer, 2nd edition.

[Burden et al., 2016] Burden, R. L., Faires, J. D., and Burden, A. M. (2016). *Numerical Analysis*. Cengage Learning, 10th edition.

[Heath, 2002] Heath, M. T. (2002). *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2nd edition.

[LeVeque, 2007] LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM.

[Quarteroni et al., 2007] Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA352 Procesos estocásticos (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA351 Estadística y Probabilidades (6to Sem-Pág 408)
  - MA210 Cálculo III (4to Sem-Pág 374)
  - CS261 Inteligencia Artificial (7mo Sem-Pág 421)

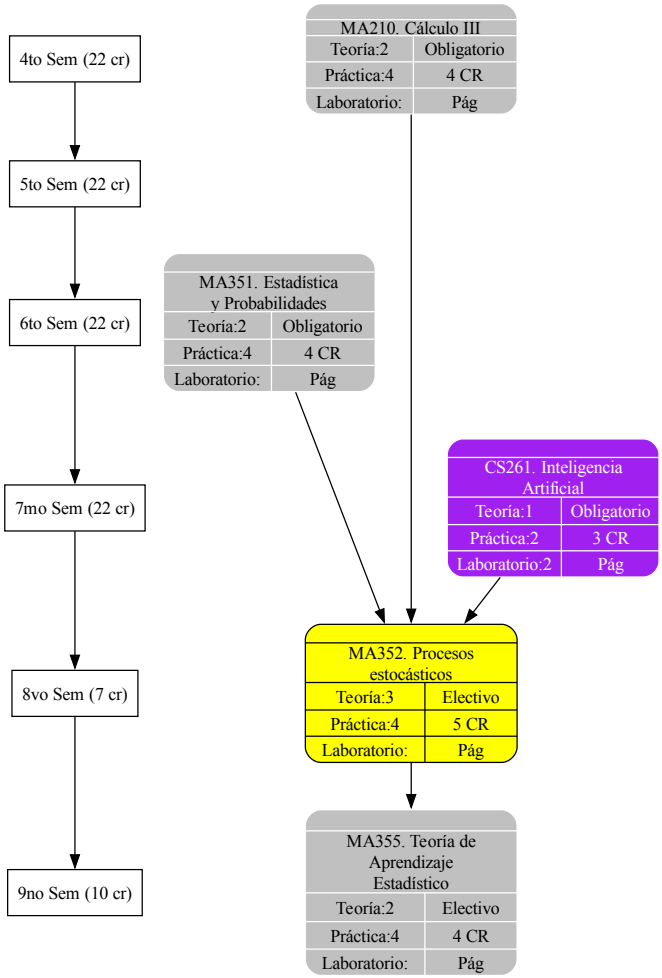


Figura 7.52: Mapa de Conexión. MA352 Procesos estocásticos

2. Fundamentación

Este curso introduce la teoría de procesos estocásticos, proporcionando las bases matemáticas para el estudio de sistemas que evolucionan aleatoriamente en el tiempo. Desarrolla habilidades para modelar fenómenos aleatorios en finanzas, ingeniería y ciencias mediante cadenas de Markov, procesos

de Poisson y movimiento browniano, con aplicaciones en simulación y análisis de sistemas estocásticos.

3. Objetivos del curso.

- Comprender los conceptos fundamentales de procesos estocásticos
- Modelar sistemas usando cadenas de Markov en tiempo discreto y continuo
- Analizar procesos de Poisson y sus aplicaciones
- Aplicar procesos estocásticos a problemas en finanzas y ciencias

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Probabilidad Avanzada y Procesos Estocásticos, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Fundamentos de probabilidad medida-teórica **II:** Variables aleatorias y distribuciones **III:** Caracterización de procesos estocásticos **IV:** Cadenas y procesos de Markov **V:** Movimiento browniano y procesos de difusión

Objetivos de Aprendizaje **I:** Aplicar conceptos medida-teóricos a espacios de probabilidad [Familiarizarse] **II:** Caracterizar diferentes tipos de procesos estocásticos [Evaluar] **III:** Analizar cadenas de Markov para comportamiento transitorio y en estado estacionario [Usar] **IV:** Modelar sistemas usando procesos estocásticos en tiempo continuo [Usar]

Bibliografía: [Ross, 2019, Karlin and Taylor, 2014]

5.2 Simulación de Sistemas Discretos, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Teoría de colas y modelos **II:** Autómatas celulares y modelos basados en agentes **III:** Métodos de simulación Monte Carlo **IV:** Cadenas y procesos de Markov **V:** Simulación de eventos discretos

Objetivos de Aprendizaje **I:** Formular y analizar modelos simples de colas [Usar] **II:** Implementar autómatas celulares y simulaciones basadas en agentes [Usar] **III:** Aplicar métodos Monte Carlo para simulación probabilística [Usar] **IV:** Analizar cadenas de Markov para transiciones de estado y comportamiento en estado estacionario [Evaluar] **V:** Diseñar simulaciones de eventos discretos para análisis de sistemas [Usar]

Bibliografía: [Ross, 2019, Karlin and Taylor, 2014]

5.3 Inferencia Estadística Computacional, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Estimación de Máxima Verosimilitud (MLE) **II:** Métodos de inferencia bayesiana **III:** Métodos de Monte Carlo mediante Cadenas de Markov (MCMC) **IV:** Algoritmo Expectación-Maximización (EM) **V:** Remuestreo bootstrap y jackknife

Objetivos de Aprendizaje I: Implementar Estimación de Máxima Verosimilitud para varias distribuciones [Usar] **II:** Aplicar métodos bayesianos para estimación de parámetros [Usar] **III:** Implementar algoritmos MCMC para muestreo posterior [Usar] **IV:** Usar el algoritmo EM para modelos con variables latentes [Usar] **V:** Aplicar métodos bootstrap para intervalos de confianza [Usar]

Bibliografía: [Ross, 2019, Karlin and Taylor, 2014]

5.4 Simulación de Sistemas Continuos, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Modelos de Ecuaciones Diferenciales Ordinarias (EDOs) **II:** Modelos de Ecuaciones Diferenciales Parciales (EDPs) **III:** Métodos de diferencias finitas para EDPs **IV:** Selección e implementación de solucionadores **V:** Sistemas stiff y métodos adaptativos

Objetivos de Aprendizaje I: Formular modelos de EDOs y EDPs para sistemas físicos [Usar] **II:** Implementar métodos de diferencias finitas para EDPs simples [Usar] **III:** Seleccionar solucionadores numéricos apropiados para diferentes tipos de EDOs/EDPs [Evaluar] **IV:** Manejar sistemas stiff usando métodos numéricos apropiados [Usar]

Bibliografía: [Ross, 2019, Karlin and Taylor, 2014]

6. Bibliografía

[Karlin and Taylor, 2014] Karlin, S. and Taylor, H. M. (2014). *A First Course in Stochastic Processes*. Academic Press, 2nd edition.

[Ross, 2019] Ross, S. M. (2019). *Introduction to Probability Models*. Academic Press, 12th edition.

[Karlin and Taylor, 2014] Karlin, S. and Taylor, H. M. (2014). *A First Course in Stochastic Processes*. Academic Press, 2nd edition.

[Ross, 2019] Ross, S. M. (2019). *Introduction to Probability Models*. Academic Press, 12th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA372 Análisis funcional II (Electivo)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA371 Análisis funcional I (6to Sem-Pág 414)

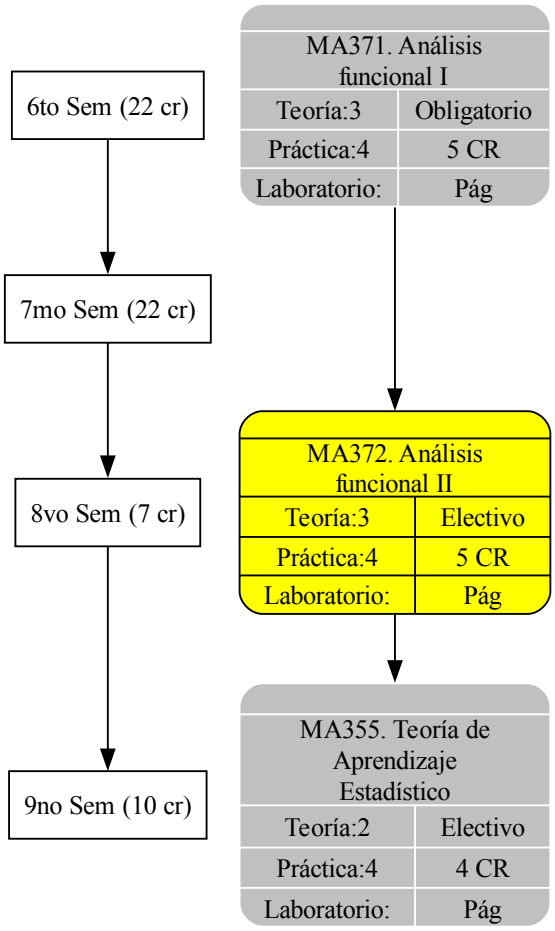


Figura 7.53: Mapa de Conexión. MA372 Análisis funcional II

2. Fundamentación

Este curso profundiza en los fundamentos teóricos de la medida y la integración, proporcionando las herramientas esenciales para el análisis avanzado en espacios de dimensión infinita. Se centra en la construcción rigurosa de medidas abstractas y la operatividad de la integral de Lebesgue. Finalmente, se estudia la estructura geométrica y de operadores en espacios de Hilbert, fundamentales para el desarrollo de métodos computacionales zy modelos en física matemática.

3. Objetivos del curso.

- Comprender y construir medidas abstractas y extensiones de medidas en espacios generales.
- Dominar la teoría de integración de Lebesgue y sus teoremas de convergencia.
- Analizar la dualidad y descomposición de medidas mediante el teorema de Radon-Nikodym.
- Estudiar la geometría avanzada y la teoría de operadores en espacios de Hilbert.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Teoría de la Medida, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Álgebras, σ-álgebras y medidas **II:** Medidas exteriores y el Teorema de Extensión de Carathéodory **III:** Medidas de Borel en la recta real **IV:** Medidas con signo y la descomposición de Hahn y Jordan **V:** Medidas complejas y variación total

Objetivos de Aprendizaje **I:** Construir medidas a partir de medidas exteriores utilizando el criterio de Carathéodory [Assessment] **II:** Aplicar la descomposición de Jordan para analizar medidas con signo [Usage] **III:** Demostrar propiedades de regularidad de medidas de Borel [Familiarity]

Bibliografía: [Folland, 1999, Royden and Fitzpatrick, 2010]

5.2 Teoría de Integración, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Funciones medibles y funciones simples **II:** La integral de Lebesgue: definición y propiedades fundamentales **III:** Teoremas fundamentales: Convergencia Monótona, Dominada y Lema de Fatou **IV:** El Teorema de Radon-Nikodym y la propiedad de Lebesgue-Stieltjes **V:** Diferenciación de medidas y el Teorema Fundamental del Cálculo en este contexto

Objetivos de Aprendizaje **I:** Evaluar la convergencia de sucesiones de funciones bajo la integral de Lebesgue [Assessment] **II:** Utilizar el Teorema de Radon-Nikodym para hallar densidades de probabilidad o medidas [Usage] **III:** Comparar la integral de Riemann con la de Lebesgue en términos de aplicabilidad [Familiarity]

Bibliografía: [Folland, 1999, Brezis, 2011]

5.3 Espacios de Hilbert, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Repaso de producto interno y completitud **II:** Ortogonalidad, bases ortonormales y el proceso de Gram-Schmidt **III:** Teorema de representación de Riesz y funcionales lineales **IV:** Operadores adjuntos, autoadjuntos y unitarios en espacios de Hilbert **V:** Introducción a la teoría espectral de operadores compactos

Objetivos de Aprendizaje **I:** Aplicar el teorema de proyección para resolver problemas de mejor aproximación [Usage] **II:** Identificar el operador adjunto de operadores lineales clásicos [Assessment] **III:** Analizar la estructura espectral de operadores autoadjuntos [Familiarity]

Bibliografía: [Kreyszig, 1989, Conway, 1990]

5.4 Teoría de Operadores, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Operadores acotados y no acotados: definiciones y dominios **II:** El espectro de un operador: espectro puntual, continuo y residual **III:** Operadores normales y el Teorema Espectral para operadores compactos **IV:** Aplicaciones de la teoría de operadores en análisis numérico

Objetivos de Aprendizaje **I:** Calcular el espectro de diversos tipos de operadores lineales [Assessment] **II:** Aplicar el teorema espectral en la descomposición de operadores [Usage] **III:** Resolver problemas operativos utilizando el cálculo funcional básico [Assessment]

Bibliografía: [Brezis, 2011, Yosida, 1980]

6. Bibliografía

[Brezis, 2011] Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer Science & Business Media.

[Conway, 1990] Conway, J. B. (1990). *A Course in Functional Analysis*. Springer-Verlag, 2nd edition.

[Folland, 1999] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. Wiley-Interscience, 2nd edition.

[Kreyszig, 1989] Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. Wiley.

[Royden and Fitzpatrick, 2010] Royden, H. and Fitzpatrick, P. (2010). *Real Analysis*. Pearson, 4th edition.

[Yosida, 1980] Yosida, K. (1980). *Functional Analysis*. Springer-Verlag, 6th edition.

[Brezis, 2011] Brezis, H. (2011). *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer Science & Business Media.

[Conway, 1990] Conway, J. B. (1990). *A Course in Functional Analysis*. Springer-Verlag, 2nd edition.

[Folland, 1999] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. Wiley-Interscience, 2nd edition.

[Kreyszig, 1989] Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. Wiley.

[Royden and Fitzpatrick, 2010] Royden, H. and Fitzpatrick, P. (2010). *Real Analysis*. Pearson, 4th edition.

[Yosida, 1980] Yosida, K. (1980). *Functional Analysis*. Springer-Verlag, 6th edition.

Noveno Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI264 Aprendizaje Profundo (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI263 Introducción al Aprendizaje de Máquina (8vo Sem-Pág 451)

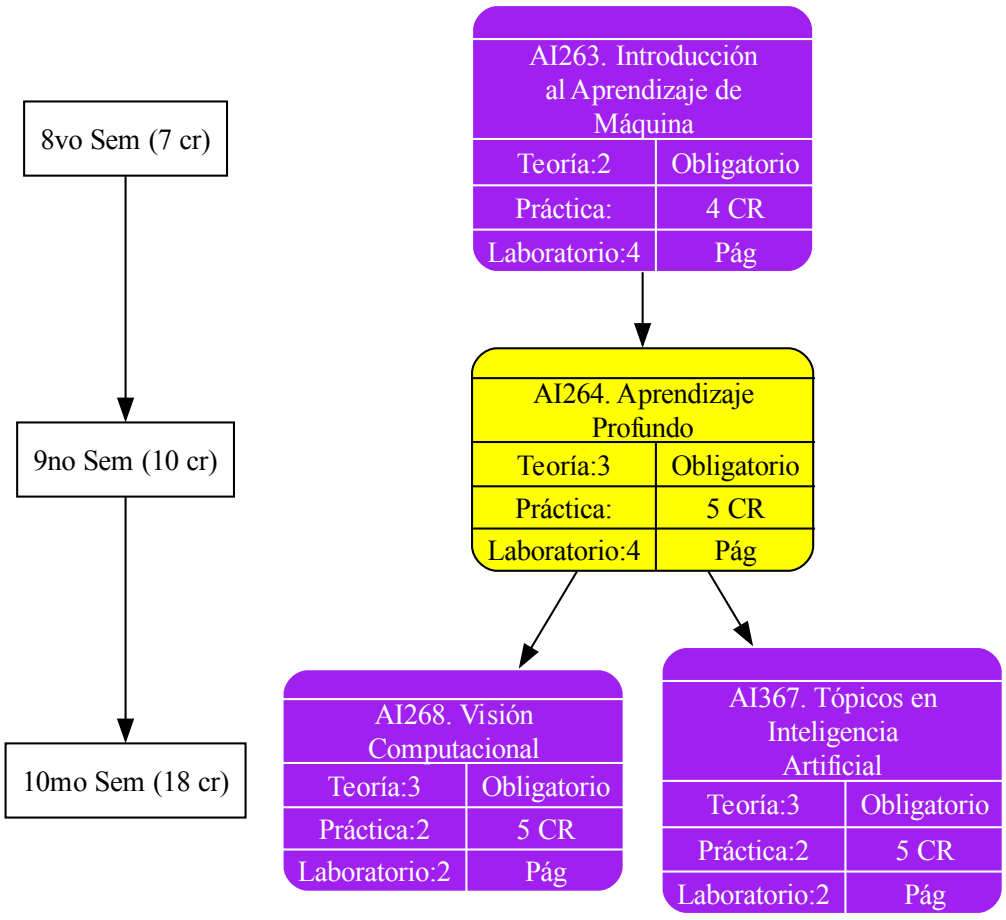


Figura 7.54: Mapa de Conexión. AI264 Aprendizaje Profundo

2. Fundamentación

Este curso cubre los fundamentos del aprendizaje profundo moderno, incluyendo redes neuronales convolucionales, arquitecturas de transformers, y técnicas de entrenamiento para sistemas de inteligencia artificial avanzada.

3. Objetivos del curso.

- Implementar redes neuronales profundas usando frameworks modernos (PyTorch/TensorFlow)
- Diseñar arquitecturas para problemas específicos (visión, lenguaje, etc.)
- Analizar limitaciones éticas de modelos profundos

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Evaluar)

5. Contenido

5.1 Entrenamiento y Optimización de Redes Neuronales, 10 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Backpropagation moderno: grafos computacionales y diferenciación automática **II:** Optimizadores avanzados: momentum, RMSProp, Adam, y programación de la tasa de aprendizaje

Objetivos de Aprendizaje **I:** Implementar una red neuronal profunda desde cero, incluyendo su paso de backpropagation [Aplicar] **II:** Seleccionar y ajustar un optimizador y su programación de tasa de aprendizaje para la estabilidad del entrenamiento [Aplicar]

Bibliografía: [Goodfellow et al., 2016]

5.2 Técnicas de Regularización para Redes Profundas, 5 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Regularización por Dropout **II:** Normalización por lotes (Batch Normalization)

Objetivos de Aprendizaje **I:** Diagnosticar el sobreajuste en un entrenamiento usando diagnósticos de regularización [Analizar] **II:** Aplicar Dropout y Batch Normalization para mejorar la generalización de un modelo [Aplicar]

Bibliografía: [Goodfellow et al., 2016]

5.3 Arquitecturas de Redes Convolucionales y Residuales, 9 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Arquitecturas de redes neuronales convolucionales profundas **II:** Conexiones residuales y ResNets

Objetivos de Aprendizaje **I:** Diseñar una arquitectura CNN profunda para una tarea de visión dada [Aplicar] **II:** Adaptar una arquitectura residual a un dominio de visión específico [Aplicar]

Bibliografía: [He et al., 2016]

5.4 Arquitecturas Transformer y Modelos Generativos, 16 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: **I:** Transformers y el mecanismo de atención **II:** Modelos generativos: autoencoders variacionales (VAEs) y redes generativas adversariales (GANs)

Objetivos de Aprendizaje **I:** Entrenar un modelo transformer en un dataset real [Aplicar] **II:** Analizar y contrastar arquitecturas de modelos generativos VAE y GAN [Analizar]

Bibliografía: [Vaswani et al., 2017]

6. Bibliografía

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *CVPR*, pages 770–778.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *CVPR*, pages 770–778.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA402 Proyecto de final de carrera II (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 8 horas;
- **Prerrequisitos**:
  - MA401 Proyecto de final de carrera I (8vo Sem-Pág 454)
  - ID203 Inglés V (7mo Sem-Pág 448)

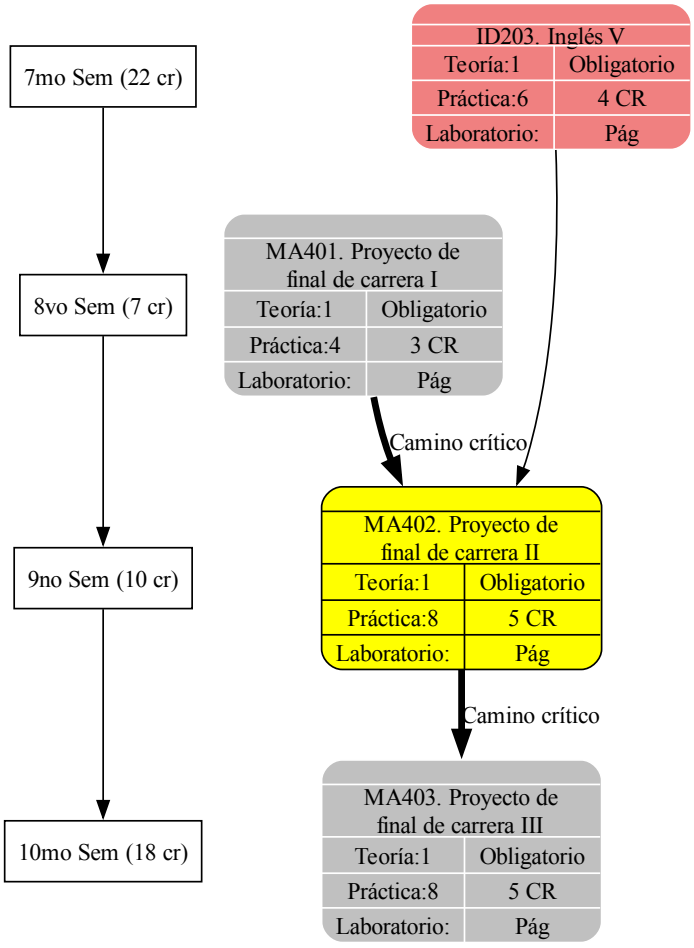


Figura 7.55: Mapa de Conexión. MA402 Proyecto de final de carrera II

2. Fundamentación

Este curso constituye la fase final del proyecto de titulación en matemáticas computacionales, donde los estudiantes completan, validan y presentan un proyecto de investigación aplicada o desarrollo tecnológico. Se enfatiza la implementación completa de soluciones computacionales, el análisis riguroso de resultados, la validación metodológica y la comunicación profesional de hallazgos. El proyecto debe demostrar dominio de las matemáticas computacionales, capacidad de innovación y contribución al

conocimiento o solución de problemas del mundo real, culminando en un trabajo de calidad profesional y una defensa pública ante un jurado especializado.

3. Objetivos del curso.

- Implementar completamente la solución computacional diseñada, aplicando técnicas avanzadas de matemáticas computacionales.
- Realizar análisis exhaustivos de resultados y validaciones metodológicas rigurosas.
- Documentar profesionalmente todo el proceso de investigación y desarrollo del proyecto.
- Evaluar el impacto y las limitaciones de la solución desarrollada, proponiendo mejoras futuras.
- Defender públicamente el proyecto ante un jurado especializado, demostrando dominio del tema y habilidades de comunicación técnica.
- Demostrar capacidad de trabajo autónomo, gestión de proyecto y ética profesional en el desarrollo de investigación aplicada.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Evaluar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Evaluar)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Evaluar)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Evaluar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Evaluar)

5. Contenido

5.1 Implementación Avanzada y Optimización, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Implementación completa de algoritmos y métodos computacionales **II:** Optimización y mejora de rendimiento de soluciones **III:** Validación técnica y pruebas exhaustivas **IV:** Manejo de casos límite y situaciones complejas **V:** Refinamiento iterativo basado en resultados

Objetivos de Aprendizaje **I:** Conducir estudios completos usando métodos de investigación [Evaluar] **II:** Demostrar formalmente propiedades de las soluciones [Evaluar] **III:** Implementar procesos matemáticos complejos en computadora [Usar] **IV:** Optimizar soluciones computacionales para eficiencia [Evaluar]

Bibliografía: [Steve, 2004, E, 1997]

5.2 Análisis de Resultados y Validación, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Análisis estadístico riguroso de resultados **II:** Validación metodológica y comparación con benchmarks **III:** Análisis de sensibilidad y robustez de soluciones **IV:** Evaluación de limitaciones y supuestos del modelo **V:** Interpretación crítica de hallazgos y conclusiones

Objetivos de Aprendizaje **I:** Analizar impacto social de soluciones científicas [Evaluar] **II:** Conducir validaciones metodológicas rigurosas [Evaluar] **III:** Reconocer estructura formal de problemas complejos [Evaluar] **IV:** Interpretar resultados en contexto aplicado [Usar]

Bibliografía: [C, 2017, P et al., 2005]

5.3 Documentación y Comunicación Profesional, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: I: Elaboración de memoria técnica profesional **II:** Preparación de artículos científicos o reportes técnicos **III:** Diseño de presentaciones efectivas para defensa **IV:** Comunicación de resultados a diferentes audiencias **V:** Consideraciones éticas en publicación de resultados

Objetivos de Aprendizaje I: Aplicar principios éticos en comunicación científica [Usar] **II:** Comunicar efectivamente resultados complejos [Evaluar] **III:** Aprender de forma continua mediante retroalimentación [Usar] **IV:** Documentar procesos de forma profesional [Evaluar]

Bibliografía: [Michael, 1996, R and W, 2008]

5.4 Defensa y Evaluación Final, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: I: Preparación para defensa pública del proyecto **II:** Estrategias de presentación y argumentación técnica **III:** Manejo de preguntas y objeciones del jurado **IV:** Evaluación crítica del propio trabajo **V:** Planificación de trabajo futuro y mejoras

Objetivos de Aprendizaje I: Desempeñarse efectivamente en defensa pública [Evaluar] **II:** Gestionar proyectos científicos complejos [Evaluar] **III:** Resolver problemas matemáticos avanzados [Evaluar] **IV:** Responder críticas constructivamente [Usar]

Bibliografía: [L, 2018, H, 2006]

6. Bibliografía

[C, 2017] C, M. D. (2017). *Design and Analysis of Experiments*. John Wiley & Sons, 9th edition.

[E, 1997] E, K. D. (1997). *The Art of Computer Programming*. Addison-Wesley.

[H, 2006] H, A. R. R. (2006). *Dazzle 'em with Style: The Art of Oral Scientific Presentation*. Elsevier Academic Press.

[L, 2018] L, T. K. (2018). *A Manual for Writers of Research Papers, Theses, and Dissertations*. University of Chicago Press.

[Michael, 1996] Michael, A. (1996). *The Craft of Scientific Writing*. Springer.

[P et al., 2005] P, B. G. E., Stuart, H. J., and G, H. W. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience.

[R and W, 2008] R, M. J. and W, M. R. (2008). *Successful Scientific Writing: A Step-by-Step Guide for the Biological and Medical Sciences*. Cambridge University Press.

[Steve, 2004] Steve, M. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

[C, 2017] C, M. D. (2017). *Design and Analysis of Experiments*. John Wiley & Sons, 9th edition.

[E, 1997] E, K. D. (1997). *The Art of Computer Programming*. Addison-Wesley.

[H, 2006] H, A. R. R. (2006). *Dazzle 'em with Style: The Art of Oral Scientific Presentation*. Elsevier Academic Press.

[L, 2018] L, T. K. (2018). *A Manual for Writers of Research Papers, Theses, and Dissertations*. University of Chicago Press.

[Michael, 1996] Michael, A. (1996). *The Craft of Scientific Writing*. Springer.

[P et al., 2005] P, B. G. E., Stuart, H. J., and G, H. W. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience.

[R and W, 2008] R, M. J. and W, M. R. (2008). *Successful Scientific Writing: A Step-by-Step Guide for the Biological and Medical Sciences*. Cambridge University Press.

[Steve, 2004] Steve, M. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS231 Redes y Comunicación (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS2S1 Sistemas Operativos (8vo Sem-Pág 457)

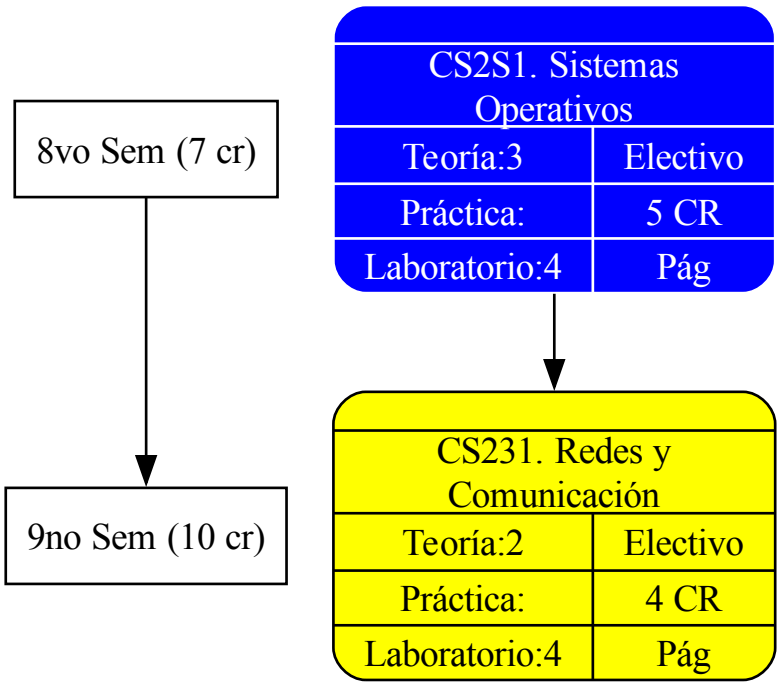


Figura 7.56: Mapa de Conexión. CS231 Redes y Comunicación

2. Fundamentación

Las redes y la comunicación son fundamentales para los sistemas de cómputo modernos, las aplicaciones distribuidas y la Internet global. Este curso ofrece una introducción completa al cuerpo de conocimiento de Redes y Comunicación del CS2023, abarcando desde la transmisión física y los protocolos de enlace de datos, pasando por el transporte confiable y el enrutamiento, hasta los servicios de aplicación, la seguridad, la movilidad y paradigmas emergentes como SDN y redes cuánticas. Los estudiantes desarrollarán comprensión teórica y habilidades prácticas aplicables al diseño y análisis de sistemas en red modernos.

3. Objetivos del curso.

- Comprender la arquitectura en capas de los modelos TCP/IP y OSI, incluyendo el rol y responsabilidades de cada capa.

- Explicar los mecanismos de comunicación de un solo salto, incluyendo protocolos MAC, codificación, Ethernet, WiFi y topologías de red local.
- Dominar los principios de transferencia confiable de datos, control de flujo y control de congestión.
- Analizar paradigmas de enrutamiento, mecanismos de reenvío y escalabilidad de IP en redes locales y de área extensa.
- Comprender los protocolos fundamentales de la capa de aplicación (DNS, HTTP, TCP, UDP) e implementar aplicaciones basadas en sockets.
- Comprender las amenazas de seguridad en redes y las contramedidas para proteger los sistemas en red.
- Describir los principios de las redes móviles, incluyendo redes celulares, LANs inalámbricas y comunicación dispositivo a dispositivo.
- Identificar paradigmas de red emergentes como SDN, virtualización de redes y redes cuánticas.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Usar)

5. Contenido

5.1 Fundamentos de Redes y Comunicaciones, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Importancia de las redes en la informática contemporánea, y desafíos asociados. Ver también: Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad **II:** Organización de Internet (por ejemplo, usuarios, Proveedores de Servicios de Internet, sistemas autónomos, proveedores de contenido, redes de entrega de contenido) **III:** Técnicas de conmutación (por ejemplo, circuito y paquete) **IV:** Capas y sus roles (aplicación, transporte, red, enlace de datos y física) **V:** Principios de estratificación (por ejemplo, encapsulación y modelo de reloj de arena). Ver también: Sistemas Digitales y Lógica, Procesamiento Secuencial y Paralelo, Máquinas de Estado **VI:** Elementos de red (por ejemplo, routers, switches, hubs, puntos de acceso y hosts) **VII:** Conceptos básicos de colas (por ejemplo, relación con latencia, congestión, niveles de servicio, etc.)

Objetivos de Aprendizaje **I:** Articular la organización de Internet [Articular] **II:** Listar y definir la terminología de red apropiada [Listar/Enumerar] **III:** Describir la estructura en capas de una arquitectura de red típica [Describir] **IV:** Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Analizar]

Bibliografía: [Kurose and Ross, 2021, Tanenbaum et al., 2021, Peterson and Davie, 2022]

5.2 Comunicación de un solo salto, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Introducción a modulación, ancho de banda y medios de comunicación **II:** Codificación y Entramado **III:** Control de Acceso al Medio (MAC) (por ejemplo, acceso aleatorio y acceso programado) **IV:** Ethernet y WiFi **V:** Conmutación (por ejemplo, árboles de expansión, VLANs). **VI:** Topologías de Red de área Local (por ejemplo, centro de datos, redes de campus).

Objetivos de Aprendizaje **I:** Describir algunos aspectos básicos de modulación, ancho de banda y medios de comunicación [Describir] **II:** Describir en detalle un protocolo MAC [Describir] **III:** Demostrar comprensión de las compensaciones (trade-offs) de soluciones de codificación y

entramado [Demostrar] **IV:** Describir detalles de la implementación de Ethernet [Describir] **V:** Describir cómo funciona la conmutación [Describir] **VI:** Describir un tipo de topología de LAN [Describir]

Bibliografía: [Tanenbaum et al., 2021, Peterson and Davie, 2022]

5.3 Aplicaciones en Red, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: I: Esquemas de denominación y direccionamiento (por ejemplo, DNS e Identificadores Uniformes de Recursos) **II:** Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación **III:** Diversidad de demandas de aplicaciones en red (por ejemplo, latencia, ancho de banda y tolerancia a pérdidas). Ver también: Comunicación, Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social, Equidad Digital y Accesibilidad, Impacto Social e IA **IV:** Cobertura de protocolos de capa de aplicación (por ejemplo, HTTP) **V:** Interacciones con APIs de TCP, UDP y Socket. Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades

Objetivos de Aprendizaje I: Definir los principios de denominación, direccionamiento y localización de recursos [Definir] **II:** Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar] **III:** Describir los detalles de un protocolo de capa de aplicación [Describir] **IV:** Implementar una aplicación simple cliente-servidor basada en sockets [Implementar]

Bibliografía: [Kurose and Ross, 2021, Peterson and Davie, 2022]

5.4 Soporte de Fiabilidad, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: I: Entrega no fiable (por ejemplo, UDP) **II:** Principios de fiabilidad (por ejemplo, entrega sin pérdidas, duplicación o fuera de orden). Ver también: Confiabilidad del Sistema **III:** Control de errores (por ejemplo, retransmisión, corrección de errores) **IV:** Control de flujo (por ejemplo, parar y esperar, basado en ventana) **V:** Control de congestión (por ejemplo, notificación implícita y explícita de congestión) **VI:** TCP y problemas de rendimiento (por ejemplo, Tahoe, Reno, Vegas, Cubic)

Objetivos de Aprendizaje I: Describir el funcionamiento de protocolos de entrega fiable [Describir] **II:** Listar los factores que afectan el rendimiento de los protocolos de entrega fiable [Listar/Enumerar] **III:** Describir algunos problemas de diseño de fiabilidad de TCP [Describir] **IV:** Diseñar e implementar un protocolo fiable simple [Diseñar]

Bibliografía: [Kurose and Ross, 2021, Tanenbaum et al., 2021]

5.5 Enrutamiento y Reenvío, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: I: Paradigmas y jerarquía de enrutamiento (por ejemplo, intra/inter dominio, centralizado y descentralizado, enrutamiento de origen, circuitos virtuales, QoS) **II:** Métodos de reenvío (por ejemplo, tablas de reenvío y algoritmos de coincidencia) **III:** IP y problemas de escalabilidad (por ejemplo, NAT, CIDR, BGP, diferentes versiones de IP)

Objetivos de Aprendizaje I: Describir varios paradigmas y jerarquías de enrutamiento [Describir] **II:** Describir cómo se reenvían los paquetes en una red IP [Describir] **III:** Describir cómo Internet aborda los desafíos de escalabilidad [Describir]

Bibliografía: [Kurose and Ross, 2021, Tanenbaum et al., 2021, Peterson and Davie, 2022]

5.6 Seguridad de Red, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Introducción general sobre seguridad (Amenazas, vulnerabilidades y contramedidas). Ver también: Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad, Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad **II:** Amenazas específicas de red y tipos de ataque (por ejemplo, denegación de servicio, suplantación, rastreo y redireccionamiento de tráfico, atacante en el medio, ataques de integridad de mensajes, ataques de enrutamiento, ransomware y análisis de tráfico). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad **III:** Contramedidas: *a*) Criptografía (por ejemplo, SSL, TLS, simétrica/asimétrica) *b*) Arquitecturas para redes seguras (por ejemplo, canales seguros, protocolos de enrutamiento seguro, DNS seguro, VPNs, DMZ, Zero Trust Network Access, hiper seguridad de red, protocolos de comunicación anónima, aislamiento) *c*) Monitoreo de red, detección de intrusiones, firewalls, protección contra suplantación y DoS, honeypots, trazabilidad, BGP Sec, RPKI

Objetivos de Aprendizaje **I:** Describir algunos de los modelos de amenaza de seguridad de red [Describir] **II:** Describir contramedidas específicas basadas en red [Describir] **III:** Analizar varios aspectos de seguridad de red a partir de un estudio de caso [Analizar]

Bibliografía: [Stallings, 2022, Kurose and Ross, 2021]

5.7 Soporte de Movilidad, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Principios de comunicación celular (por ejemplo, 4G, 5G) **II:** Principios de LANs inalámbricas (principalmente 802.11) **III:** Comunicación dispositivo a dispositivo (por ejemplo, comunicación IoT) **IV:** Redes inalámbricas multiescalón (por ejemplo, redes ad hoc, oportunistas, tolerantes a retrasos)

Objetivos de Aprendizaje **I:** Describir algunos aspectos de comunicación celular como el registro [Describir] **II:** Describir cómo 802.11 soporta usuarios móviles [Describir] **III:** Describir usos prácticos de comunicación dispositivo a dispositivo, así como multiescalón [Describir] **IV:** Describir un tipo de red móvil como ad hoc [Describir]

Bibliografía: [Tanenbaum et al., 2021, Peterson and Davie, 2022]

5.8 Temas Emergentes, 6 hr(s)

Competencias: ABET-2,ABET-6

Tópicos: **I:** Middleboxes (por ejemplo, avances en el uso de IA, redes basadas en intención, filtrado, inspección profunda de paquetes, balanceo de carga, NAT, CDN) **II:** Virtualización de Red (por ejemplo, SDN, Redes de Centros de Datos) **III:** Redes Cuánticas (por ejemplo, Introducción al dominio, teleportación, seguridad, Internet Cuántica) **IV:** Satélite, mmWave, Luz Visible

Objetivos de Aprendizaje **I:** Describir el valor de los avances en middleboxes en redes [Describir] **II:** Describir la importancia de las Redes Definidas por Software [Describir] **III:** Describir algunos de los valores añadidos logrados al usar Redes Cuánticas [Describir]

Bibliografía: [Peterson and Davie, 2022, Tanenbaum et al., 2021]

6. Bibliografía

[Kurose and Ross, 2021] Kurose, J. and Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson, 8th edition.

[Peterson and Davie, 2022] Peterson, L. L. and Davie, B. S. (2022). *Computer Networks: A Systems Approach*. Morgan Kaufmann, 6th edition.

[Stallings, 2022] Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice*. Pearson, 8th edition.

[Tanenbaum et al., 2021] Tanenbaum, A. S., Feamster, N., and Wetherall, D. J. (2021). *Computer Networks*. Pearson, 6th edition.

[Kurose and Ross, 2021] Kurose, J. and Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson, 8th edition.

[Peterson and Davie, 2022] Peterson, L. L. and Davie, B. S. (2022). *Computer Networks: A Systems Approach*. Morgan Kaufmann, 6th edition.

[Stallings, 2022] Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice*. Pearson, 8th edition.

[Tanenbaum et al., 2021] Tanenbaum, A. S., Feamster, N., and Wetherall, D. J. (2021). *Computer Networks*. Pearson, 6th edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA381 Estructuras Algebraicas (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA281 Teoría de números (5to Sem-Pág 395)

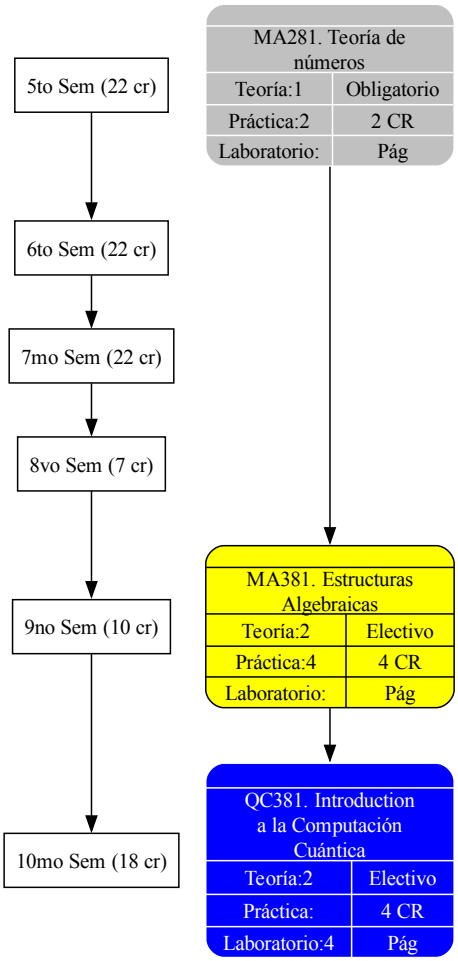


Figura 7.57: Mapa de Conexión. MA381 Estructuras Algebraicas

2. Fundamentación

El curso de Estructuras Algebraicas proporciona la base formal y abstracta esencial para las matemáticas avanzadas y sus aplicaciones computacionales. Este curso introduce las estructuras fundamentales del álgebra moderna —grupos, anillos y campos— desarrollando la capacidad de pensar en términos de axiomas, homomorfismos e invariantes. El dominio de estos conceptos es crucial para comprender los fundamentos teóricos de áreas como la teoría de códigos, la criptografía, el diseño de algoritmos y la geometría algebraica. El curso enfatiza la interacción entre la teoría abstracta y los

ejemplos concretos, particularmente aquellos que surgen en matemáticas discretas y computacionales.

3. Objetivos del curso.

- Dominar las definiciones axiomáticas y las propiedades básicas de grupos, anillos y campos.
- Comprender y aplicar los conceptos de subgrupos, subgrupos normales, grupos cociente y los teoremas fundamentales de homomorfismos para grupos.
- Analizar la estructura de los anillos, incluyendo ideales, anillos cociente, dominios de integridad y campos.
- Aplicar la teoría de campos finitos a problemas prácticos en teoría de códigos y criptografía.
- Desarrollar la capacidad de construir demostraciones rigurosas y reconocer estructuras algebraicas en diversos contextos matemáticos y computacionales.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Grupos y Subgrupos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Operaciones binarias y definición de grupo. Ejemplos básicos: grupos cíclicos, simétricos, diédricos y de matrices. **II:** Propiedades elementales de grupos: unicidad del elemento neutro y de los inversos, leyes de cancelación. **III:** Subgrupos y pruebas de subgrupo. Centralizador, normalizador y centro de un grupo. **IV:** Grupos cíclicos: clasificación, estructura y subgrupos. **V:** Grupos de permutaciones: notación cíclica, transposiciones y paridad.

Objetivos de Aprendizaje **I:** Enunciar los axiomas de grupo y verificar si un conjunto dado con una operación forma un grupo [Familiarizarse]. **II:** Aplicar las pruebas de subgrupo y generar subgrupos, incluyendo los subgrupos cíclicos generados por un elemento [Usar]. **III:** Analizar la estructura de grupos cíclicos finitos y descomponer permutaciones en ciclos disjuntos [Evaluar].

Bibliografía: [Dummit and Foote, 2004, Gallian, 2021]

5.2 Clases Laterales, Subgrupos Normales y Homomorfismos, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Clases laterales y Teorema de Lagrange. Aplicaciones al orden de los elementos. **II:** Subgrupos normales y su caracterización. Grupos cociente (o factores). **III:** Homomorfismos e isomorfismos de grupos: definiciones, propiedades y ejemplos. **IV:** Teorema Fundamental del Homomorfismo (Primer Teorema de Isomorfía) para grupos. **V:** Introducción a las acciones de grupo: definición y ejemplos básicos (órbita-estabilizador).

Objetivos de Aprendizaje **I:** Calcular clases laterales y usar el Teorema de Lagrange para deducir propiedades de grupos finitos [Familiarizarse]. **II:** Determinar si un subgrupo es normal y construir el correspondiente grupo cociente [Usar]. **III:** Utilizar el Teorema Fundamental del Homomorfismo para establecer isomorfismos entre grupos cociente y grupos conocidos [Evaluar].

Bibliografía: [Dummit and Foote, 2004, Judson, 2021]

5.3 Anillos, Ideales y Dominios de Integridad, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Definición y ejemplos básicos de anillos, anillos conmutativos y anillos con unidad. **II:** Homomorfismos e isomorfismos de anillos, propiedades básicas. **III:** Ideales y anillos cociente. Teorema Fundamental del Homomorfismo para anillos. **IV:** Operaciones con ideales: suma, producto e intersección. **V:** Dominios de integridad, campos y característica de un anillo. Los dominios de integridad finitos son campos.

Objetivos de Aprendizaje **I:** Identificar las propiedades definitorias de anillos, dominios de integridad y campos a partir de ejemplos dados [Familiarizarse]. **II:** Verificar si un subconjunto de un anillo es un ideal y construir el correspondiente anillo cociente [Usar]. **III:** Aplicar los teoremas de homomorfismo de anillos para analizar estructuras y demostrar resultados básicos sobre dominios de integridad [Evaluar].

Bibliografía: [Dummit and Foote, 2004, Hungerford, 2014]

5.4 Campos y Aplicaciones, 16 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Extensiones de campos: elementos algebraicos y trascendentes. Grado de una extensión. **II:** Campos finitos (campos de Galois): existencia, unicidad y estructura. El campo $GF(p^n)$. **III:** Anillos de polinomios sobre un campo. Polinomios irreducibles y construcción de campos finitos. **IV:** Elementos primitivos en campos finitos y su aplicación en criptografía (ej: Diffie-Hellman, AES). **V:** Breve introducción a códigos correctores de errores: distancia de Hamming y códigos lineales como espacios vectoriales sobre campos finitos.

Objetivos de Aprendizaje **I:** Describir la clasificación y propiedades básicas de los campos finitos [Familiarizarse]. **II:** Construir campos finitos pequeños utilizando polinomios irreducibles sobre $\mathbb{Z}_p$ [Usar]. **III:** Explicar cómo la estructura de los campos finitos sustenta protocolos criptográficos modernos y la teoría de códigos [Evaluar].

Bibliografía: [Gallian, 2021, Lidl and Niederreiter, 1994]

6. Bibliografía

[Dummit and Foote, 2004] Dummit, D. S. and Foote, R. M. (2004). *Abstract Algebra*. John Wiley & Sons, 3rd edition. Comprehensive and widely-used graduate-level textbook, thorough and rigorous.

[Gallian, 2021] Gallian, J. A. (2021). *Contemporary Abstract Algebra*. CRC Press, 10th edition. Very accessible undergraduate text with numerous applications and historical notes.

[Hungerford, 2014] Hungerford, T. W. (2014). *Abstract Algebra: An Introduction*. Cengage Learning, 3rd edition. Clear introduction focusing on understanding core concepts, suitable for first exposure.

[Judson, 2021] Judson, T. W. (2021). *Abstract Algebra: Theory and Applications*. Orthogonal Publishing, 2021a edition. Open-source textbook with an applied focus, including chapters on cryptography and coding theory. Freely available online.

[Lidl and Niederreiter, 1994] Lidl, R. and Niederreiter, H. (1994). *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 2nd edition. Authoritative reference on the theory and computational aspects of finite fields.

[Dummit and Foote, 2004] Dummit, D. S. and Foote, R. M. (2004). *Abstract Algebra*. John Wiley & Sons, 3rd edition. Comprehensive and widely-used graduate-level textbook, thorough and rigorous.

[Gallian, 2021] Gallian, J. A. (2021). *Contemporary Abstract Algebra*. CRC Press, 10th edition. Very accessible undergraduate text with numerous applications and historical notes.

[Hungerford, 2014] Hungerford, T. W. (2014). *Abstract Algebra: An Introduction*. Cengage Learning, 3rd edition. Clear introduction focusing on understanding core concepts, suitable for first exposure.

[Judson, 2021] Judson, T. W. (2021). *Abstract Algebra: Theory and Applications*. Orthogonal Publishing, 2021a edition. Open-source textbook with an applied focus, including chapters on cryptography and coding theory. Freely available online.

[Lidl and Niederreiter, 1994] Lidl, R. and Niederreiter, H. (1994). *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 2nd edition. Authoritative reference on the theory and computational aspects of finite fields.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI262 Procesamiento de Lenguaje Natural (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI263 Introducción al Aprendizaje de Máquina (8vo Sem-Pág 451)

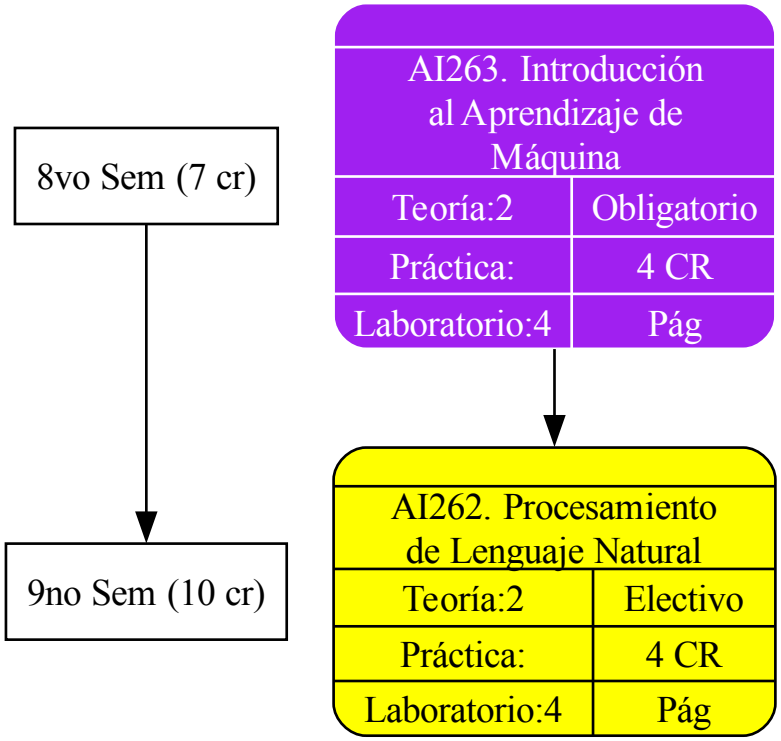


Figura 7.58: Mapa de Conexión. AI262 Procesamiento de Lenguaje Natural

2. Fundamentación

Este curso introduce los fundamentos teóricos y prácticos del Procesamiento de Lenguaje Natural (NLP), cubriendo desde técnicas clásicas hasta modelos modernos basados en deep learning. Los estudiantes aprenderán a implementar pipelines completos de NLP para aplicaciones como análisis de sentimientos, traducción automática y chatbots.

3. Objetivos del curso.

- Comprender la representación computacional del lenguaje humano.
- Implementar algoritmos para procesamiento de texto en Python.
- Aplicar modelos de NLP a problemas reales usando frameworks modernos.

4. Competencias Generales (*Outcomes*)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Fundamentos de NLP, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Tokenización y representación de texto **II:** Modelos de n-gramas y suavizado **III:** Extracción de características (TF-IDF, word embeddings)

Objetivos de Aprendizaje I: Preprocesar texto para modelos de NLP [Usar] **II:** Explicar limitaciones de modelos estadísticos clásicos [Familiarizarse]

Bibliografía: [Jurafsky and Martin, 2021, Manning and Schütze, 1999]

5.2 Modelado de Lenguaje, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Word2Vec, GloVe y FastText **II:** Redes neuronales recurrentes (RNNs) para NLP **III:** Modelos de secuencia a secuencia (seq2seq)

Objetivos de Aprendizaje I: Entrenar embeddings de palabras [Usar] **II:** Implementar RNNs básicas para generación de texto [Evaluar]

Bibliografía: [Goldberg, 2017, Mikolov et al., 2013]

5.3 Transformers y NLP Moderno, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Arquitectura Transformer (atención auto-supervisada) **II:** BERT y modelos pre-entrenados **III:** Fine-tuning para tareas específicas

Objetivos de Aprendizaje I: Aplicar modelos pre-entrenados con HuggingFace [Usar] **II:** Evaluar métricas de desempeño en NLP [Evaluar]

Bibliografía: [Vaswani et al., 2017, Devlin et al., 2019]

5.4 Aplicaciones Prácticas, 15 hr(s)

Competencias: ABET-1,ABET-2

Tópicos: I: Análisis de sentimientos con transformers **II:** Sistemas de pregunta-respuesta **III:** Despliegue de modelos con Flask/FastAPI

Objetivos de Aprendizaje I: Construir un pipeline completo de NLP [Usar] **II:** Optimizar modelos para producción [Evaluar]

Bibliografía: [Jurafsky and Martin, 2021, Chollet, 2021]

6. Bibliografía

[Chollet, 2021] Chollet, F. (2021). *Deep Learning with Python*. Manning, 2nd edition.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

[Goldberg, 2017] Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool.

[Jurafsky and Martin, 2021] Jurafsky, D. and Martin, J. H. (2021). *Speech and Language Processing*. Pearson, 3rd edition.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Chollet, 2021] Chollet, F. (2021). *Deep Learning with Python*. Manning, 2nd edition.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

[Goldberg, 2017] Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool.

[Jurafsky and Martin, 2021] Jurafsky, D. and Martin, J. H. (2021). *Speech and Language Processing*. Pearson, 3rd edition.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS273 Análisis de series temporales (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA351 Estadística y Probabilidades (6to Sem-Pág 408)

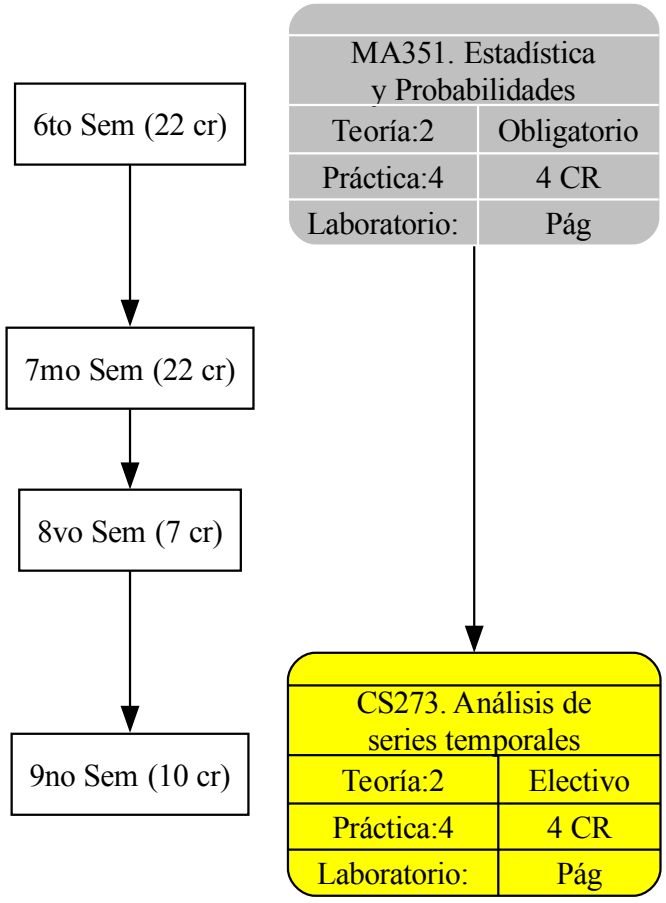


Figura 7.59: Mapa de Conexión. CS273 Análisis de series temporales

2. Fundamentación

En diversas áreas como las finanzas, la meteorología y la ingeniería, los datos se recolectan secuencialmente a través del tiempo, presentando dependencias temporales únicas. Este curso introduce los fundamentos teóricos y prácticos para analizar, modelar y pronosticar series de tiempo. Se abordan desde métodos clásicos de descomposición hasta modelos estocásticos avanzados (ARIMA, GARCH), permitiendo al estudiante extraer información relevante de procesos dinámicos y tomar decisiones basadas en predicciones rigurosas.

3. Objetivos del curso.

- Comprender los componentes fundamentales de una serie de tiempo (tendencia, estacionalidad, ciclo y residuo).
- Identificar y transformar series para alcanzar la estacionariedad.
- Modelar procesos estocásticos utilizando metodologías de Box-Jenkins.
- Evaluar la calidad de los pronósticos mediante métricas de error y validación cruzada temporal.
- Implementar modelos predictivos avanzados para series con volatilidad o dependencias complejas.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Introducción y Conceptos Fundamentales, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Definición y ejemplos de series temporales. **II:** Componentes de una serie: Tendencia, Estacionalidad y Ruido. **III:** Operadores de retardo (Lag) y diferencia. **IV:** Operaciones de suavizamiento: Promedios móviles simples y ponderados.

Objetivos de Aprendizaje I: Descomponer una serie temporal en sus componentes básicos [Usar]. **II:** Identificar patrones visuales de dependencia temporal en datos reales [Familiarizarse].

Bibliografía: [Hyndman and Athanasopoulos, 2021, Box et al., 2015]

5.2 Estacionariedad y Funciones de Correlación, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Estacionariedad débil y fuerte. **II:** Función de Autocorrelación (ACF) y Autocorrelación Parcial (PACF). **III:** Ruido blanco y Caminata Aleatoria (Random Walk). **IV:** Pruebas de raíz unidad (Dicky-Fuller aumentada).

Objetivos de Aprendizaje I: Determinar si una serie es estacionaria mediante pruebas estadísticas [Usar]. **II:** Interpretar correlogramas para identificar posibles modelos estocásticos [Evaluar].

Bibliografía: [Shumway and Stoffer, 2017, Box et al., 2015]

5.3 Modelos de Procesos Lineales (AR y MA), 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Procesos Autoregresivos (AR). **II:** Procesos de Media Móvil (MA). **III:** Invertibilidad y condiciones de estabilidad. **IV:** Estimación de parámetros y criterios de información (AIC, BIC).

Objetivos de Aprendizaje I: Modelar dependencias lineales en series de tiempo estacionarias [Usar]. **II:** Seleccionar el orden óptimo de un modelo basado en criterios estadísticos [Evaluar].

Bibliografía: [Shumway and Stoffer, 2017, Brockwell and Davis, 2016]

5.4 Metodología Box-Jenkins y Modelos ARIMA, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Modelos Integrados (ARIMA). **II:** Identificación, estimación y verificación de diagnósticos. **III:** Análisis de residuos y prueba de Ljung-Box. **IV:** Modelos estacionales (SARIMA).

Objetivos de Aprendizaje I: Implementar el ciclo completo de modelado de Box-Jenkins [Usar]. **II:** Validar la adecuación de un modelo mediante el análisis de sus residuos [Evaluar].

Bibliografía: [Box et al., 2015, Hyndman and Athanasopoulos, 2021]

5.5 Pronóstico y Evaluación de Modelos, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Pronósticos puntuales y por intervalos de confianza. **II:** Métricas de error: MAE, RMSE, MAPE y MASE. **III:** Validación cruzada para series temporales (Time Series Split). **IV:** Combinación de pronósticos.

Objetivos de Aprendizaje I: Generar predicciones futuras con sus respectivos márgenes de incertidumbre [Usar]. **II:** Comparar el desempeño de múltiples modelos predictivos de forma rigurosa [Evaluar].

Bibliografía: [Hyndman and Athanasopoulos, 2021, Brockwell and Davis, 2016]

5.6 Modelos de Volatilidad (ARCH y GARCH), 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Heterocedasticidad condicional en series financieras. **II:** Modelos ARCH (Autoregressive Conditional Heteroskedasticity). **III:** Modelos GARCH y sus extensiones. **IV:** Estimación de riesgo y valor en riesgo (VaR).

Objetivos de Aprendizaje I: Identificar la presencia de volatilidad agrupada en datos financieros [Usar]. **II:** Modelar la varianza condicional de procesos dinámicos complejos [Evaluar].

Bibliografía: [Tsay, 2010, Shumway and Stoffer, 2017]

5.7 Ética y Responsabilidad en el Pronóstico, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Consecuencias socioeconómicas de pronósticos erróneos. **II:** Transparencia en la comunicación de incertidumbre. **III:** Uso ético de datos temporales sensibles y privacidad. **IV:** Sesgos en algoritmos de predicción automatizada.

Objetivos de Aprendizaje I: Reconocer el impacto ético de las decisiones basadas en predicciones [Familiarizarse]. **II:** Comunicar limitaciones y riesgos de los modelos a los interesados [Usar].

Bibliografía: [Hyndman and Athanasopoulos, 2021]

5.8 Implementación y Aplicaciones Reales, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Herramientas computacionales (R: ‘forecast‘, ‘fable‘; Python: ‘statsmodels‘). **II:** Automatización de flujos de trabajo de pronóstico. **III:** Aplicaciones en demanda de energía, inventarios y mercados. **IV:** Presentación de un proyecto final de predicción.

Objetivos de Aprendizaje I: Desarrollar sistemas de pronóstico automatizados utilizando software especializado [Evaluar]. **II:** Resolver un problema de predicción real integrando todas las etapas del curso [Evaluar].

Bibliografía: [Tsay, 2010, Hyndman and Athanasopoulos, 2021]

6. Bibliografía

[Box et al., 2015] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. Wiley, 5th edition.

[Brockwell and Davis, 2016] Brockwell, P. J. and Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer, 3rd edition.

[Hyndman and Athanasopoulos, 2021] Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts.

[Shumway and Stoffer, 2017] Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, 4th edition.

[Tsay, 2010] Tsay, R. S. (2010). *Analysis of Financial Time Series*. Wiley, 3rd edition.

[Box et al., 2015] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. Wiley, 5th edition.

[Brockwell and Davis, 2016] Brockwell, P. J. and Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer, 3rd edition.

[Hyndman and Athanasopoulos, 2021] Hyndman, R. J. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts.

[Shumway and Stoffer, 2017] Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, 4th edition.

[Tsay, 2010] Tsay, R. S. (2010). *Analysis of Financial Time Series*. Wiley, 3rd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA341 Optimización I (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - MA221 Análisis Numérico I (5to Sem-Pág 387)

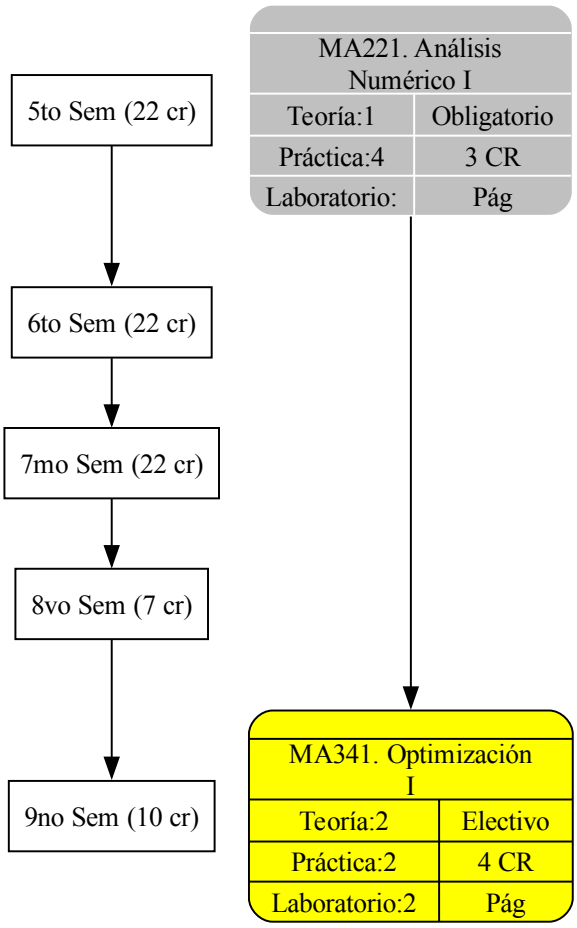


Figura 7.60: Mapa de Conexión. MA341 Optimización I

2. Fundamentación

Este curso introduce los fundamentos matemáticos de la optimización, cubriendo tanto problemas sin restricciones como con restricciones. Se estudian métodos clásicos de optimización continua, condiciones de optimalidad, algoritmos numéricos y aplicaciones en diversos campos. El curso proporciona las bases teóricas y computacionales para formular y resolver problemas de optimización en ciencia, ingeniería y economía, desarrollando habilidades analíticas para caracterizar y encontrar soluciones óptimas.

3. Objetivos del curso.

- Comprender los conceptos fundamentales de optimización y clasificar diferentes tipos de problemas de optimización.
- Aplicar condiciones de optimalidad para caracterizar soluciones en problemas de optimización continua.
- Implementar algoritmos numéricos para resolver problemas de optimización sin restricciones y con restricciones.
- Analizar la convergencia y eficiencia de diferentes métodos de optimización.
- Formular y resolver problemas de optimización del mundo real en diversos contextos aplicados.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Optimización Convexa y No Convexa, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Conjuntos y funciones convexas **II:** Condiciones de optimalidad **III:** Métodos de gradiente para optimización convexa **IV:** Desafíos de optimización no convexa **V:** Análisis de convergencia

Objetivos de Aprendizaje **I:** Caracterizar problemas de optimización convexa [Evaluar] **II:** Aplicar condiciones de optimalidad para verificar soluciones [Usar] **III:** Implementar métodos basados en gradiente para problemas convexos [Usar] **IV:** Analizar desafíos en optimización no convexa [Evaluar]

Bibliografía: [Nocedal and Wright, 2006, Boyd and Vandenberghe, 2004]

5.2 Optimización con Restricciones, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Programación lineal (método simplex) **II:** Programación entera **III:** Multiplicadores de Lagrange y condiciones KKT **IV:** Métodos de penalización y barrera **V:** Teoría de dualidad

Objetivos de Aprendizaje **I:** Resolver problemas de programación lineal [Usar] **II:** Formular y resolver problemas de programación entera [Usar] **III:** Aplicar condiciones KKT a problemas con restricciones [Usar] **IV:** Usar métodos de penalización para optimización con restricciones [Usar]

Bibliografía: [Nocedal and Wright, 2006, Boyd and Vandenberghe, 2004]

5.3 Optimización para Aprendizaje Automático, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Descenso de gradiente estocástico **II:** Momentum y métodos adaptativos (Adam, RMSProp) **III:** Regularización y dispersidad **IV:** Optimización distribuida para AA **V:** Métodos de segundo orden para AA

Objetivos de Aprendizaje I: Implementar variantes de descenso de gradiente estocástico [Usar] **II:** Aplicar métodos de optimización adaptativos [Usar] **III:** Usar regularización para generalización de modelos [Usar] **IV:** Diseñar estrategias de optimización distribuidas [Usar]

Bibliografía: [Nocedal and Wright, 2006, Bertsekas, 2016]

5.4 Metaheurísticas y Optimización Bioinspirada, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Algoritmos genéticos y estrategias evolutivas **II:** Inteligencia de enjambre (PSO, colonia de hormigas) **III:** Temple simulado **IV:** Búsqueda tabú **V:** Metaheurísticas híbridas

Objetivos de Aprendizaje I: Implementar algoritmos genéticos [Usar] **II:** Aplicar métodos de inteligencia de enjambre [Usar] **III:** Usar temple simulado para optimización global [Usar] **IV:** Diseñar enfoques metaheurísticos híbridos [Usar]

Bibliografía: [Talbi, 2009, Mirjalili, 2019]

6. Bibliografía

[Bertsekas, 2016] Bertsekas, D. P. (2016). *Nonlinear Programming*. Athena Scientific, 3rd edition.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, 1st edition.

[Mirjalili, 2019] Mirjalili, S. (2019). *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Springer, 1st edition.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition.

[Talbi, 2009] Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley, 1st edition.

[Bertsekas, 2016] Bertsekas, D. P. (2016). *Nonlinear Programming*. Athena Scientific, 3rd edition.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, 1st edition.

[Mirjalili, 2019] Mirjalili, S. (2019). *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Springer, 1st edition.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition.

[Talbi, 2009] Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. Wiley, 1st edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA355 Teoría de Aprendizaje Estadístico (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - MA352 Procesos estocásticos (8vo Sem-Pág 486)
  - MA372 Análisis funcional II (8vo Sem-Pág 489)

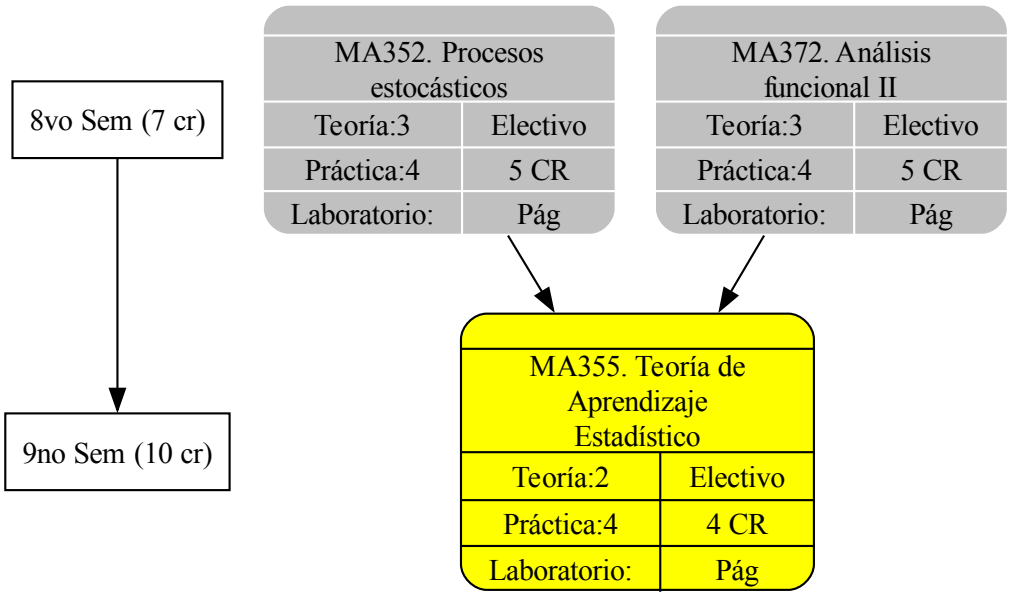


Figura 7.61: Mapa de Conexión. MA355 Teoría de Aprendizaje Estadístico

2. Fundamentación

Este curso proporciona los fundamentos teóricos del aprendizaje estadístico, cubriendo los principios matemáticos que sustentan los algoritmos de aprendizaje automático. Se enfoca en el análisis teórico de la capacidad de generalización, complejidad de modelos, y garantías de rendimiento para diversos métodos de aprendizaje. El curso conecta la teoría con la práctica a través del estudio de cotas de generalización, compensación sesgo-varianza, y fundamentos de la teoría de aprendizaje computacional.

3. Objetivos del curso.

- Comprender los fundamentos de la teoría de aprendizaje estadístico y sus bases probabilísticas.
- Analizar la compensación sesgo-varianza en la selección y evaluación de modelos de aprendizaje automático.
- Aplicar la teoría de Vapnik-Chervonenkis (VC) para analizar la complejidad y capacidad de generalización de modelos.
- Desarrollar y demostrar cotas de generalización para diferentes algoritmos de aprendizaje.
- Conectar los principios teóricos con la implementación práctica de algoritmos de aprendizaje estadístico.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Fundamentos del Aprendizaje Estadístico, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Compensación sesgo-varianza **II:** Teoría de Vapnik-Chervonenkis (VC) **III:** Minimización de riesgo empírico **IV:** Cotass de generalización **V:** Fundamentos de teoría de aprendizaje

Objetivos de Aprendizaje **I:** Analizar la compensación sesgo-varianza en selección de modelos [Evaluar] **II:** Aplicar teoría VC para analizar complejidad de modelos [Usar] **III:** Implementar minimización de riesgo empírico [Usar] **IV:** Interpretar cotas de generalización [Familiarizarse]

Bibliografía: [Vapnik, 1998, Hastie et al., 2009]

5.2 Métodos de Aprendizaje Supervisado, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Modelos lineales y extensiones **II:** Máquinas de Vectores Soporte (SVM) **III:** árboles de decisión y métodos de ensamblado **IV:** Métodos de kernel **V:** Aprendizaje supervisado probabilístico

Objetivos de Aprendizaje **I:** Implementar y analizar modelos lineales [Usar] **II:** Aplicar SVM para clasificación y regresión [Usar] **III:** Construir métodos de ensamblado (bagging, boosting) [Usar] **IV:** Usar métodos de kernel para problemas no lineales [Usar]

Bibliografía: [Hastie et al., 2009, James et al., 2013]

5.3 Métodos de Aprendizaje No Supervisado, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Algoritmos de clustering (k-means, jerárquico) **II:** Reducción de dimensionalidad (PCA, t-SNE) **III:** Estimación de densidad **IV:** Aprendizaje de variedades **V:** Autoencoders y aprendizaje profundo no supervisado

Objetivos de Aprendizaje **I:** Aplicar algoritmos de clustering para exploración de datos [Usar] **II:** Implementar PCA y otros métodos de reducción de dimensionalidad [Usar] **III:** Usar estimación de densidad para aprendizaje no supervisado [Usar] **IV:** Analizar técnicas de aprendizaje de variedades [Evaluar]

Bibliografía: [Murphy, 2012, Bishop, 2006]

5.4 Evaluación y Selección de Modelos, 20 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Estrategias de validación cruzada **II:** Métricas de rendimiento para clasificación y regresión **III:** Comparación de modelos y pruebas estadísticas **IV:** Optimización de hiperparámetros **V:** Curvas de aprendizaje y herramientas de diagnóstico

Objetivos de Aprendizaje I: Implementar validación cruzada para evaluación de modelos [Usar] **II:** Seleccionar métricas de rendimiento apropiadas [Evaluar] **III:** Comparar modelos usando pruebas estadísticas [Usar] **IV:** Optimizar hiperparámetros usando métodos sistemáticos [Usar]

Bibliografía: [Hastie et al., 2009, James et al., 2013]

6. Bibliografía

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, 1st edition.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition.

[James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer, 1st edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, 1st edition.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, 1st edition.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, 1st edition.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition.

[James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer, 1st edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, 1st edition.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, 1st edition.

Décimo Semestre

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS400 Prácticas Pre-profesionales (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - 110Cr

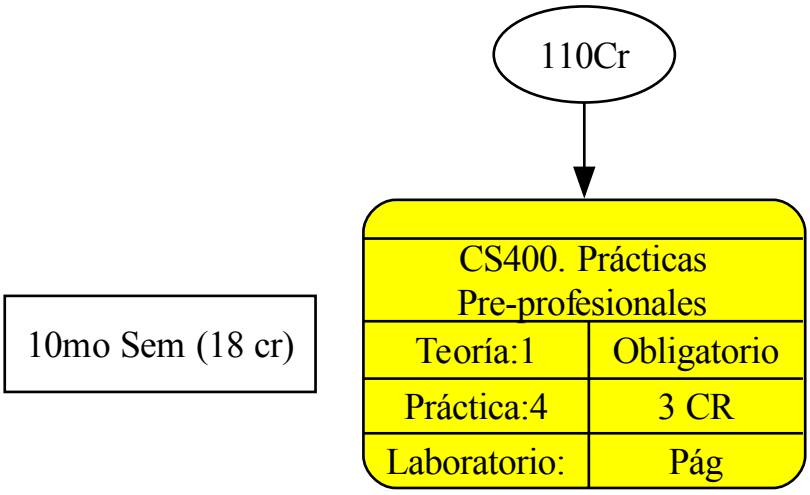


Figura 7.62: Mapa de Conexión. CS400 Prácticas Pre-profesionales

2. Fundamentación

Este curso permite a los estudiantes aplicar los conocimientos adquiridos durante su formación académica en un entorno laboral real, bajo la supervisión de una empresa y la universidad. Las prácticas pre-profesionales son esenciales para desarrollar competencias profesionales, éticas y técnicas, asegurando que el estudiante adquiera experiencia en proyectos relacionados con la carrera.

3. Objetivos del curso.

- Evaluar el desempeño del estudiante en un entorno laboral real, aplicando conocimientos de Ciencia de la Computación.
- Desarrollar habilidades profesionales, éticas y de trabajo en equipo en un contexto empresarial.
- Garantizar que las actividades realizadas estén alineadas con los objetivos de la carrera.

4. Competencias Generales (*Outcomes*)

5. Contenido

5.1 Evaluación de Desempeño en Entorno Laboral, 48 hr(s)

Competencias:

Tópicos: I: Integración del estudiante en un equipo de trabajo profesional. **II:** Aplicación de conocimientos técnicos en proyectos reales. **III:** Evaluación de competencias éticas y profesionales en el ámbito laboral. **IV:** Presentación de informes y retroalimentación por parte de la empresa y la universidad.

Objetivos de Aprendizaje I: Demostrar habilidades técnicas en un entorno laboral real [Evaluar]. **II:** Aplicar principios éticos y profesionales en sus actividades [Evaluar]. **III:** Trabajar efectivamente en equipo y comunicarse de manera profesional [Evaluar].

Bibliografía: [de Educación de la ACM, 2022, Society, 2021]

6. Bibliografía

[de Educación de la ACM, 2022] de Educación de la ACM, C. (2022). Guía para prácticas preprofesionales exitosas en ciencia de la computación. Technical report, Association for Computing Machinery. Documento oficial con estándares para prácticas en computación.

[Society, 2021] Society, I. C. (2021). *Prácticas Profesionales en Computación: Ética y Gestión*. IEEE Press. Manual de buenas prácticas para entornos laborales en TI.

[de Educación de la ACM, 2022] de Educación de la ACM, C. (2022). Guía para prácticas preprofesionales exitosas en ciencia de la computación. Technical report, Association for Computing Machinery. Documento oficial con estándares para prácticas en computación.

[Society, 2021] Society, I. C. (2021). *Prácticas Profesionales en Computación: Ética y Gestión*. IEEE Press. Manual de buenas prácticas para entornos laborales en TI.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI268 Visión Computacional (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (9no Sem-Pág 492)

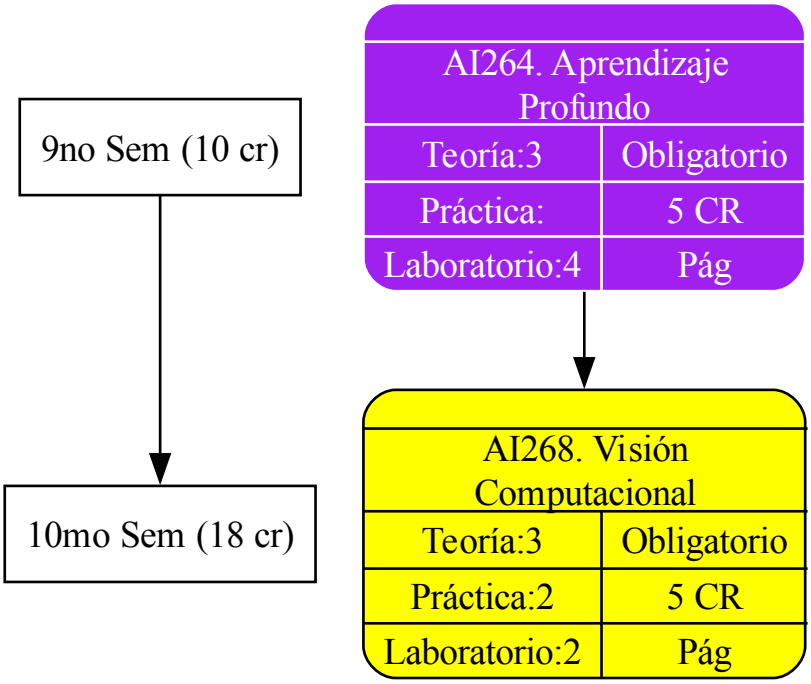


Figura 7.63: Mapa de Conexión. AI268 Visión Computacional

2. Fundamentación

La Visión por Computadora es un área clave de la Inteligencia Artificial que permite a las máquinas interpretar y comprender el mundo visual. Este curso cubre técnicas fundamentales para procesamiento de imágenes, extracción de características y reconocimiento de objetos. Con el advenimiento del aprendizaje profundo, la visión por computadora se ha vuelto esencial para vehículos autónomos, imágenes médicas y realidad aumentada. Los estudiantes aprenderán los fundamentos matemáticos y enfoques algorítmicos para resolver problemas de percepción visual del mundo real.

3. Objetivos del curso.

- Comprender los fundamentos de la formación de imágenes y la representación digital.
- Dominar técnicas de procesamiento de imágenes como filtrado y detección de bordes.
- Aplicar métodos de extracción de características para identificar patrones clave en imágenes.
- Implementar detección y reconocimiento de objetos usando modelos de aprendizaje profundo.
- Analizar técnicas de visión 3D y estimación de movimiento.

4. Competencias Generales (*Outcomes*)

5. Contenido

5.1 Introducción a la IA: Conceptos y Comportamiento Inteligente, 5 hr(s)

Competencias:

Tópicos: **I:** Visión general de los problemas de IA, Ejemplos de aplicaciones recientes exitosas de IA **II:** Definiciones de agentes con ejemplos (por ejemplo, reactivos, deliberativos) **III:** ¿Qué es el comportamiento inteligente?: *a*) La prueba de Turing y sus defectos *b*) Entrada y salida multimodal *c*) Simulación del comportamiento inteligente *d*) Razonamiento racional versus no racional **IV:** Visión general de las Aplicaciones de IA, crecimiento e impacto (económico, social, ético) **V:** Problemas filosóficos **VI:** Historia de la IA

Objetivos de Aprendizaje **I:** Describir la prueba de Turing y el experimento mental de la "Habitación China" [Explicar] **II:** Diferenciar entre razonamiento/comportamiento óptimo y razonamiento/comportamiento similar al humano [Evaluar] **III:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Explicar]

Bibliografía: [Szeliski, 2022, Gonzalez and Woods, 2018]

5.2 Características del Problema y Naturaleza de los Agentes, 5 hr(s)

Competencias:

Tópicos: **I:** Características del problema: *a*) Totalmente observable versus parcialmente observable *b*) Agente único versus multiagente *c*) Determinista versus estocástico *d*) Estático versus dinámico *e*) Discreto versus continuo **II:** Naturaleza de los agentes: *a*) Autonomía autónoma, semiautónoma, de iniciativa mixta *b*) Reflexivos, basados en metas y basados en utilidad *c*) Toma de decisiones bajo incertidumbre y con información incompleta *d*) La importancia de la percepción y las interacciones ambientales *e*) Agentes basados en aprendizaje *f*) Agentes corporeizados: 1) sensores, dinámicas, efectores **III:** Practicar la identificación de características del problema en entornos de ejemplo **IV:** Profundidad adicional sobre la naturaleza de los agentes con ejemplos **V:** Profundidad adicional sobre Aplicaciones de IA, Crecimiento e Impacto (económico, social, ético, seguridad)

Objetivos de Aprendizaje **I:** Enumerar las características de un problema específico [Aplicar]

Bibliografía: [Szeliski, 2022, Gonzalez and Woods, 2018]

5.3 Percepción y Visión por Computadora, 12 hr(s)

Competencias:

Tópicos: **I:** Visión por computadora: *a*) Adquisición, representación, procesamiento y propiedades de imágenes *b*) Representación de formas, reconocimiento de objetos y segmentación *c*) Análisis de movimiento *d*) Modelos generativos **II:** Audio y reconocimiento de voz **III:** Tacto y propiocepción **IV:** Otras modalidades (por ejemplo, olfato) **V:** Modularidad en el reconocimiento **VI:** Enfoques para el reconocimiento de patrones: *a*) Algoritmos de clasificación y medidas de calidad de clasificación *b*) Técnicas estadísticas *c*) Técnicas de aprendizaje profundo

Objetivos de Aprendizaje **I:** Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir] **II:** Listar al menos tres enfoques de segmentación de imágenes, como umbralización, algoritmos basados en bordes y basados en regiones, junto con sus características definitorias, fortalezas y debilidades [Listar/Enumerar] **III:** Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar] **IV:** Distinguir los objetivos del reconocimiento de sonido, reconocimiento de voz y reconocimiento de hablante, e identificar cómo se manejará diferente la señal de audio cruda en cada uno de estos casos [Distinguir] **V:** Proporcionar al

menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, por ejemplo, datos táctiles interpretados como imágenes 2D de banda única [Generar] **VI:** Implementar un algoritmo de extracción de características en datos reales, por ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier que describen un segmento corto de señal de audio [Aplicar] **VII:** Implementar un algoritmo que combine características en percepciones de nivel superior, por ejemplo, un contorno o polígono a partir de primitivas visuales o hipótesis de fonemas a partir de una señal de audio [Aplicar] **VIII:** Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar] **IX:** Evaluar el rendimiento de la extracción de características subyacente, relativo a al menos un enfoque alternativo posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8), anterior [Evaluar] **X:** Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir] **XI:** Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]

Bibliografía: [Szeliski, 2022, Prince, 2012]

5.4 Fundamentos de Aprendizaje Automático, 5 hr(s)

Competencias:

Tópicos: **I:** Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: *a*) Aprendizaje supervisado: 1) Clasificación 2) Regresión *b*) Aprendizaje por refuerzo *c*) Aprendizaje no supervisado: 1) Agrupamiento **II:** Ideas fundamentales: *a*) Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. *b*) Fuentes de error e indecidibilidad en el aprendizaje automático **III:** Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: *a*) Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente **IV:** El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): *a*) El compromiso entre sesgo (subajuste) - varianza (sobreajuste)

Objetivos de Aprendizaje **I:** Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar] **II:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] **III:** Formular una aplicación como un problema de clasificación, incluyendo las características de entrada disponibles y la salida a predecir (por ejemplo, identificar caracteres alfabéticos a partir de entrada de cuadrícula de píxeles) [Aplicar] **IV:** Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] **V:** Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

Bibliografía: [Goodfellow et al., 2016, Russell and Norvig, 2020]

5.5 Aprendizaje Automático: Datos, Representaciones y Evaluación, 5 hr(s)

Competencias:

Tópicos: **I:** Trabajar con Datos: *a*) Preprocesamiento de datos: 1) Importancia y peligros de las elecciones de preprocesamiento *b*) Manejo de valores faltantes (imputación, marcar como faltante): 1) Implicaciones de imputar versus marcar como faltante *c*) Codificación de variables categóricas, codificación de datos de valor real *d*) Normalización/estandarización *e*) énfasis en datos reales, no ejemplos de libros de texto **II:** Representaciones: *a*) Espacios de hipótesis y complejidad *b*) Expansión de características de base simple, como elevar al cuadrado características univariadas *c*) Representaciones de características aprendidas **III:** Evaluación del aprendizaje automático: *a*) Separación de conjuntos de entrenamiento, validación y prueba *b*) Métricas

de rendimiento para clasificadores *c*) Estimación del rendimiento en prueba en datos retenidos *d*) Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación *e*) Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones

Objetivos de Aprendizaje I: Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] **II:** Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] **III:** Explicar los procedimientos adecuados de evaluación de AA, incluyendo las diferencias entre rendimiento en entrenamiento y prueba, y qué puede salir mal con el proceso de evaluación llevando a informes inexactos del rendimiento de AA [Explicar] **IV:** Comparar dos algoritmos de aprendizaje automático en un conjunto de datos, implementando la metodología de preprocesamiento de datos y evaluación (por ejemplo, métricas y manejo de divisiones entrenamiento/prueba) desde cero [Aplicar]

Bibliografía: [Goodfellow et al., 2016, Russell and Norvig, 2020]

5.6 Redes Neuronales y Ética, 4 hr(s)

Competencias:

Tópicos: I: Redes neuronales básicas: *a*) Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos *b*) Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) **II:** ética para el Aprendizaje Automático: *a*) Enfocarse en datos reales, escenarios reales y estudios de caso *b*) Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas

Objetivos de Aprendizaje I: Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]

Bibliografía: [Goodfellow et al., 2016, Russell and Norvig, 2020]

5.7 Formulación del Aprendizaje Automático y Evaluación, 3 hr(s)

Competencias:

Tópicos: I: Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: *a*) Función objetivo *b*) Descenso de gradiente *c*) Regularización para evitar sobreajuste (formulación matemática) **II:** Conjuntos de modelos: *a*) Combinación de mayoría ponderada simple **III:** Evaluación de rendimiento: *a*) Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) *b*) Métricas de rendimiento para regresores *c*) Matriz de confusión *d*) Validación cruzada: 1) Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)

Objetivos de Aprendizaje I: Comparar y contrastar varias técnicas de aprendizaje (por ejemplo, árboles de decisión, regresión logística, Bayes ingenuo, redes neuronales y redes de creencia), proporcionando ejemplos de cuándo cada estrategia es superior [Comparar] **II:** Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] **III:** Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] **IV:** Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] **V:** Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]

Bibliografía: [Goodfellow et al., 2016]

5.8 Aprendizaje Profundo, RL y Ética, 3 hr(s)

Competencias:

Tópicos: I: Aprendizaje profundo: *a*) Redes feed-forward profundas (solo intuición, sin matemáticas) *b*) Redes neuronales convolucionales (solo intuición, sin matemáticas) *c*) Visualización de representaciones de características aprendidas de redes profundas *d*) Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) **II:** Visión general de los métodos de aprendizaje por refuerzo **III:** Dos o más aplicaciones de algoritmos de aprendizaje automático: *a*) Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego **IV:** ética para el Aprendizaje Automático: *a*) Continuar enfocándose en datos reales, escenarios reales y estudios de caso *b*) Privacidad *c*) Equidad *d*) Propiedad intelectual *e*) Explicabilidad

Objetivos de Aprendizaje I: Visualizar el progreso del entrenamiento de una red neuronal a través de curvas de aprendizaje en un kit de herramientas establecido (por ejemplo, TensorBoard) y visualizar las características aprendidas de la red [Aplicar] **II:** Dada una aplicación real de aprendizaje automático, describir problemas éticos respecto a las elecciones de datos, pasos de preprocesamiento, selección de algoritmo y visualización/presentación de resultados [Aplicar]

Bibliografía: [Goodfellow et al., 2016]

5.9 Percepción y Visión por Computadora, 12 hr(s)

Competencias:

Tópicos: I: Visión por computadora: *a*) Adquisición, representación, procesamiento y propiedades de imágenes *b*) Representación de formas, reconocimiento de objetos y segmentación *c*) Análisis de movimiento *d*) Modelos generativos **II:** Audio y reconocimiento de voz **III:** Tacto y propiocepción **IV:** Otras modalidades (por ejemplo, olfato) **V:** Modularidad en el reconocimiento **VI:** Enfoques para el reconocimiento de patrones: *a*) Algoritmos de clasificación y medidas de calidad de clasificación *b*) Técnicas estadísticas *c*) Técnicas de aprendizaje profundo

Objetivos de Aprendizaje I: Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir] **II:** Listar al menos tres enfoques de segmentación de imágenes, como umbralización, algoritmos basados en bordes y basados en regiones, junto con sus características definitorias, fortalezas y debilidades [Listar/Enumerar] **III:** Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar] **IV:** Distinguir los objetivos del reconocimiento de sonido, reconocimiento de voz y reconocimiento de hablante, e identificar cómo se manejará diferente la señal de audio cruda en cada uno de estos casos [Distinguir] **V:** Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, por ejemplo, datos táctiles interpretados como imágenes 2D de banda única [Generar] **VI:** Implementar un algoritmo de extracción de características en datos reales, por ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier que describen un segmento corto de señal de audio [Aplicar] **VII:** Implementar un algoritmo que combine características en percepciones de nivel superior, por ejemplo, un contorno o polígono a partir de primitivas visuales o hipótesis de fonemas a partir de una señal de audio [Aplicar] **VIII:** Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar] **IX:** Evaluar el rendimiento de la extracción de características subyacente, relativo a al menos un enfoque alternativo posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8), anterior [Evaluar] **X:** Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir] **XI:** Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]

Bibliografía: [Szeliski, 2022]

6. Bibliografía

[Gonzalez and Woods, 2018] Gonzalez, R. C. and Woods, R. E. (2018). *Digital Image Processing*. Pearson, 4th edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Prince, 2012] Prince, S. J. D. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Szeliski, 2022] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer, 2nd edition.

[Gonzalez and Woods, 2018] Gonzalez, R. C. and Woods, R. E. (2018). *Digital Image Processing*. Pearson, 4th edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Prince, 2012] Prince, S. J. D. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Szeliski, 2022] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

AI367 Tópicos en Inteligencia Artificial (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (9no Sem-Pág 492)

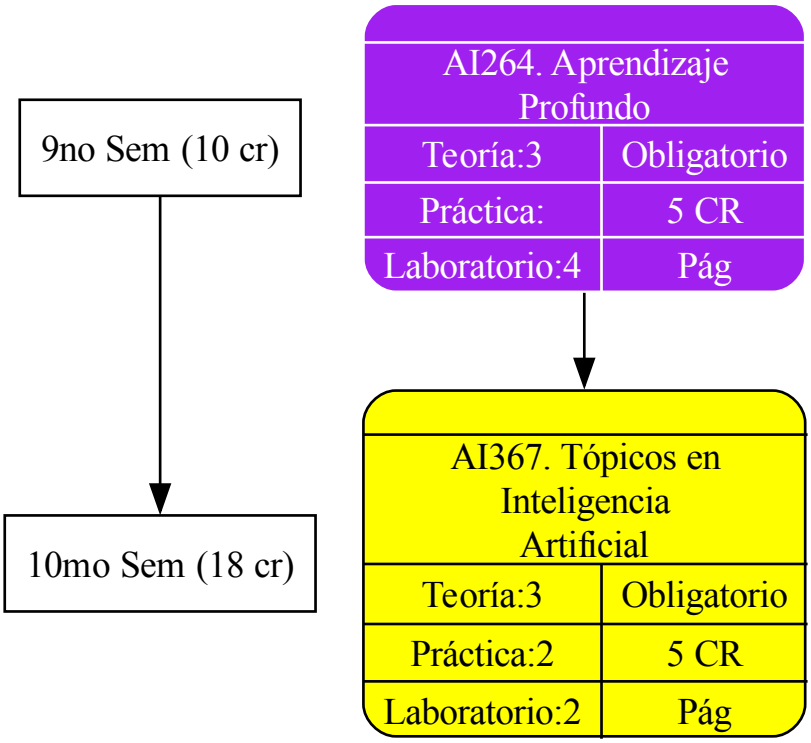


Figura 7.64: Mapa de Conexión. AI367 Tópicos en Inteligencia Artificial

2. Fundamentación

La Inteligencia Artificial es un campo en rápida evolución. Este curso explora temas avanzados y contemporáneos que están dando forma actualmente a la industria y la investigación. Se pone énfasis en la IA Generativa, Modelos de Lenguaje Grandes (LLMs) y Aprendizaje por Refuerzo Avanzado. Los estudiantes analizarán las arquitecturas detrás de los sistemas modernos de IA, como los Transformers, y discutirán los desafíos técnicos y sociales de desplegar agentes inteligentes autónomos a gran escala.

3. Objetivos del curso.

- Comprender la arquitectura y el proceso de entrenamiento de Modelos de Lenguaje Grandes.
- Dominar los principios de la IA Generativa para síntesis de texto e imágenes.

- Implementar algoritmos avanzados de Aprendizaje por Refuerzo para la toma de decisiones.
- Aplicar técnicas de ingeniería de prompts y ajuste fino a modelos existentes.
- Evaluar críticamente los últimos artículos de investigación y tendencias en IA.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

5. Contenido

5.1 Fundamentos de Aprendizaje Automático, 6 hr(s)

Competencias: ABET-1

Tópicos: **I:** Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: *a*) Aprendizaje supervisado: 1) Clasificación 2) Regresión *b*) Aprendizaje por refuerzo *c*) Aprendizaje no supervisado: 1) Agrupamiento **II:** Ideas fundamentales: *a*) Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. *b*) Fuentes de error e indecidibilidad en el aprendizaje automático **III:** Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: *a*) Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente **IV:** El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): *a*) El compromiso entre sesgo (subajuste) - varianza (sobreajuste)

Objetivos de Aprendizaje **I:** Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar] **II:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] **III:** Formular una aplicación como un problema de clasificación, incluyendo las características de entrada disponibles y la salida a predecir (por ejemplo, identificar caracteres alfabéticos a partir de entrada de cuadrícula de píxeles) [Aplicar] **IV:** Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] **V:** Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

Bibliografía: [Goodfellow et al., 2016, Foster, 2022]

5.2 Aprendizaje Automático: Datos, Representaciones y Evaluación, 5 hr(s)

Competencias: ABET-1

Tópicos: **I:** Trabajar con Datos: *a*) Preprocesamiento de datos: 1) Importancia y peligros de las elecciones de preprocesamiento *b*) Manejo de valores faltantes (imputación, marcar como faltante): 1) Implicaciones de imputar versus marcar como faltante *c*) Codificación de variables categóricas, codificación de datos de valor real *d*) Normalización/estandarización *e*) énfasis en datos reales, no ejemplos de libros de texto **II:** Representaciones: *a*) Espacios de hipótesis y complejidad *b*) Expansión de características de base simple, como elevar al cuadrado características univariadas *c*) Representaciones de características aprendidas **III:** Evaluación del aprendizaje automático: *a*) Separación de conjuntos de entrenamiento, validación y prueba *b*) Métricas de rendimiento para clasificadores *c*) Estimación del rendimiento en prueba en datos retenidos *d*) Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación *e*) Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones

Objetivos de Aprendizaje **I:** Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] **II:** Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el

clasificador y obtener una predicción en nuevos datos [Aplicar] **III:** Explicar los procedimientos adecuados de evaluación de AA, incluyendo las diferencias entre rendimiento en entrenamiento y prueba, y qué puede salir mal con el proceso de evaluación llevando a informes inexactos del rendimiento de AA [Explicar] **IV:** Comparar dos algoritmos de aprendizaje automático en un conjunto de datos, implementando la metodología de preprocesamiento de datos y evaluación (por ejemplo, métricas y manejo de divisiones entrenamiento/prueba) desde cero [Aplicar]

Bibliografía: [Goodfellow et al., 2016, Foster, 2022]

5.3 Redes Neuronales y Ética, 5 hr(s)

Competencias: ABET-1

Tópicos: I: Redes neuronales básicas: *a*) Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos *b*) Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) **II:** ética para el Aprendizaje Automático: *a*) Enfocarse en datos reales, escenarios reales y estudios de caso *b*) Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas

Objetivos de Aprendizaje I: Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]

Bibliografía: [Goodfellow et al., 2016, Foster, 2022]

5.4 Formulación del Aprendizaje Automático y Evaluación, 3 hr(s)

Competencias:

Tópicos: I: Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: *a*) Función objetivo *b*) Descenso de gradiente *c*) Regularización para evitar sobreajuste (formulación matemática) **II:** Conjuntos de modelos: *a*) Combinación de mayoría ponderada simple **III:** Evaluación de rendimiento: *a*) Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) *b*) Métricas de rendimiento para regresores *c*) Matriz de confusión *d*) Validación cruzada: 1) Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)

Objetivos de Aprendizaje I: Comparar y contrastar varias técnicas de aprendizaje (por ejemplo, árboles de decisión, regresión logística, Bayes ingenuo, redes neuronales y redes de creencia), proporcionando ejemplos de cuándo cada estrategia es superior [Comparar] **II:** Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] **III:** Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] **IV:** Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] **V:** Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]

Bibliografía: [Goodfellow et al., 2016]

5.5 Aprendizaje Profundo, RL y Ética, 3 hr(s)

Competencias:

Tópicos: I: Aprendizaje profundo: *a*) Redes feed-forward profundas (solo intuición, sin matemáticas) *b*) Redes neuronales convolucionales (solo intuición, sin matemáticas) *c*) Visualización de representaciones de características aprendidas de redes profundas *d*) Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) **II:** Visión general de los métodos de aprendizaje por refuerzo **III:** Dos o más aplicaciones de algoritmos de aprendizaje automático: *a*) Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego **IV:** ética para el Aprendizaje Automático: *a*) Continuar enfocándose en datos reales, escenarios reales y estudios de caso *b*) Privacidad *c*) Equidad *d*) Propiedad intelectual *e*) Explicabilidad

Objetivos de Aprendizaje I: Visualizar el progreso del entrenamiento de una red neuronal a través de curvas de aprendizaje en un kit de herramientas establecido (por ejemplo, TensorBoard) y visualizar las características aprendidas de la red [Aplicar] **II:** Dada una aplicación real de aprendizaje automático, describir problemas éticos respecto a las elecciones de datos, pasos de preprocesamiento, selección de algoritmo y visualización/presentación de resultados [Aplicar]

Bibliografía: [Goodfellow et al., 2016]

5.6 Procesamiento del Lenguaje Natural, 20 hr(s)

Competencias: ABET-1

Tópicos: I: Gramáticas deterministas y estocásticas **II:** Algoritmos de análisis sintáctico: *a*) CFGs y analizadores de tabla (por ejemplo, CYK) *b*) CFGs probabilísticos y CYK ponderado **III:** Representación del significado/Semántica: *a*) Representaciones del conocimiento basadas en lógica *b*) Roles semánticos *c*) Representaciones temporales *d*) Creencias, deseos e intenciones **IV:** Métodos basados en corpus **V:** N-gramas y HMMs **VI:** Suavizado y retroceso **VII:** Ejemplos de uso: Etiquetado POS y morfología **VIII:** Recuperación de información: *a*) Modelo de espacio vectorial: 1) TF & IDF *b*) Precisión y exhaustividad **IX:** Extracción de información **X:** Traducción de lenguaje **XI:** Clasificación de texto, categorización: *a*) Modelo de bolsa de palabras **XII:** Aprendizaje profundo para PLN: *a*) RNNs *b*) Transformers *c*) Incrustaciones multimodales (por ejemplo, imágenes + texto) *d*) Modelos de lenguaje generativos

Objetivos de Aprendizaje I: Definir y contrastar gramáticas deterministas y estocásticas, proporcionando ejemplos para mostrar la adecuación de cada una [Definir] **II:** Simular, aplicar o implementar algoritmos clásicos y estocásticos para analizar lenguaje natural [Aplicar] **III:** Identificar los desafíos de representar el significado [Analizar] **IV:** Listar las ventajas de usar corpus estándar. Identificar ejemplos de corpus actuales para una variedad de tareas de PLN [Listar/Enumerar] **V:** Identificar técnicas para recuperación de información, traducción de lenguaje y clasificación de texto [Analizar] **VI:** Implementar una transformación TF/IDF, usarla para extraer características de un corpus, y entrenar un algoritmo de aprendizaje automático estándar usando esas características para hacer clasificación de texto [Aplicar]

Bibliografía: [Tunstall et al., 2022, Russell and Norvig, 2020]

5.7 Fundamentos de Aprendizaje Automático, 6 hr(s)

Competencias: ABET-1

Tópicos: I: Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: *a*) Aprendizaje supervisado: 1) Clasificación 2) Regresión *b*) Aprendizaje por refuerzo *c*) Aprendizaje no supervisado: 1) Agrupamiento **II:** Ideas fundamentales: *a*) Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. *b*) Fuentes de error e indecidibilidad en el aprendizaje automático **III:** Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: *a*) Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente **IV:** El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): *a*) El compromiso entre sesgo (subajuste) - varianza (sobreajuste)

Objetivos de Aprendizaje I: Describir las diferencias entre los tres estilos principales de aprendizaje (supervisado, por refuerzo y no supervisado) y determinar cuál es apropiado para un dominio de problema particular [Explicar] **II:** Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] **III:** Formular una aplicación como un problema de clasificación, incluyendo las características de entrada disponibles y la salida a predecir (por ejemplo, identificar caracteres alfabéticos a partir de entrada de cuadrícula de píxeles) [Aplicar] **IV:** Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] **V:** Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]

Bibliografía: [Sutton and Barto, 2018]

5.8 Aprendizaje Automático: Datos, Representaciones y Evaluación, 5 hr(s)

Competencias: ABET-1

Tópicos: **I:** Trabajar con Datos: *a*) Preprocesamiento de datos: 1) Importancia y peligros de las elecciones de preprocesamiento *b*) Manejo de valores faltantes (imputación, marcar como faltante): 1) Implicaciones de imputar versus marcar como faltante *c*) Codificación de variables categóricas, codificación de datos de valor real *d*) Normalización/estandarización *e*) énfasis en datos reales, no ejemplos de libros de texto **II:** Representaciones: *a*) Espacios de hipótesis y complejidad *b*) Expansión de características de base simple, como elevar al cuadrado características univariadas *c*) Representaciones de características aprendidas **III:** Evaluación del aprendizaje automático: *a*) Separación de conjuntos de entrenamiento, validación y prueba *b*) Métricas de rendimiento para clasificadores *c*) Estimación del rendimiento en prueba en datos retenidos *d*) Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación *e*) Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones

Objetivos de Aprendizaje **I:** Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] **II:** Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] **III:** Explicar los procedimientos adecuados de evaluación de AA, incluyendo las diferencias entre rendimiento en entrenamiento y prueba, y qué puede salir mal con el proceso de evaluación llevando a informes inexactos del rendimiento de AA [Explicar] **IV:** Comparar dos algoritmos de aprendizaje automático en un conjunto de datos, implementando la metodología de preprocesamiento de datos y evaluación (por ejemplo, métricas y manejo de divisiones entrenamiento/prueba) desde cero [Aplicar]

Bibliografía: [Sutton and Barto, 2018]

5.9 Redes Neuronales y Ética, 5 hr(s)

Competencias: ABET-1

Tópicos: **I:** Redes neuronales básicas: *a*) Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos *b*) Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) **II:** ética para el Aprendizaje Automático: *a*) Enfocarse en datos reales, escenarios reales y estudios de caso *b*) Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas

Objetivos de Aprendizaje **I:** Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]

Bibliografía: [Sutton and Barto, 2018]

5.10 Aplicaciones e Impacto Social de la IA, 6 hr(s)

Competencias: ABET-1

Tópicos: **I:** Al menos una aplicación de IA a un problema y campo específico, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir al menos uno para el Núcleo CS): *a*) Formular y evaluar una aplicación específica como un problema de IA: 1) Cómo lidiar con problemas mal especificados o mal planteados *b*) Disponibilidad/escasez y limpieza de datos: 1) Limpieza y preprocesamiento básico de datos 2) Sesgo del conjunto de datos *c*) Sesgo algorítmico *d*) Sesgo de evaluación *e*) Evaluación de las implicaciones sociales de la aplicación **II:** Modelos generativos profundos desplegados: *a*) Visión general de alto nivel de modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion, etc.),

sus usos y sus limitaciones/peligros. *b*) Visión general de alto nivel de modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard, etc.), sus usos y sus limitaciones/peligros. **III:** Visión general del impacto social de la IA: *a*) ética *b*) Equidad *c*) Confianza/explicabilidad *d*) Privacidad y uso de datos de entrenamiento *e*) Autonomía humana y supervisión/regulaciones/requisitos legales *f*) Sostenibilidad

Objetivos de Aprendizaje I: Dado un dominio de aplicación y problema del mundo real, formular una solución de IA para ello, identificando datos/entrada apropiados, preprocesamiento, representaciones, técnicas de IA y métricas/metodología de evaluación [Aplicar]

Bibliografía: [Russell and Norvig, 2020, Bostrom, 2014]

5.11 Aplicaciones Adicionales e Impacto Social de la IA, 6 hr(s)

Competencias: ABET-1

Tópicos: I: Una o más aplicaciones adicionales de IA a un amplio conjunto de problemas y diversos campos, como medicina, salud, sostenibilidad, redes sociales, economía, educación, robótica, etc. (elegir un área diferente de la elegida para el Núcleo CS): *a*) Formular y evaluar una aplicación específica como un problema de IA: 1) Cómo lidiar con problemas mal especificados o mal planteados *b*) Disponibilidad/escasez y limpieza de datos: 1) Limpieza y preprocesamiento básico de datos 2) Sesgo del conjunto de datos *c*) Sesgo algorítmico *d*) Sesgo de evaluación *e*) Evaluación de las implicaciones sociales de la aplicación **II:** Profundidad adicional sobre modelos generativos profundos desplegados: *a*) Introducción a cómo funcionan los modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion) incluyendo discusión de atención *b*) Introducción a cómo funcionan los modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard) incluyendo discusión de atención *c*) Idea de modelos fundamentales, cómo usarlos, y los beneficios/problemas con entrenarlos a partir de grandes datos **III:** Análisis y discusión del impacto social de la IA: *a*) ética *b*) Equidad *c*) Confianza/explicabilidad *d*) Privacidad y uso de datos de entrenamiento *e*) Autonomía humana y supervisión/regulaciones/requisitos legales *f*) Sostenibilidad

Objetivos de Aprendizaje I: Analizar el impacto social de una o más aplicaciones específicas de IA del mundo real, identificando problemas respecto a ética, equidad, sesgo, confianza y explicabilidad [Evaluar] **II:** Describir algunos de los modos de falla de los modelos generativos profundos actuales para lenguaje o imágenes, y cómo esto podría afectar su uso en una aplicación [Describir]

Bibliografía: [Russell and Norvig, 2020, Bostrom, 2014]

6. Bibliografía

[Bostrom, 2014] Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

[Foster, 2022] Foster, D. (2022). *Generative Deep Learning*. O'Reilly Media, 2nd edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

[Tunstall et al., 2022] Tunstall, L., von Werra, L., and Wolf, T. (2022). *Natural Language Processing with Transformers*. O'Reilly Media.

[Bostrom, 2014] Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

[Foster, 2022] Foster, D. (2022). *Generative Deep Learning*. O'Reilly Media, 2nd edition.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Russell and Norvig, 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

[Tunstall et al., 2022] Tunstall, L., von Werra, L., and Wolf, T. (2022). *Natural Language Processing with Transformers*. O'Reilly Media.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

MA403 Proyecto de final de carrera III (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 8 horas;
- **Prerrequisitos**:
  - MA402 Proyecto de final de carrera II (9no Sem-Pág 495)

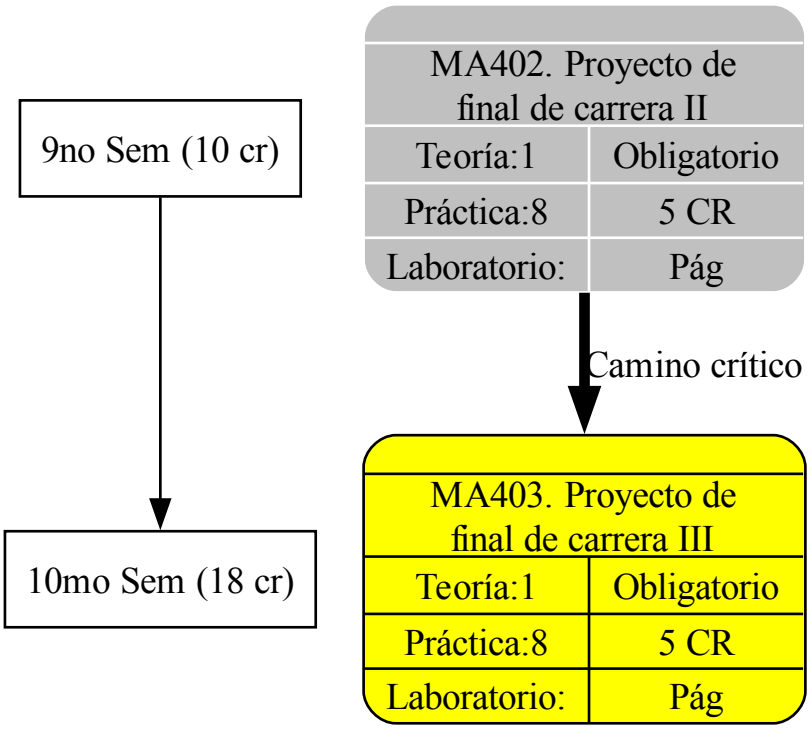


Figura 7.65: Mapa de Conexión. MA403 Proyecto de final de carrera III

2. Fundamentación

La asignatura de Tesis de Grado en Matemática Computacional (MA403) es la culminación del proceso de formación académica e investigación del estudiante. Este curso se enfoca en la consolidación del informe final de tesis, asegurando el cumplimiento de los estándares científicos y de probidad académica. El estudiante deberá demostrar una integración total de conocimientos matemáticos y habilidades computacionales para resolver un problema de investigación, finalizando con la presentación formal del manuscrito y la preparación técnica para la defensa ante la comunidad académica y profesional.

3. Objetivos del curso.

- Implementar completamente la solución computacional diseñada, aplicando técnicas avanzadas de matemáticas computacionales.

- Realizar análisis exhaustivos de resultados y validaciones metodológicas rigurosas.
- Documentar profesionalmente todo el proceso de investigación y desarrollo del proyecto.
- Evaluar el impacto y las limitaciones de la solución desarrollada, proponiendo mejoras futuras.
- Defender públicamente el proyecto ante un jurado especializado, demostrando dominio del tema y habilidades de comunicación técnica.
- Demostrar capacidad de trabajo autónomo, gestión de proyecto y ética profesional en el desarrollo de investigación aplicada.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Evaluar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Evaluar)

ABET-4) Habilidad para comunicarse de manera efectiva con diversos tipos de audiencias. (Evaluar)

ABET-5) Habilidad para comprender las responsabilidades éticas y profesionales y el impacto de las soluciones técnicas y/o científicas en contextos globales, económicos, ambientales y sociales. (Evaluar)

ABET-6) Habilidad para desempeñarse efectivamente en equipos que establecen metas, planifican tareas, cumplen con los plazos y analizan el riesgo y la incertidumbre. (Evaluar)

5. Contenido

5.1 Consolidación Técnica y Optimización Final, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Revisión final de la implementación algorítmica y eficiencia computacional **II:** Optimización de código y escalabilidad de la solución **III:** Verificación formal y validación de resultados numéricos **IV:** Gestión de depósitos de software y reproducibilidad científica

Objetivos de Aprendizaje **I:** Conducir estudios completos usando métodos de investigación [Evaluar] **II:** Demostrar formalmente propiedades de las soluciones [Evaluar] **III:** Implementar procesos matemáticos complejos en computadora [Usar] **IV:** Optimizar soluciones computacionales para eficiencia [Evaluar]

Bibliografía: [Steve, 2004, E, 1997]

5.2 Validación Metodológica y Análisis Crítico, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: **I:** Análisis comparativo de resultados frente al estado del arte **II:** Discusión profunda sobre el impacto y aplicaciones del modelo **III:** Validación estadística final y análisis de error **IV:** Evaluación de la contribución original del trabajo

Objetivos de Aprendizaje **I:** Analizar impacto social de soluciones científicas [Evaluar] **II:** Conducir validaciones metodológicas rigurosas [Evaluar] **III:** Reconocer estructura formal de problemas complejos [Evaluar] **IV:** Interpretar resultados en contexto aplicado [Usar]

Bibliografía: [C, 2017, P et al., 2005]

5.3 Redacción Académica y Memoria de Tesis, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: I: Estructura final del manuscrito de tesis según estándares universitarios **II:** Técnicas de redacción científica para matemáticas y computación **III:** Gestión de citas, bibliografía y probidad académica **IV:** Revisión de pares y procesos de corrección editorial

Objetivos de Aprendizaje I: Aplicar principios éticos en comunicación científica [Usar] **II:** Comunicar efectivamente resultados complejos [Evaluar] **III:** Aprender de forma continua mediante retroalimentación [Usar] **IV:** Documentar procesos de forma profesional [Evaluar]

Bibliografía: [Michael, 1996, R and W, 2008]

5.4 Sustentación y Proyección Profesional, 20 hr(s)

Competencias: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6

Tópicos: I: Preparación del protocolo de defensa pública **II:** Oratoria técnica y argumentación ante pares académicos **III:** Visualización de datos avanzada para presentaciones de impacto **IV:** Plan de publicación de resultados en foros especializados

Objetivos de Aprendizaje I: Desempeñarse efectivamente en defensa pública [Evaluar] **II:** Gestionar proyectos científicos complejos [Evaluar] **III:** Resolver problemas matemáticos avanzados [Evaluar] **IV:** Responder críticas constructivamente [Usar]

Bibliografía: [L, 2018, H, 2006]

6. Bibliografía

[C, 2017] C, M. D. (2017). *Design and Analysis of Experiments*. John Wiley & Sons, 9th edition.

[E, 1997] E, K. D. (1997). *The Art of Computer Programming*. Addison-Wesley.

[H, 2006] H, A. R. R. (2006). *Dazzle 'em with Style: The Art of Oral Scientific Presentation*. Elsevier Academic Press.

[L, 2018] L, T. K. (2018). *A Manual for Writers of Research Papers, Theses, and Dissertations*. University of Chicago Press.

[Michael, 1996] Michael, A. (1996). *The Craft of Scientific Writing*. Springer.

[P et al., 2005] P, B. G. E., Stuart, H. J., and G, H. W. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience.

[R and W, 2008] R, M. J. and W, M. R. (2008). *Successful Scientific Writing: A Step-by-Step Guide for the Biological and Medical Sciences*. Cambridge University Press.

[Steve, 2004] Steve, M. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

[C, 2017] C, M. D. (2017). *Design and Analysis of Experiments*. John Wiley & Sons, 9th edition.

[E, 1997] E, K. D. (1997). *The Art of Computer Programming*. Addison-Wesley.

[H, 2006] H, A. R. R. (2006). *Dazzle 'em with Style: The Art of Oral Scientific Presentation*. Elsevier Academic Press.

[L, 2018] L, T. K. (2018). *A Manual for Writers of Research Papers, Theses, and Dissertations*. University of Chicago Press.

[Michael, 1996] Michael, A. (1996). *The Craft of Scientific Writing*. Springer.

[P et al., 2005] P, B. G. E., Stuart, H. J., and G, H. W. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience.

[R and W, 2008] R, M. J. and W, M. R. (2008). *Successful Scientific Writing: A Step-by-Step Guide for the Biological and Medical Sciences*. Cambridge University Press.

[Steve, 2004] Steve, M. (2004). *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press, 2nd edition.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

QC381 Introduction a la Computación Cuántica (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - QC251 Mecánica Cuántica (8vo Sem-Pág 468)
  - MA381 Estructuras Algebraicas (9no Sem-Pág 504)

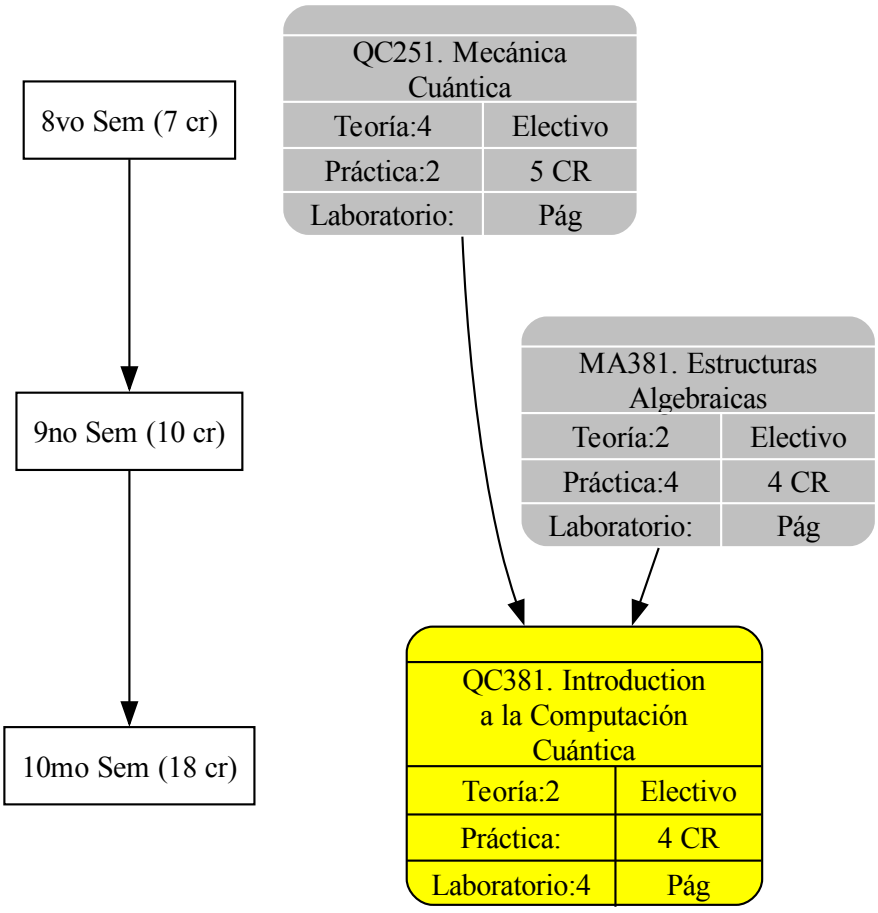


Figura 7.66: Mapa de Conexión. QC381 Introduction a la Computación Cuántica

2. Fundamentación

Este curso introduce los principios fundamentales de la computación cuántica, incluyendo cúbits, superposición, entrelazamiento, algoritmos cuánticos y hardware. Los estudiantes contrastarán modelos clásicos y cuánticos, explorando aplicaciones en criptografía, optimización, simulación física y aprendizaje automático cuántico, usando frameworks como Qiskit o Cirq. Incluye un módulo práctico

sobre arquitecturas de hardware (superconductores, trampas de iones) y su impacto en el diseño de algoritmos.

3. Objetivos del curso.

- Comprender los postulados de la mecánica cuántica aplicados a la computación.
- Implementar circuitos cuánticos básicos y algoritmos de optimización.
- Analizar el impacto ético, técnico y comercial de la computación cuántica.
- Evaluar las limitaciones del hardware en aplicaciones del mundo real.

4. Competencias Generales (*Outcomes*)

ABET-1) Habilidad para identificar, formular y resolver problemas técnicos o científicos ampliamente definidos mediante la aplicación de conocimientos de matemáticas y ciencias y/o temas técnicos en áreas relevantes para la disciplina. (Usar)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

5. Contenido

5.1 Conceptos fundamentales, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Cúbits y representación en la esfera de Bloch. **II:** Postulados de la mecánica cuántica. **III:** Compuertas cuánticas (Hadamard, CNOT, Toffoli).

Objetivos de Aprendizaje **I:** Modelar matemáticamente un cúbit [Usar]. **II:** Simular operaciones básicas en Qiskit [Evaluar].

Bibliografía: [Nielsen and Chuang, 2010, Team, 2023]

5.2 Arquitecturas Cuánticas, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: **I:** Principios: *a*) El principio de dualidad onda-partícula *b*) El principio de incertidumbre en el experimento de doble rendija *c*) ¿Qué es un Qubit? Superposición, interferencia y medición. Fotones como qubits *d*) Sistemas de dos qubits, Entrelazamiento, estados de Bell, El teorema de No Señalización **II:** Axiomas de la MC: principio de superposición, axioma de medición, evolución unitaria **III:** Compuertas de un solo qubit para el modelo de circuito de computación cuántica: X, Z, H **IV:** Compuertas de dos qubits y productos tensoriales, trabajando con matrices **V:** El Teorema de No Clonación. El protocolo de Teleportación Cuántica **VI:** Algoritmos: *a*) Algoritmos cuánticos simples: Bernstein-Vazirani, algoritmo de Simon *b*) Implementar Deutsch-Josza con Interferómetros Mach-Zehnder *c*) Factorización cuántica (Algoritmo de Shor) *d*) Búsqueda cuántica (Algoritmo de Grover) **VII:** Aspectos de implementación: *a*) La implementación física de qubits *b*) Control clásico de una Unidad de Procesamiento Cuántico (QPU) *c*) Mitigación y control de errores, NISQ y más allá *d*) Enfoques de medición **VIII:** Aplicaciones Emergentes: *a*) Cifrado post-cuántico *b*) La Internet Cuántica *c*) Computación cuántica adiabática (AQC) y recocido cuántico

Objetivos de Aprendizaje **I:** Discutir cómo un objeto cuántico producido como partícula se propaga como una onda y se detecta como partícula con una distribución de probabilidad correspondiente a la onda [Debatir] **II:** Discutir la naturaleza a nivel cuántico que es inherentemente probabilística [Debatir] **III:** Expresar tu punto de vista sobre el entrelazamiento que puede usarse para crear correlaciones no clásicas, pero no hay forma de usar el entrelazamiento cuántico para enviar mensajes más rápido que la velocidad de la luz [Evaluar] **IV:** Describir el paralelismo cuántico y el papel de la interferencia constructiva vs destructiva en algoritmos cuánticos dada la naturaleza probabilística de las mediciones [Describir] **V:** Analizar un fragmento de código

proporcionando el papel de la transformada cuántica de Fourier (QFT) en el algoritmo de Shor [Analizar] **VI:** Escribir un programa para implementar el algoritmo de Shor en un simulador, destacando los componentes clásicos y aspectos del algoritmo de Shor [Escribir] **VII:** Enumerar los detalles de cada modalidad de qubit (por ejemplo, ion atrapado, superconductor, espín de silicio, fotónico, punto cuántico, átomo neutro, topológico, centro de color, electrón sobre helio) [Enumerar] **VIII:** Contrastar AQC con el modelo de compuertas de computación cuántica y los problemas para los que cada uno es más adecuado resolver [Contrastar]

Bibliografía: [Möttönen and Vartiainen, 2023, IBM Research, 2023]

5.3 Estrategias Algorítmicas, 12 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Computación cuántica **II:** Algoritmo de Deutsch-Jozsa. **III:** Búsqueda de Grover (aplicada a optimización combinatoria). **IV:** Factorización de Shor y su impacto en RSA.

Objetivos de Aprendizaje I: Implementar Grover para problemas SAT [Usar].

Bibliografía: [Preskill, 2018, Team, 2023]

5.4 Lenguajes Formales y Autómatas, 10 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Computación cuántica: *a*) Postulados de la mecánica cuántica: 1) Espacio de estados 2) Evolución del estado 3) Composición de estados 4) Medición del estado *b*) Representaciones de vector columna de qubits *c*) Representaciones matriciales de operaciones cuánticas *d*) Compuertas cuánticas simples (por ejemplo, XNOT, CNOT) **II:** Algoritmo de Optimización Cuántica Aproximada (QAOA). **III:** Aplicaciones en logística/finanzas (Optimización de Cartera).

Objetivos de Aprendizaje I: Para un sistema cuántico dar ejemplos que expliquen los siguientes postulados: *a*) Espacio de Estados - estado del sistema representado como un vector unitario en espacio de Hilbert *b*) Evolución del Estado - el uso de operadores unitarios para evolucionar el estado del sistema *c*) Composición de Estados - el uso de producto tensorial para componer estados del sistema *d*) Medición del Estado - la salida probabilística de medir un estado del sistema. [Familiarizarse]. **II:** Diseñar un QAOA para problemas de enrutamiento [Evaluar].

Bibliografía: [Schuld and Petruccione, 2021, Bravyi et al., 2022]

5.5 Ethics and Post-Quantum Security, 6 hr(s)

Competencias: ABET-1,ABET-3

Tópicos: I: Criptografía post-cuántica (Estandarización NIST PQC). **II:** Supremacía cuántica: debate técnico/ético. **III:** Patentes y geopolítica cuántica.

Objetivos de Aprendizaje I: Proponer estrategias de migración post-cuántica [Familiarizarse].

Bibliografía: [Team, 2019]

6. Bibliografía

[Bravyi et al., 2022] Bravyi, S., Dial, O., Gambetta, J. M., Gil, D., and Nazario, Z. (2022). Quantum algorithms for fixed qubit architectures. *Nature Reviews Physics*, 4(8):499–512. Survey on quantum algorithms for optimization and simulation with near-term devices.

[IBM Research, 2023] IBM Research (2023). Ibm quantum hardware roadmap. Technical report, IBM Research. Accedido: 2026-02-13. La página original ha sido movida o actualizada.

[Möttönen and Vartiainen, 2023] Möttönen, M. and Vartiainen, J. (2023). *Quantum Computing: From Qubits to Quantum Machines*. Springer.

[Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

[Preskill, 2018] Preskill, J. (2018). Lecture notes on quantum computation. Curso Ph219/CS219, California Institute of Technology. Material disponible en línea. La URL original `http://theory.caltech.edu/~preskill/ph219/` no está operativa (febrero 2026). Se recomienda buscar la versión actual en el sitio web del autor.

[Schuld and Petruccione, 2021] Schuld, M. and Petruccione, F. (2021). *Machine Learning with Quantum Computers*. Springer, Cham, Switzerland, 2nd edition. Comprehensive introduction to quantum machine learning algorithms and implementations.

[Team, 2019] Team, G. A. Q. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574.

[Team, 2023] Team, I. Q. (2023). *Qiskit Textbook*. IBM.

[Bravyi et al., 2022] Bravyi, S., Dial, O., Gambetta, J. M., Gil, D., and Nazario, Z. (2022). Quantum algorithms for fixed qubit architectures. *Nature Reviews Physics*, 4(8):499–512. Survey on quantum algorithms for optimization and simulation with near-term devices.

[IBM Research, 2023] IBM Research (2023). Ibm quantum hardware roadmap. Technical report, IBM Research. Accedido: 2026-02-13. La página original ha sido movida o actualizada.

[Möttönen and Vartiainen, 2023] Möttönen, M. and Vartiainen, J. (2023). *Quantum Computing: From Qubits to Quantum Machines*. Springer.

[Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

[Preskill, 2018] Preskill, J. (2018). Lecture notes on quantum computation. Curso Ph219/CS219, California Institute of Technology. Material disponible en línea. La URL original `http://theory.caltech.edu/~preskill/ph219/` no está operativa (febrero 2026). Se recomienda buscar la versión actual en el sitio web del autor.

[Schuld and Petruccione, 2021] Schuld, M. and Petruccione, F. (2021). *Machine Learning with Quantum Computers*. Springer, Cham, Switzerland, 2nd edition. Comprehensive introduction to quantum machine learning algorithms and implementations.

[Team, 2019] Team, G. A. Q. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574.

[Team, 2023] Team, I. Q. (2023). *Qiskit Textbook*. IBM.

Universidad Nacional Mayor de San Marcos
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

CS254 Visualización de Datos Avanzada (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - CS253 Análisis exploratorio de datos (8vo Sem-Pág 472)

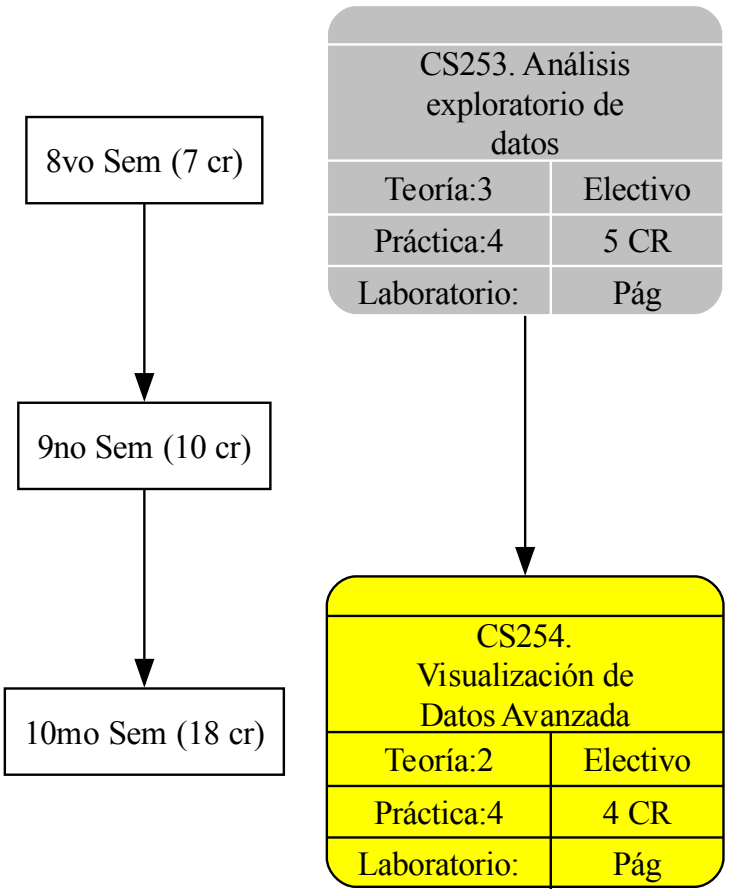


Figura 7.67: Mapa de Conexión. CS254 Visualización de Datos Avanzada

2. Fundamentación

En la era del Big Data, la capacidad de comunicar información compleja de manera visual y efectiva es fundamental. Este curso profundiza en los principios de la percepción visual, el diseño de tableros (*dashboards*) interactivos y el uso de herramientas avanzadas para la narrativa de datos (*storytelling*). Se centra en transformar análisis abstractos en representaciones visuales que faciliten la toma de decisiones estratégicas, integrando estética, funcionalidad y ética en el manejo de la información.

3. Objetivos del curso.

- Aplicar principios de la psicología de la Gestalt y percepción visual al diseño de gráficos.
- Desarrollar visualizaciones interactivas utilizando bibliotecas de vanguardia.
- Diseñar narrativas de datos (storytelling) que faciliten la comprensión de problemas complejos.
- Implementar visualizaciones para datos multidimensionales y geoespaciales.
- Evaluar críticamente la efectividad y la ética de las representaciones visuales.

4. Competencias Generales (*Outcomes*)

ABET-3) Habilidad para desarrollar y llevar a cabo experimentos o probar hipótesis, analizar e interpretar datos y utilizar el juicio científico para obtener conclusiones. (Usar)

ABET-2) Habilidad para formular o diseñar un sistema, proceso, procedimiento o programa para el propósito previsto. (Usar)

5. Contenido

5.1 Percepción Visual y Principios de Diseño, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Teoría del color y luminancia en la visualización. **II:** Principios de la Gestalt aplicados al diseño de información. **III:** Atributos pre-atentivos y carga cognitiva. **IV:** Crítica de visualizaciones: Claridad vs. Decoración (Chartjunk).

Objetivos de Aprendizaje I: Explicar cómo el sistema visual humano procesa la información gráfica [Familiarizarse]. **II:** Seleccionar paletas de colores y tipografías que mejoren la legibilidad [Usar].

Bibliografía: [Cairo, 2016, Ware, 2020]

5.2 Visualización Interactiva y Dinámica, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Introducción a D3.js y gramática de gráficos avanzada. **II:** Interacciones: Filtrado, zoom, brushing y linking. **III:** Animación para mostrar cambios temporales. **IV:** Diseño responsivo para diferentes dispositivos.

Objetivos de Aprendizaje I: Implementar elementos interactivos que permitan la exploración del usuario [Usar]. **II:** Construir prototipos funcionales de visualización dinámica [Evaluar].

Bibliografía: [Munzner, 2014, Murray, 2017]

5.3 Visualización de Datos Espaciales y Temporales, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Proyecciones cartográficas y mapas coropléticos. **II:** Mapas de burbujas, densidad (heatmaps) y flujos. **III:** Series de tiempo avanzadas: Horizon charts y Streamgraphs. **IV:** Visualización de redes y grafos.

Objetivos de Aprendizaje I: Representar datos geográficos evitando distorsiones comunes [Usar]. **II:** Analizar patrones temporales complejos mediante gráficos especializados [Evaluar].

Bibliografía: [Wilke, 2019, Cairo, 2016]

5.4 Datos Multidimensionales y de Alta Dimensión, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Coordenadas paralelas y gráficos de radar. **II:** Matrices de diagramas de dispersión (SPLOM). **III:** Técnicas de proyección: t-SNE y UMAP para visualización. **IV:** Visualización de jerarquías: Treemaps y Sunbursts.

Objetivos de Aprendizaje I: Reducir la dimensionalidad para visualizar estructuras ocultas [Usar]. **II:** Diseñar representaciones para datos jerárquicos de gran escala [Evaluar].

Bibliografía: [Munzner, 2014, Wilke, 2019]

5.5 Diseño de Dashboards y Business Intelligence, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Arquitectura de la información en tableros de control. **II:** Indicadores Clave de Desempeño (KPI) y alertas visuales. **III:** Herramientas de BI: Tableau, PowerBI y su extensibilidad. **IV:** Usabilidad y experiencia de usuario (UX) en visualización.

Objetivos de Aprendizaje I: Integrar múltiples visualizaciones en un tablero coherente [Usar]. **II:** Evaluar la eficacia de un dashboard mediante pruebas de usabilidad [Evaluar].

Bibliografía: [Few, 2013, Cairo, 2019]

5.6 Narrativa de Datos y Comunicación, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Estructuras narrativas para datos: Scrollytelling. **II:** Eliminación de ruido y énfasis en el mensaje principal. **III:** Anotaciones y guía visual para el espectador. **IV:** El papel del analista como comunicador científico.

Objetivos de Aprendizaje I: Crear una historia convincente a partir de hallazgos estadísticos [Usar]. **II:** Diseñar presentaciones visuales para audiencias no técnicas [Evaluar].

Bibliografía: [Knaflic, 2015, Cairo, 2019]

5.7 Ética y Decepción Visual, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Gráficos engañosos y manipulación de ejes. **II:** Sesgos cognitivos en la interpretación visual. **III:** Accesibilidad: Diseño para daltónicos y diversidad funcional. **IV:** Responsabilidad en la representación de datos sensibles.

Objetivos de Aprendizaje I: Identificar y corregir visualizaciones que distorsionan la realidad [Evaluar]. **II:** Implementar estándares de accesibilidad en productos visuales [Usar].

Bibliografía: [Cairo, 2019, Few, 2013]

5.8 Proyecto Final: Portafolio de Visualización, 8 hr(s)

Competencias: ABET-2,ABET-3

Tópicos: I: Selección de un conjunto de datos complejo y real. **II:** Proceso iterativo de diseño y feedback. **III:** Documentación técnica y justificación estética. **IV:** Defensa del proyecto ante jurado especializado.

Objetivos de Aprendizaje I: Construir una solución de visualización completa de nivel profesional [Evaluar]. **II:** Sustentar las decisiones técnicas y de diseño basadas en teoría [Evaluar].

Bibliografía: [Knaflic, 2015, Murray, 2017]

6. Bibliografía

[Cairo, 2016] Cairo, A. (2016). *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders.

[Cairo, 2019] Cairo, A. (2019). *How Charts Lie: Getting Smarter about Visual Information*. W. W. Norton & Company.

[Few, 2013] Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Analytics Press, 2nd edition.

[Knaflic, 2015] Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.

[Munzner, 2014] Munzner, T. (2014). *Visualization Analysis and Design*. A K Peters/CRC Press.

[Murray, 2017] Murray, S. (2017). *Interactive Data Visualization for the Web*. O'Reilly Media, 2nd edition.

[Ware, 2020] Ware, C. (2020). *Information Visualization: Perception for Design*. Morgan Kaufmann, 4th edition.

[Wilke, 2019] Wilke, C. O. (2019). *Fundamentals of Data Visualization*. O'Reilly Media.

[Cairo, 2016] Cairo, A. (2016). *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders.

[Cairo, 2019] Cairo, A. (2019). *How Charts Lie: Getting Smarter about Visual Information*. W. W. Norton & Company.

[Few, 2013] Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Analytics Press, 2nd edition.

[Knaflic, 2015] Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.

[Munzner, 2014] Munzner, T. (2014). *Visualization Analysis and Design*. A K Peters/CRC Press.

[Murray, 2017] Murray, S. (2017). *Interactive Data Visualization for the Web*. O'Reilly Media, 2nd edition.

[Ware, 2020] Ware, C. (2020). *Information Visualization: Perception for Design*. Morgan Kaufmann, 4th edition.

[Wilke, 2019] Wilke, C. O. (2019). *Fundamentals of Data Visualization*. O'Reilly Media.

Capítulo 8

Profesores & Cursos

Cursos asignados por profesor

No hay profesores asignados en este documento.

Profesor asignado por curso

- Primer Semestre
  - CS111. Introducción a la Programación (Pág 287)
    - No hay profesor asignado a esta materia
  - MA100. Matemática Básica (Pág 292)
    - No hay profesor asignado a esta materia
  - FG101. Lenguaje (Pág 295)
    - No hay profesor asignado a esta materia
  - FG105. Género y Sociedad (Pág 298)
    - No hay profesor asignado a esta materia
  - FG109. Ambiente y Desarrollo Sostenible (Pág 301)
    - No hay profesor asignado a esta materia
  - FG112. Métodos de gestión personal universitaria (Pág 304)
    - No hay profesor asignado a esta materia
  - FG140. Fundamentos de Investigación Científica (Pág 307)
    - No hay profesor asignado a esta materia
- Segundo Semestre
  - CS100. Introducción a la Ciencia de la Computación (Pág 310)
    - No hay profesor asignado a esta materia
  - CS112. Programación Orientada a Objetos I (Pág 321)
    - No hay profesor asignado a esta materia
  - AI161. IA Aplicada (Pág 326)
    - No hay profesor asignado a esta materia
  - MA111. Cálculo I (Pág 331)
    - No hay profesor asignado a esta materia
  - FI101. Física General (Pág 333)
    - No hay profesor asignado a esta materia
- Tercer Semestre
  - CS113. Programación Orientada a Objetos II (Pág 336)
    - No hay profesor asignado a esta materia
  - CS1D1. Estructuras Discretas I (Pág 342)
    - No hay profesor asignado a esta materia
  - CS2B1. Desarrollo Basado en Plataformas (Pág 345)
    - No hay profesor asignado a esta materia
  - MA112. Cálculo II (Pág 350)

    - No hay profesor asignado a esta materia
  - MA121. Algebra Lineal I (Pág 352)
    - No hay profesor asignado a esta materia
  - ID101. Inglés I (Pág 355)
    - No hay profesor asignado a esta materia
- Cuarto Semestre
  - CS1D2. Estructuras Discretas II (Pág 359)
    - No hay profesor asignado a esta materia
  - CS210. Algoritmos y Estructuras de Datos (Pág 361)
    - No hay profesor asignado a esta materia
  - CS271. Bases de Datos I (Pág 366)
    - No hay profesor asignado a esta materia
  - MA122. Algebra Lineal II (Pág 371)
    - No hay profesor asignado a esta materia
  - MA210. Cálculo III (Pág 374)
    - No hay profesor asignado a esta materia
  - ID102. Inglés II (Pág 377)
    - No hay profesor asignado a esta materia
- Quinto Semestre
  - CS251. Computación Gráfica (Pág 381)
    - No hay profesor asignado a esta materia
  - MA211. Cálculo IV (Pág 384)
    - No hay profesor asignado a esta materia
  - MA221. Análisis Numérico I (Pág 387)
    - No hay profesor asignado a esta materia
  - MA222. Análisis Real I (Pág 390)
    - No hay profesor asignado a esta materia
  - MA251. Introducción a la Estadística (Pág 393)
    - No hay profesor asignado a esta materia
  - MA281. Teoría de números (Pág 395)
    - No hay profesor asignado a esta materia
  - ID201. Inglés III (Pág 399)
    - No hay profesor asignado a esta materia
- Sexto Semestre
  - MA212. Ecuaciones Diferenciales Ordinarias (Pág 402)
    - No hay profesor asignado a esta materia
  - MA325. Análisis Numérico II (Pág 405)
    - No hay profesor asignado a esta materia
  - MA351. Estadística y Probabilidades (Pág 408)
    - No hay profesor asignado a esta materia
  - MA371. Análisis funcional I (Pág 414)
    - No hay profesor asignado a esta materia
  - ID202. Inglés IV (Pág 417)
    - No hay profesor asignado a esta materia
- Séptimo Semestre
  - CS261. Inteligencia Artificial (Pág 421)
    - No hay profesor asignado a esta materia
  - CS3P1. Computación Paralela y Distribuída (Pág 428)
    - No hay profesor asignado a esta materia
  - MA213. Ecuaciones Diferenciales Parciales (Pág 438)
    - No hay profesor asignado a esta materia
  - MA326. Análisis Numérico III (Pág 441)
    - No hay profesor asignado a esta materia
  - MA361. Modelado Matemático (Pág 444)
    - No hay profesor asignado a esta materia
  - ID203. Inglés V (Pág 448)

- No hay profesor asignado a esta materia

- Octavo Semestre
  - AI263. Introducción al Aprendizaje de Máquina (Pág 451)
    - No hay profesor asignado a esta materia
  - MA401. Proyecto de final de carrera I (Pág 454)
    - No hay profesor asignado a esta materia
  - CS2S1. Sistemas Operativos (Pág 457)
    - No hay profesor asignado a esta materia
  - CS3I1. Seguridad en Computación (Pág 462)
    - No hay profesor asignado a esta materia
  - QC251. Mecánica Cuántica (Pág 468)
    - No hay profesor asignado a esta materia
  - CS253. Análisis exploratorio de datos (Pág 472)
    - No hay profesor asignado a esta materia
  - CS370. Big Data y Analítica de Datos (Pág 476)
    - No hay profesor asignado a esta materia
  - STA252. Estadística Inferencial (Pág 480)
    - No hay profesor asignado a esta materia
  - MA327. Análisis Numérico IV (Pág 483)
    - No hay profesor asignado a esta materia
  - MA352. Procesos estocásticos (Pág 486)
    - No hay profesor asignado a esta materia
  - MA372. Análisis funcional II (Pág 489)
    - No hay profesor asignado a esta materia
- Noveno Semestre
  - AI264. Aprendizaje Profundo (Pág 492)
    - No hay profesor asignado a esta materia
  - MA402. Proyecto de final de carrera II (Pág 495)
    - No hay profesor asignado a esta materia
  - CS231. Redes y Comunicación (Pág 499)
    - No hay profesor asignado a esta materia
  - MA381. Estructuras Algebraicas (Pág 504)
    - No hay profesor asignado a esta materia
  - AI262. Procesamiento de Lenguaje Natural (Pág 508)
    - No hay profesor asignado a esta materia
  - CS273. Análisis de series temporales (Pág 511)
    - No hay profesor asignado a esta materia
  - MA341. Optimización I (Pág 515)
    - No hay profesor asignado a esta materia
  - MA355. Teoría de Aprendizaje Estadístico (Pág 518)
    - No hay profesor asignado a esta materia
- Décimo Semestre
  - CS400. Prácticas Pre-profesionales (Pág 521)
    - No hay profesor asignado a esta materia
  - AI268. Visión Computacional (Pág 523)
    - No hay profesor asignado a esta materia
  - AI367. Tópicos en Inteligencia Artificial (Pág 529)
    - No hay profesor asignado a esta materia
  - MA403. Proyecto de final de carrera III (Pág 536)
    - No hay profesor asignado a esta materia
  - QC381. Introduction a la Computación Cuántica (Pág 540)
    - No hay profesor asignado a esta materia
  - CS254. Visualización de Datos Avanzada (Pág 544)
    - No hay profesor asignado a esta materia

Capítulo 9

Equivalencias con otros planes curriculares

A continuación se pueden observar las equivalencias de la presente malla con el(los) Plan(es) curricular(es) anterior(ES). Para ver mayores detalles de la malla propuesta observar la sección 6.2 (Pág. 278).

Equivalencia del Plan 2026 al 2018

Primer Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS111	Introducción a la Programación	5	183033	Algorítmica y Fundamentos de Programación	3er Sem	6
MA100	Matemática Básica	5	CBO105	Matemática Básica	1er Sem	4
FG101	Lenguaje	2	CBO101	Lenguaje	1er Sem	4
FG109	Ambiente y Desarrollo Sostenible	3	CBO202	Medio Ambiente y Desarrollo Sostenible	2do Sem	4
FG112	Métodos de gestión personal universitaria	2	CBO102	Métodos de Estudio Universitario	1er Sem	3
FG140	Fundamentos de Investigación Científica	3	CBO201	Fundamentos de Investigación Científica	2do Sem	3

Segundo Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS100	Introducción a la Ciencia de la Computación	3			2do Sem	
CS112	Programación Orientada a Objetos I	4			2do Sem	
AI161	IA Aplicada	4			2do Sem	
MA111	Cálculo I	5	CBO104	Cálculo I	1er Sem	4
FI101	Física General	5	CBO206	Física General	2do Sem	4

Tercer Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS113	Programación Orientada a Objetos II	3			3er Sem	
CS1D1	Estructuras Discretas I	4	183034	Elementos de Matemática Discreta y Computacional	3er Sem	4
CS2B1	Desarrollo Basado en Plataformas	4			3er Sem	
MA112	Cálculo II	4	CBO204	Cálculo II	2do Sem	4
MA121	Algebra Lineal I	4	183032	Elementos de Álgebra y Geometría	3er Sem	6

Cuarto Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS1D2	Estructuras Discretas II	3			4to Sem	
CS210	Algoritmos y Estructuras de Datos	4	183044	Estructura de Datos	4to Sem	6
CS271	Bases de Datos I	3	183053	Base de Datos e Ingeniería de Software	5to Sem	6
MA122	Algebra Lineal II	4	183042	Álgebra	4to Sem	6
MA210	Cálculo III	4	183031	Cálculo III	3er Sem	6

Quinto Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS251	Computación Gráfica	3	183072	Computación Gráfica	7mo Sem	6
MA211	Cálculo IV	3	183041	Cálculo IV	4to Sem	6
MA221	Análisis Numérico I	3	183054	Métodos Numéricos y Programación I	5to Sem	5
MA222	Análisis Real I	5	183051	Análisis Real	5to Sem	6
MA251	Introducción a la Estadística	3			5to Sem	
MA281	Teoría de números	2			5to Sem	

Sexto Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
MA212	Ecuaciones Diferenciales Ordinarias	6	183052	Ecuaciones Diferenciales Ordinarias	5^{to} Sem	6
MA325	Análisis Numérico II	4	183063	Métodos Númericos y Programación II	6^{to} Sem	6
MA371	Análisis funcional I	5	183061	Análisis Funcional I	6^{to} Sem	5

Séptimo Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS261	Inteligencia Artificial	3	183074	Inteligencia Artificial	7^{mo} Sem	5
CS3P1	Computación Paralela y Distribuída	3			7^{mo} Sem	
MA213	Ecuaciones Diferenciales Parciales	4	183081	Ecuaciones Diferenciales Parciales I	8^{vo} Sem	6
MA326	Análisis Numérico III	4	183093	Métodos Númericos y Programación III	9^{no} Sem	6
MA361	Modelado Matemático	4	183065	Técnicas de Modelamiento	6^{to} Sem	5

Octavo Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
AI263	Introducción al Aprendizaje de Máquina	4			8^{vo} Sem	
MA401	Proyecto de final de carrera I	3	183084	Proyecto de Investigación	8^{vo} Sem	4
CS2S1	Sistemas Operativos	5	183064	Sistemas Operativos	6^{to} Sem	5
CS3I1	Seguridad en Computación	5			8^{vo} Sem	
QC251	Mecánica Cuántica	5	183123	Mecánica Cuántica	8^{vo} Sem	5
CS253	Análisis exploratorio de datos	5			8^{vo} Sem	
CS370	Big Data y Analítica de Datos	5			8^{vo} Sem	
STA252	Estadística Inferencial	5	183062	Probabilidad e Inferencia Estadística	6^{to} Sem	5
MA327	Análisis Numérico IV	5			8^{vo} Sem	
MA352	Procesos estocásticos	5			8^{vo} Sem	
MA372	Análisis funcional II	5	183071	Análisis Funcional II	7^{mo} Sem	5

Noveno Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
AI264	Aprendizaje Profundo	5			9^{no} Sem	
MA402	Proyecto de final de carrera II	5	183094	Desarrollo del Proyecto de Investigación	9^{no} Sem	5
CS231	Redes y Comunicación	4	183083	Redes, Arquitecturas y Comunicaciones	8^{vo} Sem	6
MA381	Estructuras Algebraicas	4			9^{no} Sem	
AI262	Procesamiento de Lenguaje Natural	4			9^{no} Sem	
CS273	Análisis de series temporales	4			9^{no} Sem	
MA341	Optimización I	4	183091	Optimización	9^{no} Sem	5
MA355	Teoría de Aprendizaje Estadístico	4			9^{no} Sem	

Décimo Semestre – Plan2026			Plan 2018			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CS400	Prácticas Pre-profesionales	3	183103	Práctica Preprofesional	10^{mo} Sem	4
AI268	Visión Computacional	5			10^{mo} Sem	
AI367	Tópicos en Inteligencia Artificial	5			10^{mo} Sem	
MA403	Proyecto de final de carrera III	5	183104	Presentación y Sustentación del Proyecto de Investigación	10^{mo} Sem	6
QC381	Introduction a la Computación Cuántica	4			10^{mo} Sem	
CS254	Visualización de Datos Avanzada	4			10^{mo} Sem	

Equivalencia del Plan 2018 al 2026

Primer Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CBO105	Matemática Básica	4	MA100	Matemática Básica	1er Sem	5
CBO101	Lenguaje	4	FG101	Lenguaje	1er Sem	2
CBO102	Métodos de Estudio Universitario	3	FG112	Métodos de gestión personal universitaria	1er Sem	2
CBO104	Cálculo I	4	MA111	Cálculo I	2do Sem	5
CBE004	Género y Sociedad					

Segundo Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
CBO202	Medio Ambiente y Desarrollo Sostenible	4	FG109	Ambiente y Desarrollo Sostenible	1er Sem	3
CBO201	Fundamentos de Investigación Científica	3	FG140	Fundamentos de Investigación Científica	1er Sem	3
			AI161	IA Aplicada	2do Sem	4
CBO206	Física General	4	FI101	Física General	2do Sem	5
CBO204	Cálculo II	4	MA112	Cálculo II	3er Sem	4

Tercer Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183033	Algorítmica y Fundamentos de Programación	6	CS111	Introducción a la Programación	1er Sem	5
183034	Elementos de Matemática Discreta y Computacional	4	CS1D1	Estructuras Discretas I	3er Sem	4
			CS2B1	Desarrollo Basado en Plataformas	3er Sem	4
183032	Elementos de Álgebra y Geometría	6	MA121	Algebra Lineal I	3er Sem	4
183031	Cálculo III	6	MA210	Cálculo III	4to Sem	4
183035	Inglés I					

Cuarto Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
			CS1D2	Estructuras Discretas II	4to Sem	3
183044	Estructura de Datos	6	CS210	Algoritmos y Estructuras de Datos	4to Sem	4
183042	Álgebra	6	MA122	Algebra Lineal II	4to Sem	4
183041	Cálculo IV	6	MA211	Cálculo IV	5to Sem	3
183045	Inglés II					

Quinto Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183053	Base de Datos e Ingeniería de Software	6	CS271	Bases de Datos I	4^{to} Sem	3
183054	Métodos Numéricos y Programación I	5	MA221	Análisis Numérico I	5^{to} Sem	3
183051	Análisis Real	6	MA222	Análisis Real I	5^{to} Sem	5
			MA251	Introducción a la Estadística	5^{to} Sem	3
183052	Ecuaciones Diferenciales Ordinarias	6	MA212	Ecuaciones Diferenciales Ordinarias	6^{to} Sem	6
183055	Inglés III					

Sexto Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183063	Métodos Númericos y Programación II	6	MA325	Análisis Numérico II	6^{to} Sem	4
183061	Análisis Funcional I	5	MA371	Análisis funcional I	6^{to} Sem	5
183065	Técnicas de Modelamiento	5	MA361	Modelado Matemático	7^{mo} Sem	4
183064	Sistemas Operativos	5	CS2S1	Sistemas Operativos	8^{vo} Sem	5
183062	Probabilidad e Inferencia Estadística	5	STA252	Estadística Inferencial	8^{vo} Sem	5

Séptimo Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183072	Computación Gráfica	6	CS251	Computación Gráfica	5^{to} Sem	3
183074	Inteligencia Artificial	5	CS261	Inteligencia Artificial	7^{mo} Sem	3
183071	Análisis Funcional II	5	MA372	Análisis funcional II	8^{vo} Sem	5

Octavo Semestre – Plan 2018			Plan2026			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183081	Ecuaciones Diferenciales Parciales I	6	MA213	Ecuaciones Diferenciales Parciales	7^{mo} Sem	4
183084	Proyecto de Investigación	4	MA401	Proyecto de final de carrera I	8^{vo} Sem	3
183123	Mecánica Cuántica	5	QC251	Mecánica Cuántica	8^{vo} Sem	5
			MA352	Procesos estocásticos	8^{vo} Sem	5
183083	Redes, Arquitecturas y Comunicaciones	6	CS231	Redes y Comunicación	9^{no} Sem	4

Noveno Semestre – Plan 2018			**Plan2026**			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183093	Métodos Númericos y Programación III	6	MA326	Análisis Numérico III	7^{mo} Sem	4
183094	Desarrollo del Proyecto de Investigación	5	MA402	Proyecto de final de carrera II	9^{no} Sem	5
183091	Optimización	5	MA341	Optimización I	9^{no} Sem	4
			MA355	Teoría de Aprendizaje Estadístico	9^{no} Sem	4

Décimo Semestre – Plan 2018			**Plan2026**			
Código	**Curso**	**Cr**	**Código**	**Curso**	**Sem**	**Cr**
183103	Práctica Preprofesional	4	CS400	Prácticas Pre-profesionales	10^{mo} Sem	3
183104	Presentación y Sustentación del Proyecto de Investigación	6	MA403	Proyecto de final de carrera III	10^{mo} Sem	5
			CS254	Visualización de Datos Avanzada	10^{mo} Sem	4

Capítulo 10

Laboratorios

CS111. Introducción a la Programación (Obligatorio) 1er Sem, Lab: 4 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS112. Programación Orientada a Objetos I (Obligatorio) 2do Sem, Lab: 4 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

AI161. IA Aplicada (Obligatorio) 2do Sem, Lab: 4 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS113. Programación Orientada a Objetos II (Obligatorio) 3er Sem, Lab: 2 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS1D1. Estructuras Discretas I (Obligatorio) 3er Sem, Lab: 2 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS2B1. Desarrollo Basado en Plataformas (Obligatorio) 3er Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS1D2. Estructuras Discretas II (Obligatorio) 4to Sem, Lab: 2 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS210. Algoritmos y Estructuras de Datos (Obligatorio) 4to Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS271. Bases de Datos I (Obligatorio) 4to Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.

- *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
- *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS251. Computación Gráfica (Obligatorio) 5to Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS261. Inteligencia Artificial (Obligatorio) 7mo Sem, Lab: 2 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS3P1. Computación Paralela y Distribuída (Obligatorio) 7mo Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

MA361. Modelado Matemático (Obligatorio) 7mo Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

AI263. Introducción al Aprendizaje de Máquina (Obligatorio) 8vo Sem, Lab: 4 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.
- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS2S1. Sistemas Operativos (Electivo) 8vo Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS3I1. Seguridad en Computación (Electivo) 8vo Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

CS370. Big Data y Analítica de Datos (Electivo) 8vo Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

AI264. Aprendizaje Profundo (Obligatorio) 9no Sem, Lab: 4 hr(s)

Laboratorio Básico de Computación:

- **Hardware**: 30 equipos (Intel Core i5, 8GB RAM, SSD 256GB, gráficos/sonido integrados, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Ubuntu LTS) / Windows 11 Pro (arranque dual).
- **Lenguajes**: Python, Java, C++ (C++11/14/17/20), C#, Prolog, Lex/Yacc.

- **Herramientas**: Git, VS Code, Eclipse, Jupyter Notebook.

CS231. Redes y Comunicación (Electivo) 9no Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

AI262. Procesamiento de Lenguaje Natural (Electivo) 9no Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

MA341. Optimización I (Electivo) 9no Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

AI268. Visión Computacional (Obligatorio) 10mo Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

AI367. Tópicos en Inteligencia Artificial (Obligatorio) 10mo Sem, Lab: 2 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.

QC381. Introduction a la Computación Cuántica (Electivo) 10mo Sem, Lab: 4 hr(s)

Laboratorio Avanzado de Computación:

- **Hardware**: 30 PCs de alto rendimiento (Intel Core i7, 16GB RAM, SSD NVMe 512GB, NVIDIA GTX 1660, Ethernet Gigabit).
- **Sistemas Operativos**: Linux (Fedora) / Windows 11 Pro.
- **Lenguajes**: C++17/20, Python (con librerías científicas).
- **Software**:
  - *Desarrollo*: J2EE, .NET 6, MATLAB, GCC/Clang.
  - *Bases de datos/Cloud*: PostgreSQL, MongoDB, Docker.
  - *Especializado*: Rational Rose, VTK, LEDA, emuladores 8086.