Universidad Nacional de Ingeniería (UNI)

Facultad de Ciencias

Escuela Profesional de Inteligencia Artificial

Plan Curricular 2026

– Reporte Final–

Última modificación: Abril de 2025

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 de Ingeniería (UNI).

Equipo de Trabajo

Dr. Ciro Nuñez Iturri
Director (e) de la Escuela de Ciencia de la Computación
email: *ciro.nunez.i@uni.edu.pe*

Dr. Carlos Javier Solano Salinas
Director del Departamento de Ciencia de la Computación
e-mail: *jsolano@uni.edu.pe*

Dr. José Luis Segovia-Juárez
PhD en Ciencia de la Computación. Wayne State University (Detroit, USA) (2001)
Profesor del Departamento de Ciencia de la Computacion. UNI
email: *jsegoviaj@uni.edu.pe*
http://www.tecnociencia9.com/jlsegovia/es

Dr. Marcos Antonio Alania Vicente
PhD en Ciencias Universidad de Amberes. Bélgica (2017)
Profesor del Departamento de Ciencia de la Computacion. UNI
Desarrollador de algoritmos de IA para procesos complejos interdisiplinarios
email: *alania.vicente@gmail.com*

MSc. Eduardo Yauri Lozano
MSc. en Ciencia de la Computación. UNI (2024)
Profesor del Departamento de Ciencia de la Computacion. UNI
email: *eduardo.yauri@uni.edu.pe*

MSc. Ronald Ricardo Martinez Chunga
MSc. Ciencias Ing Electrónica (2021)
Profesor del Departamento de Ciencia de la Computacion. UNI
email: *rmartinezch@uni.edu.pe*
https://www.linkedin.com/in/rmartinezchunga/

Cesar Martin Cruz-Salazar
Mag(c) en Automática e Instrumentación, UNI
Profesor de Ciencia de la Computación, UNI, Lima-Perú
email: *ccruz@uni.edu.pe*

Prof. Miguel Arrunategui Angulo
Docente del Departamento de Ciencia de la Computación
email: *garrunateguia@uni.edu.pe*

Mag. Yuri Ccoicca
Departmento de Ciencia de la Computación, UNI, Lima-Perú
email: *yccoicca@uni.edu.pe*

M.Sc. Raul Huarote Zegarra
Mag en Ciencia de la Computación, UNT (2014)
Profesor del Departamento de Ciencia de la Computación, UNI, Lima-Perú
email: *rhuarotez@uni.edu.pe*

M.Sc. Carlos Alberto Bazán Cabanillas
Magister en Ciencias, Mención matemáticas, UNI 1980
Licenciado en Matemáticas, UNI 1977
email: *cbazanc@uni.edu.pe*

Dr. 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*

Resumen ejecutivo

Todo el contenido de este documento está basado en la propuesta internacional denominada *Computing Curricula* en el área específica de Inteligencia Artificial. Este documento es el resultado de un trabajo conjunto de la Association for Computing Machinery (ACM) y la Sociedad de Computación de IEEE (IEEE-CS) y puede ser accesado a través de la dirección http://www.acm.org/education en sus versiones CS2001, CS2008, CS2013 y ahora la CS2020 que ya considera a la Inteligencia Artificial como un área emergente.

Considerando las particularidades de aplicación en nuestros países, el modelo de *Computing Curricula* fue utilizado como base para proponer el presente documento de malla curricular.

La Inteligencia Artificial, como área emergente, derivada de la Ciencia de la Computación, se suma a las siete áreas de formación profesional en computación claramente definidas en la actualidad:

- Ingeniería de Computación (*Computer Engineering* – CE) [Soldan et al., 2016],
- Ciencia de la Computación (*Computer Science* – CS) [ACM/IEEE-CS/AAAI Joint Task Force, 2023],
- Sistemas de Información (*Information Systems* – IS) [The Joint ACM/AIS IS2020 Task Force, 2020],
- Ingeniería de Software (*Software Engineering* – SE), [LeBlanc et al., 2015]
- Tecnología de la Información (*Information Technology* – IT) [ACM and IEEE-CS, 2017].
- Ciberseguridad (*Cybersecurity* – CY) [ACM/IEEE-CS Joint Task Force on Cybersecurity Education, 2017].
- Ciencia de Datos (*Data Science* – DS) [ACM Data Science Task Force, 2021].
- Inteligencia Artificial[1] (*Artificial Intelligence*).

Los pilares fundamentales considerados en esta propuesta curricular son:

- Una sólida formación profesional en el área de Inteligencia Artificial,
- Preparación para contribuir al desarrollo de empresas basadas en tecnología,
- Una formación ética sólida y proyección hacia la sociedad.

Estos pilares redundarán en la formación de profesionales capaces de desempeñarse en cualquier parte del mundo y de contribuir de manera significativa al avance de la industria de la Inteligencia Artificial en nuestro país.

El resto de este documento está organizado de la siguiente manera: el Capítulo 1 define y explica el campo de acción de la Inteligencia Artificial, además de proporcionar una breve explicación de las diferentes áreas de estudio propuestas por IEEE-CS y ACM.

El Capítulo line:63 **??** muestra las áreas de conocimiento de la Inteligencia Artificial, tomando como su base fundamental a la Ciencia de la Computación indicando los temas y objetivos de aprendizaje correspondientes a cada una.

El Capítulo 2 contiene la distribución de los cursos por semestre, por áreas, por niveles, comparaciones con propuestas internacionales, materia por habilidades y distribución de temas por curso.

El Capítulo 3 contiene información detallada sobre cada curso, incluidas las habilidades que contribuye a desarrollar, la bibliografía correspondiente y el número de horas mínimo por unidad.

En el Capítulo 4 se presenta una propuesta de laboratorios necesarios para cubrir el dictado de clases.

En el Capítulo 5 se ofrece un listado de cursos por profesor y de profesores por curso.

[1] Área emergente

Índice general

Agradecimientos xiii

Abreviaturas xv

1. Introducción 1
1.1. Definiciones básicas . . . 1
1.2. Campo y mercado ocupacional . . . 3
1.3. Importancia de la carrera en la sociedad . . . 4
1.4. Información institutional . . . 4
1.4.1. Misión . . . 4
1.4.2. Visión . . . 4
1.5. Información de acreditacion . . . 5
1.5.1. Resultados de la carrera *Outcomes* . . . 5
1.5.2. Objetivos educacionales . . . 5
1.5.3. Aprendizaje esperado (*Learning Outcomes*) . . . 6
1.6. Perfiles . . . 6
1.6.1. Perfil del estudiante . . . 6
1.6.2. Perfil del graduado . . . 7
1.6.3. Perfil Profesional . . . 7
1.6.4. Perfil del docente . . . 8
1.7. Grados y Títulos . . . 8
1.8. Recursos para dictado de clases . . . 8
1.9. Organización del documento . . . 8

2. Plan de estudios 11
2.1. Codificación de los cursos . . . 11
2.2. Estructura Curricular . . . 12
2.3. Tópicos distribuídos por curso . . . 17
2.4. Resultados esperados distribuídos por curso . . . 17
2.5. Cursos por resultado (*Outcome*) . . . 24
2.5.1. Outcome: AG-C01) Impacto social de soluciones computacionales. . . . 24
2.5.2. Outcome: AG-C02) Ética profesional en computación. . . . 24
2.5.3. Outcome: AG-C03) Trabajo individual y en equipo. . . . 24
2.5.4. Outcome: AG-C04) Comunicación efectiva en computación. . . . 25
2.5.5. Outcome: AG-C05) Gestión de proyectos en computación. . . . 25
2.5.6. Outcome: AG-C06) Aprendizaje continuo en computación. . . . 25
2.5.7. Outcome: AG-C07) Conocimientos de computación. . . . 25
2.5.8. Outcome: AG-C08) Análisis de problemas computacionales. . . . 26
2.5.9. Outcome: AG-C09) Diseño y desarrollo de soluciones. . . . 26
2.5.10. Outcome: AG-C10) Indagación en computación. . . . 27
2.5.11. Outcome: AG-C11) Uso de herramientas computacionales. . . . 27
2.5.12. Outcome: AG-C12) Gestión de tecnologías seguras de computación. . . . 27
2.6. Distribución de cursos en la carrera . . . 28
2.7. Departamentos . . . 30

2.7.1. DAISI . . . 30
2.7.2. DLL . . . 31
2.8. Compatibilidad de la carrera con relación a estandares internacionales . . . 32

3. Contenido detallado por curso **49**
Primer Semestre . . . 49
BIC101 Introducción a la Computación (Obligatorio) . . . 49
BMA101 Álgebra Lineal (Obligatorio) . . . 54
BMA102 Cálculo Diferencial (Obligatorio) . . . 57
BQU101 Química I (Obligatorio) . . . 60
BEI101 Inglés I (Obligatorio) . . . 65
Segundo Semestre . . . 69
CS100 Introducción a la Ciencia de la Computación (Obligatorio) . . . 69
CS112 Programación Orientada a Objetos I (Obligatorio) . . . 80
CS1D1 Estructuras Discretas (Obligatorio) . . . 85
BMA103 Cálculo Integral (Obligatorio) . . . 88
BFI101 Física I (Obligatorio) . . . 90
BEI102 Inglés II (Obligatorio) . . . 95
Tercer Semestre . . . 99
CS113 Programación Orientada a Objetos II (Obligatorio) . . . 99
CS2B1 Desarrollo Basado en Plataformas (Obligatorio) . . . 105
AI161 IA Aplicada (Obligatorio) . . . 109
STA251 Estadística y Probabilidades (Obligatorio) . . . 113
FG106 Teatro (Obligatorio) . . . 118
BEI201 Inglés III (Obligatorio) . . . 122
EC101 Economía General (Electivo) . . . 126
FG120 Realidad Nacional (Electivo) . . . 129
Cuarto Semestre . . . 132
CS210 Algoritmos y Estructuras de Datos (Obligatorio) . . . 132
CS211 Teoría de la Computación (Obligatorio) . . . 137
CS221 Arquitectura de Computadores (Obligatorio) . . . 142
CS271 Bases de Datos I (Obligatorio) . . . 147
CS401 Metodología de la Investigación (Obligatorio) . . . 152
AI261 Probabilidad y Estadistica para IA (Obligatorio) . . . 156
FG350 Liderazgo y Desempeño (Obligatorio) . . . 159
BEI202 Inglés IV (Electivo) . . . 162
Quinto Semestre . . . 166
CS212 Análisis y Diseño de Algoritmos (Obligatorio) . . . 166
CS261 Inteligencia Artificial (Obligatorio) . . . 171
CS291 Ingeniería de Software I (Obligatorio) . . . 178
CS2S1 Sistemas Operativos (Obligatorio) . . . 184
AI162 Política y gobernanza de la IA (Obligatorio) . . . 189
MA202 Métodos Numéricos (Obligatorio) . . . 191
BEI203 Inglés V (Electivo) . . . 196
Sexto Semestre . . . 200
CS311 Algoritmos para Problemas Complejos (Obligatorio) . . . 200
AI262 Procesamiento de Lenguaje Natural (Obligatorio) . . . 203
AI263 Introducción a Machine Learning (Obligatorio) . . . 206
AI267 Interaccion Humano-IA (Obligatorio) . . . 209
MA307 Matemática aplicada a la computación (Obligatorio) . . . 214
FG211 Ética Profesional (Obligatorio) . . . 217
Séptimo Semestre . . . 220
CS231 Redes y Comunicación (Obligatorio) . . . 220
CS251 Computación Gráfica (Obligatorio) . . . 225
AI264 Aprendizaje Profundo (Obligatorio) . . . 228
AI265 Aprendizaje por Refuerzo (Obligatorio) . . . 231

AI266 IA para Ingenieria de Software (Obligatorio) . . . 234
AI400 Metodología de la Investigación (Obligatorio) . . . 236
Octavo Semestre . . . 239
CS402 Proyecto de tesis 1 (Obligatorio) . . . 239
AI362 Inteligencia Artificial para edge computing (Obligatorio) . . . 243
AI364 Inteligencia Artificial para Ciberseguridad I (Obligatorio) . . . 245
AI366 Sistemas de IA Distribuidos (Obligatorio) . . . 247
AI401 Proyecto de tesis 1 (Obligatorio) . . . 249
BI101 Biología (Obligatorio) . . . 253
EX301 Actividades Extracurriculares (Obligatorio) . . . 256
Noveno Semestre . . . 258
CS3I1 Seguridad en Computación (Obligatorio) . . . 258
CS3P1 Computación Paralela y Distribuída (Obligatorio) . . . 264
CS400 Prácticas Pre-profesionales (Obligatorio) . . . 274
CS403 Proyecto de tesis 2 (Obligatorio) . . . 276
AI402 Proyecto de tesis 2 (Obligatorio) . . . 280
CB309 Bioinformática (Obligatorio) . . . 284
AI268 Visión Computacional (Electivo) . . . 288
AI363 Sistemas Autónomos (Electivo) . . . 290
AI365 Modelos Generativos Avanzados en IA (Electivo) . . . 292
Décimo Semestre . . . 294
CS404 Taller de Investigación (Obligatorio) . . . 294
AI403 Taller de Investigación (Obligatorio) . . . 298
AI369 Robótica (Electivo) . . . 302
CS281 Computación en la Sociedad (Electivo) . . . 306
CS351 Tópicos en Computación Gráfica (Electivo) . . . 312
CS353 Computación Cuántica (Electivo) . . . 316
CS392 Tópicos en Ingeniería de Software (Electivo) . . . 320
CS393 Sistemas de Infomación (Electivo) . . . 327
CS3P3 Internet de las Cosas (Electivo) . . . 330

4. Laboratorios **333**

5. Profesores & Cursos **345**
Cursos asignados por profesor . . . 345
Profesor asignado por curso . . . 345

Índice de figuras

1.1. Campo acción de la Inteligencia Artificial 2

2.1. Esquema de codificación para los cursos. 11
2.2. Créditos por área por semestre 29
2.3. Distribución de cursos por áreas considerando creditaje. 30
2.4. Distribución de créditos por niveles de cursos. 30
2.5. DAISI 31
2.6. DLL 32
2.7. Comparación de AI-UNI con CS de ACM/IEEE-CS (Gráfico de curvas). 35
2.8. Comparación de AI-UNI con CS de ACM/IEEE-CS (Gráfico de araña). 35
2.9. Comparación de AI-UNI con DS de ACM/IEEE-CS (Gráfico de curvas). 36
2.10. Comparación de AI-UNI con DS de ACM/IEEE-CS (Gráfico de araña). 37
2.11. Comparación de AI-UNI con SE de ACM/IEEE-CS (Gráfico de curvas). 38
2.12. Comparación de AI-UNI con SE de ACM/IEEE-CS (Gráfico de araña). 39
2.13. Comparación de AI-UNI con CY de ACM/IEEE-CS (Gráfico de curvas). 40
2.14. Comparación de AI-UNI con CY de ACM/IEEE-CS (Gráfico de araña). 41
2.15. Comparación de AI-UNI con CE de ACM/IEEE-CS (Gráfico de curvas). 42
2.16. Comparación de AI-UNI con CE de ACM/IEEE-CS (Gráfico de araña). 43
2.17. Comparación de AI-UNI con IT de ACM/IEEE-CS (Gráfico de curvas). 44
2.18. Comparación de AI-UNI con IT de ACM/IEEE-CS (Gráfico de araña). 45
2.19. Comparación de AI-UNI con IS de ACM/IEEE-CS (Gráfico de curvas). 46
2.20. Comparación de AI-UNI con IS de ACM/IEEE-CS (Gráfico de araña). 47

3.1. Mapa de Conexión. BIC101 Introducción a la Computación 49
3.2. Mapa de Conexión. BMA101 Álgebra Lineal 54
3.3. Mapa de Conexión. BMA102 Cálculo Diferencial 57
3.4. Mapa de Conexión. BQU101 Química I 60
3.5. Mapa de Conexión. BEI101 Inglés I 65
3.6. Mapa de Conexión. CS100 Introducción a la Ciencia de la Computación 69
3.7. Mapa de Conexión. CS112 Programación Orientada a Objetos I 80
3.8. Mapa de Conexión. CS1D1 Estructuras Discretas 85
3.9. Mapa de Conexión. BMA103 Cálculo Integral 88
3.10. Mapa de Conexión. BFI101 Física I 90
3.11. Mapa de Conexión. BEI102 Inglés II 95
3.12. Mapa de Conexión. CS113 Programación Orientada a Objetos II 99
3.13. Mapa de Conexión. CS2B1 Desarrollo Basado en Plataformas 105
3.14. Mapa de Conexión. AI161 IA Aplicada 109
3.15. Mapa de Conexión. STA251 Estadística y Probabilidades 113
3.16. Mapa de Conexión. FG106 Teatro 118
3.17. Mapa de Conexión. BEI201 Inglés III 122
3.18. Mapa de Conexión. EC101 Economía General 126
3.19. Mapa de Conexión. FG120 Realidad Nacional 129
3.20. Mapa de Conexión. CS210 Algoritmos y Estructuras de Datos 132
3.21. Mapa de Conexión. CS211 Teoría de la Computación 137

3.22. Mapa de Conexión. CS221 Arquitectura de Computadores . . . 142
3.23. Mapa de Conexión. CS271 Bases de Datos I . . . 147
3.24. Mapa de Conexión. CS401 Metodología de la Investigación . . . 152
3.25. Mapa de Conexión. AI261 Probabilidad y Estadistica para IA . . . 156
3.26. Mapa de Conexión. FG350 Liderazgo y Desempeño . . . 159
3.27. Mapa de Conexión. BEI202 Inglés IV . . . 162
3.28. Mapa de Conexión. CS212 Análisis y Diseño de Algoritmos . . . 166
3.29. Mapa de Conexión. CS261 Inteligencia Artificial . . . 171
3.30. Mapa de Conexión. CS291 Ingeniería de Software I . . . 178
3.31. Mapa de Conexión. CS2S1 Sistemas Operativos . . . 184
3.32. Mapa de Conexión. AI162 Política y gobernanza de la IA . . . 189
3.33. Mapa de Conexión. MA202 Métodos Numéricos . . . 191
3.34. Mapa de Conexión. BEI203 Inglés V . . . 196
3.35. Mapa de Conexión. CS311 Algoritmos para Problemas Complejos . . . 200
3.36. Mapa de Conexión. AI262 Procesamiento de Lenguaje Natural . . . 203
3.37. Mapa de Conexión. AI263 Introducción a Machine Learning . . . 206
3.38. Mapa de Conexión. AI267 Interaccion Humano-IA . . . 209
3.39. Mapa de Conexión. MA307 Matemática aplicada a la computación . . . 214
3.40. Mapa de Conexión. FG211 Ética Profesional . . . 217
3.41. Mapa de Conexión. CS231 Redes y Comunicación . . . 220
3.42. Mapa de Conexión. CS251 Computación Gráfica . . . 225
3.43. Mapa de Conexión. AI264 Aprendizaje Profundo . . . 228
3.44. Mapa de Conexión. AI265 Aprendizaje por Refuerzo . . . 231
3.45. Mapa de Conexión. AI266 IA para Ingenieria de Software . . . 234
3.46. Mapa de Conexión. AI400 Metodología de la Investigación . . . 236
3.47. Mapa de Conexión. CS402 Proyecto de tesis 1 . . . 239
3.48. Mapa de Conexión. AI362 Inteligencia Artificial para edge computing . . . 243
3.49. Mapa de Conexión. AI364 Inteligencia Artificial para Ciberseguridad I . . . 245
3.50. Mapa de Conexión. AI366 Sistemas de IA Distribuidos . . . 247
3.51. Mapa de Conexión. AI401 Proyecto de tesis 1 . . . 249
3.52. Mapa de Conexión. BI101 Biología . . . 253
3.53. Mapa de Conexión. EX301 Actividades Extracurriculares . . . 256
3.54. Mapa de Conexión. CS3I1 Seguridad en Computación . . . 258
3.55. Mapa de Conexión. CS3P1 Computación Paralela y Distribuída . . . 264
3.56. Mapa de Conexión. CS400 Prácticas Pre-profesionales . . . 274
3.57. Mapa de Conexión. CS403 Proyecto de tesis 2 . . . 276
3.58. Mapa de Conexión. AI402 Proyecto de tesis 2 . . . 280
3.59. Mapa de Conexión. CB309 Bioinformática . . . 284
3.60. Mapa de Conexión. AI268 Visión Computacional . . . 288
3.61. Mapa de Conexión. AI363 Sistemas Autónomos . . . 290
3.62. Mapa de Conexión. AI365 Modelos Generativos Avanzados en IA . . . 292
3.63. Mapa de Conexión. CS404 Taller de Investigación . . . 294
3.64. Mapa de Conexión. AI403 Taller de Investigación . . . 298
3.65. Mapa de Conexión. AI369 Robótica . . . 302
3.66. Mapa de Conexión. CS281 Computación en la Sociedad . . . 306
3.67. Mapa de Conexión. CS351 Tópicos en Computación Gráfica . . . 312
3.68. Mapa de Conexión. CS353 Computación Cuántica . . . 316
3.69. Mapa de Conexión. CS392 Tópicos en Ingeniería de Software . . . 320
3.70. Mapa de Conexión. CS393 Sistemas de Infomación . . . 327
3.71. Mapa de Conexión. CS3P3 Internet de las Cosas . . . 330

Índice de cuadros

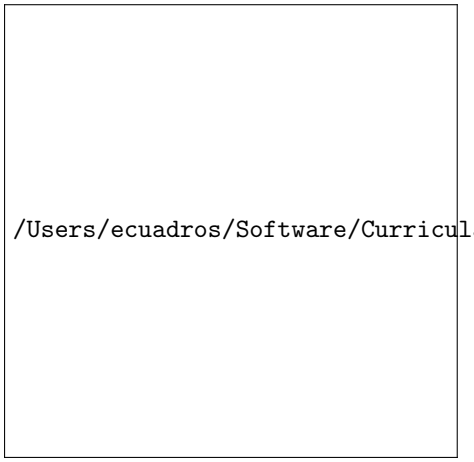
2.1. Resultados esperados por curso 1er al 2do Semestre 17
2.2. Resultados esperados por curso 3er Semestre 18
2.3. Resultados esperados por curso 4to al 5to Semestre 19
2.4. Resultados esperados por curso 6to al 7mo Semestre 20
2.5. Resultados esperados por curso 8vo Semestre 21
2.6. Resultados esperados por curso 9no Semestre 22
2.7. Resultados esperados por curso 10mo Semestre 23
2.8. Distribución de créditos por área por semestre 28

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).
- Cuerpo docente de: CS-UCSP, CS-UNSA, CS-UTEC, CS-UNI.

De forma muy especial deseamos expresar nuestro agradecimiento al Patronato de la Universidad Nacional de Ingeniería (UNI) por su apoyo y auspicio en el desarrollo de esta propuesta de Malla Curricular, así como por su compromiso constante con la mejora de la formación académica de sus estudiantes.



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: **Computación**.

Abreviaturas

ACM *Association for Computing Machinery*

AIS *Association for Information Systems*

CC Ciencia de la Computación

CS Ciencia de la Computación – *Computer Science*

IEEE-CS IEEE *Computer Society*

.

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 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 del país.

1.1. Definiciones básicas

La referencia más sólida a nivel mundial en cuanto a la propuesta de carreras de computación para nivel de pregrado es la que fue propuesta en conjunto por la *Association for Computing Machinery* (ACM), IEEE *Computer Society* (IEEE-CS) y la *Association for Information Systems* (AIS). Estas tres organizaciones propusieron la Computing Curricula en el documento denominado: *Joint Task Force for Computing Curricula 2005, Computing Curricula 2005. Overview Report*[Shakelford et al., 2005] cuya version mas reciente es la denominada *Computing Curricula 2020*[ACM/IEEE-CS, 2020].

La Ciencia de la Computación es un término de origen estadounidense Ciencia de la Computación – *Computer Science* (CS). Este término es conocido también como informática en el ámbito europeo[1].

Según el diccionario de la Real Academia de la Lengua Española (http://www.rae.es) ambos términos también son sinónimos.

A nivel internacional, la computación presenta siete perfiles claramente definidos:

- Ciencia de la Computación (*Computer Science*) [ACM/IEEE-CS/AAAI Joint Task Force, 2023],
- Ingeniería de Computación (*Computer Engineering*) [Soldan et al., 2016],
- Sistemas de Información (*Information Systems*) [The Joint ACM/AIS IS2020 Task Force, 2020],
- Ingeniería de Software (*Software Engineering*) [Díaz-Herrera and Hilburn, 2004],
- Tecnología de la Información (*Information Technology*) [ACM and IEEE-CS, 2017],
- Ciencia de Datos (*Data Science*) [ACM Data Science Task Force, 2021],
- Ciberseguridad (*Cybersecurity*) [ACM/IEEE-CS Joint Task Force on Cybersecurity Education, 2017].

La Figura 1.1 es tomada de la definición propuesta en la *Computing Curricula*[Shakelford et al., 2005] en el área de Ciencia de la Computación (CC) La CC cubre la mayor parte entre el extremo superior y el extremo inferior, porque el profesional en CC no trata "solamente con el hardware" que utiliza un software o de "solamente la organización" que hace uso de la información que la computación le puede proveer.

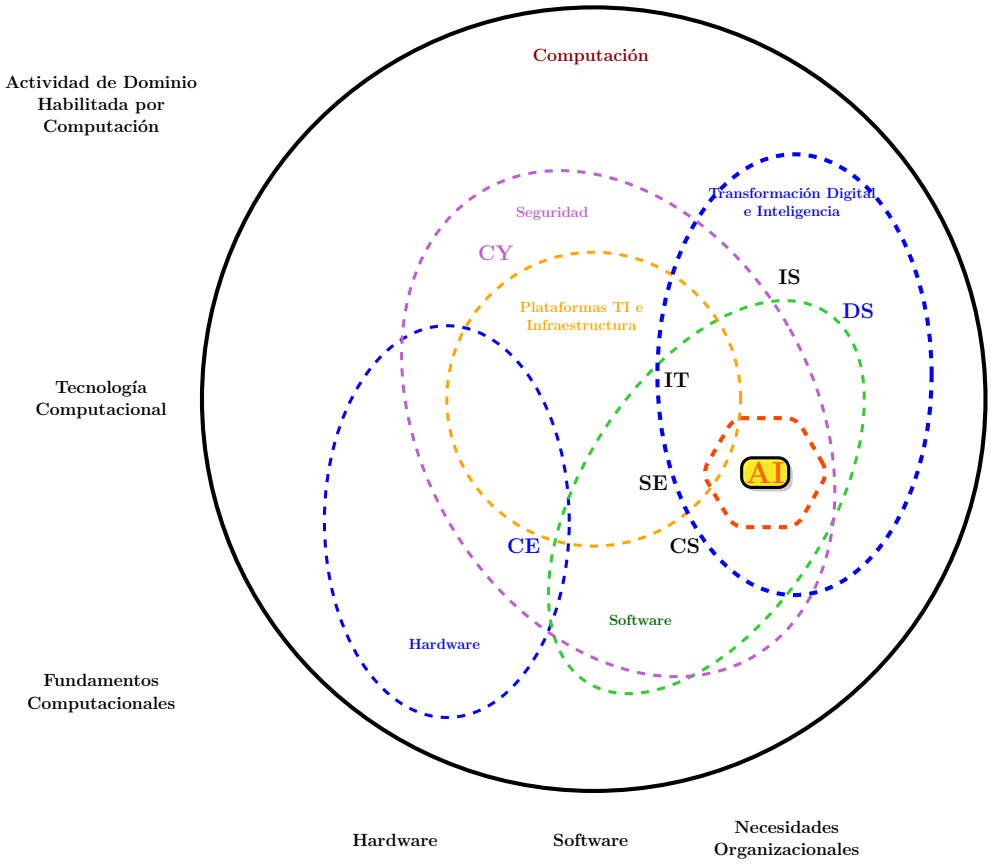


Figura 1.1: Campo acción de la Inteligencia Artificial

Ciencia de la Computación cubren un amplio rango, desde sus fundamentos teóricos y algorítmicos hasta los últimos desarrollos en robótica, visión por computadora, Inteligencia Artificial, bioinformática, y otras áreas emocionantes. Podemos pensar que el trabajo de un científico de la computación pertenece a las siguientes tres categorías:

- **Diseño e implementación de software**. Los científicos de computación se encargan de desafiantes labores de programación. También supervisan otros programadores, haciéndolos concientes de nuevas aproximaciones.

[1] El término europeo es derivado del vocablo francés *Informatique*.

- **Instrumentación de nuevas formas para usar computadoras**. El progreso en las áreas de ciencias de la computación como redes, bases de datos, e interfaces humano-computadora permitieron el desarrollo de la www y actualmente se trabaja en el desarrollo de metasistemas Grid. Además, los investigadores trabajan ahora en hacer que los robots sean ayudantes prácticos y demuestren inteligencia, utilizan las bases de datos para crear nuevos conocimientos, y están utilizando computadoras para decifrar los secretos de nuestro ADN.
- **Desarrollo de formas efectivas de resolver problemas de computación.** Por ejemplo, los científicos de la computación desarrollan las mejores formas posibles de almacenar información en bases de datos, enviar datos a través de la red, y desplegar imágenes complejas. Sus bases teóricas les permiten determinar el mejor desempeño posible, y su estudio de algoritmos les ayuda a desarrollar nuevas aproximaciones para proveer un mejor desempeño.

La Ciencia de la Computación cubren todo el rango desde la teoría hasta la programación. Mientras otras disciplinas pueden producir titulados mejor preparados para trabajos específicos, las ciencias de la computación ofrecen un amplio fundamento que permite a sus titulados adaptarse a nuevas tecnologías y nuevas ideas.

El profesional en CC se preocupa por casi todo en medio de estas áreas. En dirección hacia el hardware, este profesional llega a desarrollar software que permite el funcionamiento de dispositivos *devices*. En dirección a aspectos organizacionales, el profesional de CC ayuda a que los sistemas de información operen correctamente en las organizaciones. Él genera la tecnología que permite que otras áreas como los sistemas de información se desarrollen adecuadamente.

El profesional en CC diseña y desarrolla todo tipo de software, desde infraestructura de plataformas (sistemas operativos, programas de comunicación, etc.) hasta aplicación de tecnologías (navegadores de Internet, bases de datos, motores de búsqueda, etc.). Este profesional crea estas capacidades, pero no está orientado al uso de las mismas. Por lo tanto, el área sombreada (fig. 1.1) para CC se estrecha y finaliza en la medida que nos movamos hacia la aplicación y configuración de productos.

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 al desarrollo de sistemas de Inteligencia Artificial con calidad internacional.
- Empresas, instituciones y organizaciones que requieran soluciones basadas en Inteligencia Artificial 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 soluciones basadas en Inteligencia Artificial. Considerando que países en vias de desarrollo tienen un costo de vida mucho menor que en América del Norte o Europa, una posibilidad que se muestra interesante es la exportación de soluciones de Inteligencia Artificial, 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 soluciones basadas en Inteligencia Artificial en nuestro país y venderlas fuera es más rentable que salir al extranjero y comercializarlas allá.

El campo ocupacional de un egresado es amplio y está en continua expansión y cambio. Prácticamente toda empresa u organización hace uso de tecnologías de Inteligencia Artificial 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 solo 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 aplicación de la Inteligencia Artificial en 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 considerando a mediano y largo plazo. El campo para las tareas de investigación y desarrollo de problemas complejos en Inteligencia Artificial es también muy amplio y está creciendo día a día a nivel mundial.

Debido a la capacidad innovadora de nuestro egresado, existe una mayor 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 Inteligencia Artificial es que se dedique a desarrollar Inteligencia Artificial o que se integre a las empresas que producen soluciones basadas en IA. En el ámbito de la IA, es común observar que los países cuentan con Asociaciones de Productores de Software con políticas orientadas a la exportación. Por lo tanto, es esencial preparar a nuestros alumnos para el mercado global.

Nuestros futuros profesionales deben estar orientados a crear nuevas empresas de base tecnológica que puedan incrementar las exportaciones de soluciones basadas en IA peruanas. Este nuevo perfil está dirigido a generar una industria innovadora. Si somos capaces de exportar soluciones de IA competitivas, también estaremos en condiciones de atraer nuevas inversiones. Estas inversiones generarían más empleos bien remunerados y con un costo bajo en comparación con otros sectores industriales. Bajo esta perspectiva, la carrera de IA será un motor que impulsará el desarrollo del país de manera decisiva con una inversión relativamente baja.

Es importante recordar que muchas de las innovaciones en productos comerciales recientes utilizan tecnologías que eran conocidas en el mundo académico hace décadas. Por lo tanto, es esencial que los profesionales de IA estén actualizados en los avances tecnológicos y sean capaces de generar nuevas tecnologías, no solo de utilizar las existentes.

Además, nuestros alumnos deben estar preparados para resolver los problemas futuros que enfrentará la sociedad en los próximos años. En un mundo en constante evolución, la investigación en IA es fundamental para anticiparse a los desafíos y ofrecer soluciones innovadoras.

1.4. Información institutional

1.4.1. Misión

Por lo antes mencionado, nuestra misión es:

Contribuir al desarrollo científico, tecnológico y técnico del país formando profesionales competentes, orientados a la creación de nueva ciencia y tecnología computacional, como motor que impulse y consolide la industria del software en base a la investigación científica y tecnológica en áreas innovadoras formando, EN NUESTROS profesionales, un conjunto de habilidades y destrezas para la solución de problemas computacionales con un compromiso social.

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 Inteligencia Artificial 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*):

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad.

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación.

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos.

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación.

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos.

AG-C06) Aprendizaje a lo largo de la vida: Reconoce la importancia del aprendizaje continuo y la adaptación a nuevas tecnologías.

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación.

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación.

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación.

AG-C10) Indagación: Estudia problemas complejos de computación usando métodos de ciencias de la información.

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas.

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos.

Las competencias y/o resultados propuestas en esta malla curricular corresponden a las competencias y/o resultados propuestas por Instituto de Acreditación de Carreras de Computación, Ingeniería y Tecnología (ICACIT).

1.5.2. Objetivos educacionales

Después de cinco años de egresado de la carrera profesional de Inteligencia Artificial, 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 Inteligencia Artificial.
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 Inteligencia Artificial 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 Inteligencia Artificial 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 y los resultados de aprendizaje (*Learning Outcomes*) que los estudiantes deben alcanzar en lo que respecta a los temas especificados. 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 los resultados del aprendizaje asociados.

Cada resultado de aprendizaje tiene un nivel asociado de dominio. En la definición de los diferentes niveles que dibujamos de otros enfoques curriculares, especialmente la taxonomía de Bloom, que ha sido bien explorada dentro de la Inteligencia Artificial. En este documento no se aplicó directamente los niveles de Bloom en parte porque varios de ellos son impulsados por contexto pedagógico, que introduciría demasiada pluralidad en un documento de este tipo; en parte porque tenemos la intención de los niveles de dominio para ser indicativa y no imponer restricción teórica 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 iteración en el desarrollo de software (for, while e iteradores). En el plano de la "familiaridad", se espera que un estudiante tenga una definición del concepto de iteración en el desarrollo de software y saber por qué esta técnica es útil.

Con el fin de mostrar el dominio del nivel "Uso", el estudiante debe ser capaz de escribir un programa adecuadamente usando una forma de iteración.

En el nivel de "Evaluación", en la iteración se requeriría que un estudiante comprenda múltiples métodos de iteración y que sea capaz de seleccionar apropiadamente entre ellos para diferentes aplicaciones.

1.6. Perfiles

1.6.1. Perfil del estudiante

El aspirante a ingresar a la Escuela Profesional de Inteligencia Artificial de la Universidad Nacional de Ingeniería (UNI) debe tener:
Conocimientos de:

1. Conceptos fundamentales de álgebra, geometría, trigonometría y análisis matemático elemental.
2. Nociones básicas de lógica proposicional y razonamiento deductivo.
3. Principios elementales de algoritmos y secuencias lógicas para la resolución de problemas.
4. Nociones de probabilidad y estadística descriptiva para el análisis de datos.

Habilidades para:

1. Pensar de forma abstracta, identificando patrones, estructuras y relaciones más allá de lo concreto.
2. Modelar problemas del mundo real utilizando representaciones simbólicas y formales.
3. Descomponer problemas complejos en subproblemas más simples y manejables.
4. Aprender de manera autónoma y autorregulada, adquiriendo nuevas tecnologías y paradigmas sin necesidad de instrucción formal continua.
5. Adaptarse rápidamente a entornos tecnológicos cambiantes y a la evolución vertiginosa de herramientas y lenguajes.
6. Comunicar ideas técnicas y soluciones de forma clara, tanto oralmente como por escrito, utilizando terminología apropiada.

7. Mantener la concentración en tareas analíticas que demanden esfuerzo sostenido y atención al detalle.
8. Sintetizar información de múltiples fuentes y extraer conceptos esenciales para la toma de decisiones.
9. Evaluar críticamente soluciones existentes y proponer mejoras o alternativas innovadoras.

Actitudes de:

1. Interés genuino y curiosidad por la tecnología, la programación y el funcionamiento de los sistemas computacionales.
2. Disposición para el trabajo colaborativo en equipos de desarrollo y proyectos multidisciplinarios.
3. Iniciativa para explorar problemas más allá del currículo y perseverancia ante desafíos intelectuales como la depuración de código o la optimización.
4. Mentalidad de crecimiento y apertura para aprender continuamente en un campo que evoluciona permanentemente.
5. Compromiso con la calidad, la eficiencia y las buenas prácticas en el desarrollo de soluciones.

1.6.2. Perfil del graduado

El graduado del Programa Profesional de Inteligencia Artificial es capaz de:

1. Diseñar, desarrollar e implementar sistemas de inteligencia artificial y aprendizaje automático que resuelvan problemas complejos en diversos dominios aplicados (visión por computadora, procesamiento de lenguaje natural, robótica).
2. Aplicar fundamentos matemáticos y estadísticos avanzados (álgebra lineal, cálculo, probabilidad, optimización) para el diseño y análisis teórico de algoritmos de IA y ML.
3. Implementar arquitecturas de deep learning (redes neuronales convolucionales, recurrentes, transformadores) utilizando frameworks modernos (TensorFlow, PyTorch) y técnicas de entrenamiento eficiente a escala.
4. Desarrollar pipelines de datos robustos para el preprocesamiento, limpieza y aumento de conjuntos de datos complejos y masivos, aplicando principios de ingeniería de software para sistemas de IA.
5. Investigar, evaluar y adaptar algoritmos y modelos del estado del arte, comprendiendo sus fundamentos teóricos, limitaciones y potencial de mejora.
6. Diseñar y desplegar sistemas de IA en producción, considerando aspectos de escalabilidad, latencia, monitoreo de modelos (MLOps) y consumo eficiente de recursos computacionales.
7. Analizar críticamente las implicaciones éticas, sociales y de sesgo algorítmico en sistemas de IA, e implementar mecanismos para garantizar transparencia, equidad y privacidad (XAI, AI Ethics).
8. Colaborar en equipos multidisciplinarios para integrar soluciones de IA en productos, procesos o servicios existentes, comunicando conceptos técnicos a expertos de dominio no técnicos.
9. Crear y gestionar proyectos de emprendimiento basados en IA, identificando oportunidades de mercado, desarrollando prototipos y construyendo modelos de negocio tecnológicamente viables.
10. Continuar estudios avanzados (maestría, doctorado) en IA/ML en instituciones de prestigio mundial, contribuyendo a la frontera del conocimiento mediante investigación de vanguardia y publicación académica.

1.6.3. Perfil Profesional

El profesional en Inteligencia Artificial tiene las siguientes características:

- Diseñar y desarrollar modelos de ciencia de la computacion, aprendizaje automático, aprendizaje profundo (*deep learning*) y sistemas inteligentes basados en fundamentos matemáticos.
- Implementar arquitecturas de redes neuronales, algoritmos de optimización y técnicas de procesamiento de datos para resolver problemas complejos de clasificación, regresión y clustering.
- Trabajar con grandes volúmenes de datos estructurados y no estructurados, aplicando pipelines de preprocesamiento, tratamiento de características y técnicas de validación.
- Evaluar y seleccionar frameworks de IA (TensorFlow, PyTorch, Scikit-learn) para construir soluciones escalables y eficientes en recursos computacionales.

- Aplicar principios de ética en IA, detectar sesgos en modelos y garantizar transparencia, explicabilidad y cumplimiento normativo en sistemas automatizados.
- Colaborar en equipos multidisciplinarios con expertos en dominio para desarrollar soluciones innovadoras en salud, robótica, visión computacional y lenguaje natural.
- Optimizar modelos de IA para despliegue en producción, implementando MLOps, monitorización continua y reentrenamiento automatizado.
- Comunicar resultados técnicos complejos a audiencias no técnicas, justificando decisiones de diseño de modelos y trade-offs entre precisión y eficiencia.
- Mantenerse actualizado en avances de IA generativa, transformadores, reinforcement learning y agentes autónomos, evaluando su adopción estratégica.
- Crear *spin-offs* de investigación o startups de base tecnológica, transformando investigación académica en productos de mercado con impacto social.

1.6.4. Perfil del docente

El docente de la carrera de Inteligencia Artificial 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 Ciencia de la Computación y

Titulo Profesional: Licenciado en Ciencia de la Computación

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 4.

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 Inteligencia Artificial, además se hace una muy breve explicación de las distintas

carreras relacionadas a esta disciplina. En el caso particular del area de computación, las propuestas de ACM/IEEE-CS *Computing Curricula* [ACM/IEEE-CS, 2020].

El Capítulo 2 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 3 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 5 podemos observar un listado de cursos por profesor y de profesor por curso.

En el Capítulo 4 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

Plan de estudios

2.1. 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: CS403

Los cursos se encuentran codificados bajo el esquema que se muestra en la Figura 2.1.

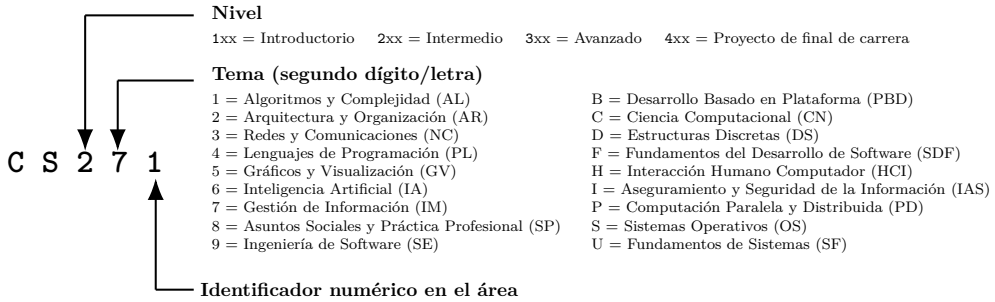


Figura 2.1: Esquema de codificación para los cursos.

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. **BIC** Área de Computación (*Computer Science* – CS);
3. **AI** Área de Inteligencia Artificial (*Artificial Intelligence* – AI);
4. **MA** Matemáticas (MA);
5. **BMA** Matemáticas (BMA);
6. **STA** Estadística (ST);
7. **CB** Área de Ciencias Básicas (Física, Biología);
8. **BFI** Ciencias Físicas
9. **BQU** Ciencias Químicas (BQU);
10. **EC** Economía (EC);
11. **BI** Área de Ciencias Básicas (Física, Biología)
12. **FG** Área de Formación General;
13. **EX** Actividades extracurriculares
14. **BEI** Área de Idiomas

Al mismo tiempo, los cursos están clasificados en:

- Estudios generales
- Específicos
- Especialización

2.2. 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**
BIC101	Introducción a la Computación (Pág 49)	Específicos	2	4	4	O	
BMA101	Álgebra Lineal (Pág 54)	Estudios generales	3	2	4	O	
BMA102	Cálculo Diferencial (Pág 57)	Estudios generales	4	2	5	O	
BQU101	Química I (Pág 60)	Estudios generales	4	2	5	O	
BEI101	Inglés I (Pág 65)	Estudios generales		4	2	O	
					20		

Segundo Semestre							
Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
CS100	Introducción a la Ciencia de la Computación (Pág 69)	Específicos	2	2	3	O	
CS112	Programación Orientada a Objetos I (Pág 80)	Específicos	2	4	4	O	
CS1D1	Estructuras Discretas (Pág 85)	Específicos	2	4	4	O	
BMA103	Cálculo Integral (Pág 88)	Estudios generales	4	2	5	O	BMA102 (1er Sem)
BFI101	Física I (Pág 90)	Estudios generales	4	4	5	O	
BEI102	Inglés II (Pág 95)	Estudios generales		4	2	O	BEI101 (1er Sem)
					23		

	Tercer Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS113	Programación Orientada a Objetos II (Pág 99)	Específicos	2	4	4	O	CS112 (2do Sem)
	CS2B1	Desarrollo Basado en Plataformas (Pág 105)	Específicos	2	4	4	O	CS112 (2do Sem)
	AI161	IA Aplicada (Pág 109)	Específicos	2	4	4	O	CS100 (2do Sem)
	STA251	Estadística y Probabilidades (Pág 113)	Estudios generales	2	2	3	O	BMA102 (1er Sem)
	FG106	Teatro (Pág 118)	Estudios generales	1	2	2	O	
	BEI201	Inglés III (Pág 122)	Estudios generales		4	2	O	BEI102 (2do Sem)
Electivo	EC101	Economía General (Pág 126)	Estudios generales	2	2	3	E	
	FG120	Realidad Nacional (Pág 129)	Estudios generales	2	2	3	E	
						22		

	Cuarto Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS210	Algoritmos y Estructuras de Datos (Pág 132)	Específicos	2	4	4	O	CS113 (3er Sem)
	CS211	Teoría de la Computación (Pág 137)	Específicos	2	4	4	O	CS1D1 (2do Sem)
	CS221	Arquitectura de Computadores (Pág 142)	Específicos	2	4	4	O	CS1D1 (2do Sem)
	CS271	Bases de Datos I (Pág 147)	Específicos	2	4	4	O	CS112 (2do Sem), CS1D1 (2do Sem)
	CS401	Metodología de la Investigación (Pág 152)	Especialización	1	2	2	O	CS100 (2do Sem)
	AI261	Probabilidad y Estadistica para IA (Pág 156)	Específicos	2	4	4	O	BMA102 (1er Sem), BMA101 (1er Sem), STA251 (3er Sem)
	FG350	Liderazgo y Desempeño (Pág 159)	Estudios generales	2		2	O	FG106 (3er Sem)
Electivo	BEI202	Inglés IV (Pág 162)	Estudios generales		4	2	E	BEI201 (3er Sem)
						26		

	Quinto Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS212	Análisis y Diseño de Algoritmos (Pág 166)	Específicos	2	4	4	O	CS210 (4to Sem), CS211 (4to Sem)
	CS261	Inteligencia Artificial (Pág 171)	Específicos	2	4	4	O	STA251 (3er Sem)
	CS291	Ingeniería de Software I (Pág 178)	Específicos	2	4	4	O	CS113 (3er Sem), CS271 (4to Sem)
	CS2S1	Sistemas Operativos (Pág 184)	Específicos	2	4	4	O	CS221 (4to Sem)
	AI162	Política y gobernanza de la IA (Pág 189)	Específicos	2	4	4	O	
	MA202	Métodos Numéricos (Pág 191)	Estudios generales	2	2	3	O	BMA103 (2do Sem)
Electivo	BEI203	Inglés V (Pág 196)	Estudios generales		4	2	E	BEI202 (4to Sem)
						25		

Sexto Semestre							
Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
CS311	Algoritmos para Problemas Complejos (Pág 200)	Específicos	2	2	3	O	CS212 (5to Sem)
AI262	Procesamiento de Lenguaje Natural (Pág 203)	Específicos	2	4	4	O	STA251 (3er Sem)
AI263	Introducción a Machine Learning (Pág 206)	Específicos	2	4	4	O	STA251 (3er Sem), AI161 (3er Sem), CS210 (4to Sem)
AI267	Interaccion Humano-IA (Pág 209)	Específicos	2	4	4	O	CS291 (5to Sem)
MA307	Matemática aplicada a la computación (Pág 214)	Específicos	2	4	4	O	CS212 (5to Sem)
FG211	Ética Profesional (Pág 217)	Específicos	2	2	3	O	
					22		

Séptimo Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
CS231	Redes y Comunicación (Pág 220)	Específicos	1	4	3	O	CS2S1 (5to Sem)
CS251	Computación Gráfica (Pág 225)	Especialización	2	4	4	O	MA202 (5to Sem)
AI264	Aprendizaje Profundo (Pág 228)	Específicos	2	4	4	O	AI263 (6to Sem), CS261 (5to Sem)
AI265	Aprendizaje por Refuerzo (Pág 231)	Específicos	2	4	4	O	AI263 (6to Sem)
AI266	IA para Ingenieria de Software (Pág 234)	Específicos	2	4	4	O	AI262 (6to Sem), AI263 (6to Sem)
AI400	Metodología de la Investigación (Pág 236)	Especialización	1	2	2	O	CS212 (5to Sem), BEI201 (3er Sem)
					21		

Octavo Semestre

Código	Curso	Área	HT	HP	CR	T	Requisitos
CS402	Proyecto de tesis 1 (Pág 239)	Especialización	2	4	4	O	CS401 (4to Sem)
AI362	Inteligencia Artificial para edge computing (Pág 243)	Específicos	2	4	4	O	AI264 (7mo Sem)
AI364	Inteligencia Artificial para Ciberseguridad I (Pág 245)	Específicos	2	4	4	O	AI264 (7mo Sem), AI265 (7mo Sem)
AI366	Sistemas de IA Distribuidos (Pág 247)	Específicos	2	4	4	O	AI264 (7mo Sem), AI265 (7mo Sem)
AI401	Proyecto de tesis 1 (Pág 249)	Especialización	1	5	4	O	CS400 (9no Sem), FG211 (6to Sem)
BI101	Biología (Pág 253)	Estudios generales	2	4	4	O	
EX301	Actividades Extracurriculares (Pág 256)	Estudios generales	1	2	2	O	
					26		

	Noveno Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS3I1	Seguridad en Computación (Pág 258)	Específicos	1	4	3	O	CS231 (7mo Sem)
	CS3P1	Computación Paralela y Distribuída (Pág 264)	Específicos	2	4	4	O	CS212 (5to Sem), CS231 (7mo Sem)
	CS400	Prácticas Pre-profesionales (Pág 274)	Especialización	1	2	2	O	110Cr
	CS403	Proyecto de tesis 2 (Pág 276)	Especialización	2	4	4	O	CS402 (8vo Sem)
	AI402	Proyecto de tesis 2 (Pág 280)	Especialización	1	5	4	O	CS401 (4to Sem)
	CB309	Bioinformática (Pág 284)	Especialización	2	4	4	O	BI101 (8vo Sem)
Electivo	AI268	Visión Computacional (Pág 288)	Especialización	2	4	4	E	AI264 (7mo Sem)
Electivo	AI363	Sistemas Autónomos (Pág 290)	Específicos	2	4	4	E	AI264 (7mo Sem)
Electivo	AI365	Modelos Generativos Avanzados en IA (Pág 292)	Específicos	2	4	4	E	AI264 (7mo Sem)
						25		

	Décimo Semestre							
	Código	**Curso**	**Área**	**HT**	**HP**	**CR**	**T**	**Requisitos**
	CS404	Taller de Investigación (Pág 294)	Especialización	2	4	4	O	CS403 (9no Sem)
	AI403	Taller de Investigación (Pág 298)	Especialización	1	5	4	O	CS402 (8vo Sem)
Electivo	AI369	Robótica (Pág 302)	Especialización	2	2	3	E	AI268 (9no Sem)
Electivo	CS281	Computación en la Sociedad (Pág 306)	Especialización	2		2	E	
Electivo	CS351	Tópicos en Computación Gráfica (Pág 312)	Especialización	2	2	3	E	CS251 (7mo Sem)
Electivo	CS353	Computación Cuántica (Pág 316)	Especialización	2	2	3	E	CS3I1 (9no Sem), MA307 (6to Sem)
Electivo	CS392	Tópicos en Ingeniería de Software (Pág 320)	Especialización	2	2	3	E	CS291 (5to Sem)
Electivo	CS393	Sistemas de Infomación (Pág 327)	Específicos	2	2	3	E	CS291 (5to Sem)
Electivo	CS3P3	Internet de las Cosas (Pág 330)	Específicos	1	4	3	E	CS3P1 (9no Sem)
						11		

Total de créditos de la carrera: 221 .

2.3. Tópicos distribuídos por curso

2.4. Resultados esperados distribuídos por curso

Las siquientes tablas nos muestras una visión global de los resultados que se esperan lograr en cada curso de la presente malla curricular. La lista completa de resultados esperados se encuentra en la Sección: 1.5.1.

		Primer Sem					Segundo Sem					
	Competencia/Curso	BIC101	BMA101	BMA102	BQU101	BEI101	CS100	CS112	CS1D1	BMA103	BFI101	BEI102
AG-C01)	Impacto social de soluciones computacionales.											
AG-C02)	Ética profesional en computación.						1					
AG-C03)	Trabajo individual y en equipo.					1						1
AG-C04)	Comunicación efectiva en computación.					2						2
AG-C05)	Gestión de proyectos en computación.											
AG-C06)	Aprendizaje continuo en computación.											
AG-C07)	Conocimientos de computación.		1	1	1				2	1	1	
AG-C08)	Análisis de problemas computacionales.	1					1					
AG-C09)	Diseño y desarrollo de soluciones.							2				
AG-C10)	Indagación en computación.											
AG-C11)	Uso de herramientas computacionales.											
AG-C12)	Gestión de tecnologías seguras de computación.	2					1	2	2			

Cuadro 2.1: Resultados esperados por curso 1er al 2do Semestre

Tercer Sem	CS113	CS2B1	AI161	STA251	FG106	BEI201	EC101	FG120
Competencia/Curso								
AG-C01) Impacto social de soluciones computacionales.			1				1	
AG-C02) Ética profesional en computación.					2			2
AG-C03) Trabajo individual y en equipo.						1		
AG-C04) Comunicación efectiva en computación.		2			2	2		2
AG-C05) Gestión de proyectos en computación.								
AG-C06) Aprendizaje continuo en computación.							1	
AG-C07) Conocimientos de computación.				1				
AG-C08) Análisis de problemas computacionales.								
AG-C09) Diseño y desarrollo de soluciones.	2	2						
AG-C10) Indagación en computación.								
AG-C11) Uso de herramientas computacionales.			2					
AG-C12) Gestión de tecnologías seguras de computación.	3							

Cuadro 2.2: Resultados esperados por curso 3er Semestre

		Cuarto Sem								Quinto Sem						
	Competencia/Curso	CS210	CS211	CS221	CS271	CS401	AI261	FG350	BEI202	CS212	CS261	CS291	CS2S1	AI162	MA202	BEI203
AG-C01)	Impacto social de soluciones computacionales.													2		
AG-C02)	Ética profesional en computación.					3		2						2		
AG-C03)	Trabajo individual y en equipo.			2					1			2	2			1
AG-C04)	Comunicación efectiva en computación.					3		2	2							2
AG-C05)	Gestión de proyectos en computación.					3										
AG-C06)	Aprendizaje continuo en computación.															
AG-C07)	Conocimientos de computación.						2									
AG-C08)	Análisis de problemas computacionales.	2	2			3				2	2				2	
AG-C09)	Diseño y desarrollo de soluciones.	2		2	2	3					2	2	2		2	
AG-C10)	Indagación en computación.															
AG-C11)	Uso de herramientas computacionales.														2	
AG-C12)	Gestión de tecnologías seguras de computación.		2		2					2						

Cuadro 2.3: Resultados esperados por curso 4to al 5to Semestre

		Sexto Sem						Séptimo Sem					
	Competencia/Curso	CS311	AI262	AI263	AI267	MA307	PG211	CS231	CS251	AI264	AI265	AI266	AI400
AG-C01)	Impacto social de soluciones computacionales.												
AG-C02)	Ética profesional en computación.						2						3
AG-C03)	Trabajo individual y en equipo.				2			2					
AG-C04)	Comunicación efectiva en computación.						2						3
AG-C05)	Gestión de proyectos en computación.												3
AG-C06)	Aprendizaje continuo en computación.												
AG-C07)	Conocimientos de computación.					3			2				
AG-C08)	Análisis de problemas computacionales.	2		2					2		2		3
AG-C09)	Diseño y desarrollo de soluciones.		2	2	2			2	3	2	2	2	3
AG-C10)	Indagación en computación.												
AG-C11)	Uso de herramientas computacionales.		2			2				3			
AG-C12)	Gestión de tecnologías seguras de computación.	2				3						2	

Cuadro 2.4: Resultados esperados por curso 6to al 7mo Semestre

Octavo Sem	CS402	AI362	AI364	AI366	AI401	BI101	EX301
Competencia/Curso							
AG-C01) Impacto social de soluciones computacionales.							1
AG-C02) Ética profesional en computación.	3	2	2		3		
AG-C03) Trabajo individual y en equipo.							
AG-C04) Comunicación efectiva en computación.	3				3		
AG-C05) Gestión de proyectos en computación.	3			2	3		
AG-C06) Aprendizaje continuo en computación.							1
AG-C07) Conocimientos de computación.						1	
AG-C08) Análisis de problemas computacionales.	3				3		
AG-C09) Diseño y desarrollo de soluciones.	3	3	3	2	3		
AG-C10) Indagación en computación.						1	
AG-C11) Uso de herramientas computacionales.							
AG-C12) Gestión de tecnologías seguras de computación.		3	3				

Cuadro 2.5: Resultados esperados por curso 8vo Semestre

Noveno Sem								
Competencia/Curso	CS311	CS3P1	CS400	CS403	AI402	CB309	AI268	AI363	AI365
AG-C01) Impacto social de soluciones computacionales.			2						
AG-C02) Ética profesional en computación.			3	3	3			2	
AG-C03) Trabajo individual y en equipo.			3						
AG-C04) Comunicación efectiva en computación.			3	3	3				
AG-C05) Gestión de proyectos en computación.			2	3	3				
AG-C06) Aprendizaje continuo en computación.			3						
AG-C07) Conocimientos de computación.							2		
AG-C08) Análisis de problemas computacionales.				3	3			3	3
AG-C09) Diseño y desarrollo de soluciones.	2	2	3	3	3				
AG-C10) Indagación en computación.						2			
AG-C11) Uso de herramientas computacionales.	2	2				2	2		3
AG-C12) Gestión de tecnologías seguras de computación.								3	

Cuadro 2.6: Resultados esperados por curso 9no Semestre

		Décimo Sem								
	Competencia/Curso	CS404	AI403	AI369	CS281	CS351	CS353	CS392	CS393	CS383
AG-C01)	Impacto social de soluciones computacionales.				2					
AG-C02)	Ética profesional en computación.	3	3		2					
AG-C03)	Trabajo individual y en equipo.							2	2	
AG-C04)	Comunicación efectiva en computación.	3	3							
AG-C05)	Gestión de proyectos en computación.	3	3							
AG-C06)	Aprendizaje continuo en computación.									
AG-C07)	Conocimientos de computación.									
AG-C08)	Análisis de problemas computacionales.	3	3							
AG-C09)	Diseño y desarrollo de soluciones.	3	3	2		2		2	2	2
AG-C10)	Indagación en computación.						2			
AG-C11)	Uso de herramientas computacionales.			3		2	2			2
AG-C12)	Gestión de tecnologías seguras de computación.									

Cuadro 2.7: Resultados esperados por curso 10mo Semestre

2.5. Cursos por resultado (*Outcome*)

2.5.1. Outcome: AG-C01) Impacto social de soluciones computacionales.

- AI161 IA Aplicada (3er Sem-Pág 109)
- EC101 Economía General (3er Sem-Pág 126)
- AI162 Política y gobernanza de la IA (5to Sem-Pág 189)
- EX301 Actividades Extracurriculares (8vo Sem-Pág 256)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS281 Computación en la Sociedad (10mo Sem-Pág 306)

2.5.2. Outcome: AG-C02) Ética profesional en computación.

- CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)
- FG106 Teatro (3er Sem-Pág 118)
- FG120 Realidad Nacional (3er Sem-Pág 129)
- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- FG350 Liderazgo y Desempeño (4to Sem-Pág 159)
- AI162 Política y gobernanza de la IA (5to Sem-Pág 189)
- FG211 Ética Profesional (6to Sem-Pág 217)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)
- AI362 Inteligencia Artificial para edge computing (8vo Sem-Pág 243)
- AI364 Inteligencia Artificial para Ciberseguridad I (8vo Sem-Pág 245)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- AI363 Sistemas Autónomos (9no Sem-Pág 290)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)
- CS281 Computación en la Sociedad (10mo Sem-Pág 306)

2.5.3. Outcome: AG-C03) Trabajo individual y en equipo.

- BEI101 Inglés I (1er Sem-Pág 65)
- BEI102 Inglés II (2do Sem-Pág 95)
- BEI201 Inglés III (3er Sem-Pág 122)
- CS221 Arquitectura de Computadores (4to Sem-Pág 142)
- BEI202 Inglés IV (4to Sem-Pág 162)
- CS291 Ingeniería de Software I (5to Sem-Pág 178)
- CS2S1 Sistemas Operativos (5to Sem-Pág 184)
- BEI203 Inglés V (5to Sem-Pág 196)
- AI267 Interaccion Humano-IA (6to Sem-Pág 209)
- CS231 Redes y Comunicación (7mo Sem-Pág 220)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS392 Tópicos en Ingeniería de Software (10mo Sem-Pág 320)
- CS393 Sistemas de Infomación (10mo Sem-Pág 327)

2.5.4. Outcome: AG-C04) Comunicación efectiva en computación.

- BEI101 Inglés I (1er Sem-Pág 65)
- BEI102 Inglés II (2do Sem-Pág 95)
- CS2B1 Desarrollo Basado en Plataformas (3er Sem-Pág 105)
- FG106 Teatro (3er Sem-Pág 118)
- BEI201 Inglés III (3er Sem-Pág 122)
- FG120 Realidad Nacional (3er Sem-Pág 129)
- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- FG350 Liderazgo y Desempeño (4to Sem-Pág 159)
- BEI202 Inglés IV (4to Sem-Pág 162)
- BEI203 Inglés V (5to Sem-Pág 196)
- FG211 Ética Profesional (6to Sem-Pág 217)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)

2.5.5. Outcome: AG-C05) Gestión de proyectos en computación.

- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)
- AI366 Sistemas de IA Distribuidos (8vo Sem-Pág 247)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)

2.5.6. Outcome: AG-C06) Aprendizaje continuo en computación.

- EC101 Economía General (3er Sem-Pág 126)
- EX301 Actividades Extracurriculares (8vo Sem-Pág 256)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)

2.5.7. Outcome: AG-C07) Conocimientos de computación.

- BMA101 Álgebra Lineal (1er Sem-Pág 54)
- BMA102 Cálculo Diferencial (1er Sem-Pág 57)
- BQU101 Química I (1er Sem-Pág 60)
- CS1D1 Estructuras Discretas (2do Sem-Pág 85)
- BMA103 Cálculo Integral (2do Sem-Pág 88)
- BFI101 Física I (2do Sem-Pág 90)
- STA251 Estadística y Probabilidades (3er Sem-Pág 113)
- AI261 Probabilidad y Estadistica para IA (4to Sem-Pág 156)

- MA307 Matemática aplicada a la computación (6to Sem-Pág 214)
- CS251 Computación Gráfica (7mo Sem-Pág 225)
- BI101 Biología (8vo Sem-Pág 253)

2.5.8. Outcome: AG-C08) Análisis de problemas computacionales.

- BIC101 Introducción a la Computación (1er Sem-Pág 49)
- CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)
- CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 132)
- CS211 Teoría de la Computación (4to Sem-Pág 137)
- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)
- CS261 Inteligencia Artificial (5to Sem-Pág 171)
- MA202 Métodos Numéricos (5to Sem-Pág 191)
- CS311 Algoritmos para Problemas Complejos (6to Sem-Pág 200)
- AI263 Introducción a Machine Learning (6to Sem-Pág 206)
- CS251 Computación Gráfica (7mo Sem-Pág 225)
- AI265 Aprendizaje por Refuerzo (7mo Sem-Pág 231)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- AI268 Visión Computacional (9no Sem-Pág 288)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)

2.5.9. Outcome: AG-C09) Diseño y desarrollo de soluciones.

- CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)
- CS113 Programación Orientada a Objetos II (3er Sem-Pág 99)
- CS2B1 Desarrollo Basado en Plataformas (3er Sem-Pág 105)
- CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 132)
- CS221 Arquitectura de Computadores (4to Sem-Pág 142)
- CS271 Bases de Datos I (4to Sem-Pág 147)
- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- CS261 Inteligencia Artificial (5to Sem-Pág 171)
- CS291 Ingeniería de Software I (5to Sem-Pág 178)
- CS2S1 Sistemas Operativos (5to Sem-Pág 184)
- MA202 Métodos Numéricos (5to Sem-Pág 191)
- AI262 Procesamiento de Lenguaje Natural (6to Sem-Pág 203)
- AI263 Introducción a Machine Learning (6to Sem-Pág 206)
- AI267 Interaccion Humano-IA (6to Sem-Pág 209)
- CS231 Redes y Comunicación (7mo Sem-Pág 220)
- CS251 Computación Gráfica (7mo Sem-Pág 225)
- AI264 Aprendizaje Profundo (7mo Sem-Pág 228)
- AI265 Aprendizaje por Refuerzo (7mo Sem-Pág 231)
- AI266 IA para Ingenieria de Software (7mo Sem-Pág 234)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)

- AI362 Inteligencia Artificial para edge computing (8vo Sem-Pág 243)
- AI364 Inteligencia Artificial para Ciberseguridad I (8vo Sem-Pág 245)
- AI366 Sistemas de IA Distribuidos (8vo Sem-Pág 247)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS3I1 Seguridad en Computación (9no Sem-Pág 258)
- CS3P1 Computación Paralela y Distribuída (9no Sem-Pág 264)
- CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- AI363 Sistemas Autónomos (9no Sem-Pág 290)
- AI365 Modelos Generativos Avanzados en IA (9no Sem-Pág 292)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)
- AI369 Robótica (10mo Sem-Pág 302)
- CS351 Tópicos en Computación Gráfica (10mo Sem-Pág 312)
- CS392 Tópicos en Ingeniería de Software (10mo Sem-Pág 320)
- CS393 Sistemas de Infomación (10mo Sem-Pág 327)
- CS3P3 Internet de las Cosas (10mo Sem-Pág 330)

2.5.10. Outcome: AG-C10) Indagación en computación.

- BI101 Biología (8vo Sem-Pág 253)
- CB309 Bioinformática (9no Sem-Pág 284)
- CS353 Computación Cuántica (10mo Sem-Pág 316)

2.5.11. Outcome: AG-C11) Uso de herramientas computacionales.

- AI161 IA Aplicada (3er Sem-Pág 109)
- MA202 Métodos Numéricos (5to Sem-Pág 191)
- AI262 Procesamiento de Lenguaje Natural (6to Sem-Pág 203)
- MA307 Matemática aplicada a la computación (6to Sem-Pág 214)
- AI264 Aprendizaje Profundo (7mo Sem-Pág 228)
- CS3I1 Seguridad en Computación (9no Sem-Pág 258)
- CS3P1 Computación Paralela y Distribuída (9no Sem-Pág 264)
- CB309 Bioinformática (9no Sem-Pág 284)
- AI268 Visión Computacional (9no Sem-Pág 288)
- AI365 Modelos Generativos Avanzados en IA (9no Sem-Pág 292)
- AI369 Robótica (10mo Sem-Pág 302)
- CS351 Tópicos en Computación Gráfica (10mo Sem-Pág 312)
- CS353 Computación Cuántica (10mo Sem-Pág 316)
- CS3P3 Internet de las Cosas (10mo Sem-Pág 330)

2.5.12. Outcome: AG-C12) Gestión de tecnologías seguras de computación.

- BIC101 Introducción a la Computación (1er Sem-Pág 49)
- CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)
- CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)
- CS1D1 Estructuras Discretas (2do Sem-Pág 85)
- CS113 Programación Orientada a Objetos II (3er Sem-Pág 99)
- CS211 Teoría de la Computación (4to Sem-Pág 137)

- CS271 Bases de Datos I (4to Sem-Pág 147)
- CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)
- CS311 Algoritmos para Problemas Complejos (6to Sem-Pág 200)
- MA307 Matemática aplicada a la computación (6to Sem-Pág 214)
- AI266 IA para Ingenieria de Software (7mo Sem-Pág 234)
- AI362 Inteligencia Artificial para edge computing (8vo Sem-Pág 243)
- AI364 Inteligencia Artificial para Ciberseguridad I (8vo Sem-Pág 245)
- AI363 Sistemas Autónomos (9no Sem-Pág 290)

2.6. 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).

	Estudios generales	Específicos	Especialización	
1er Sem	16	4		20
2do Sem	12	11		23
3er Sem	10	12		22
4to Sem	4	20	2	26
5to Sem	5	20		25
6to Sem		22		22
7mo Sem		15	6	21
8vo Sem	6	12	8	26
9no Sem		11	14	25
10mo Sem		3	8	11
Total	53	130	38	221
	23.9 %	58.8 %	17.1 %	

Cuadro 2.8: Distribución de créditos por área por semestre

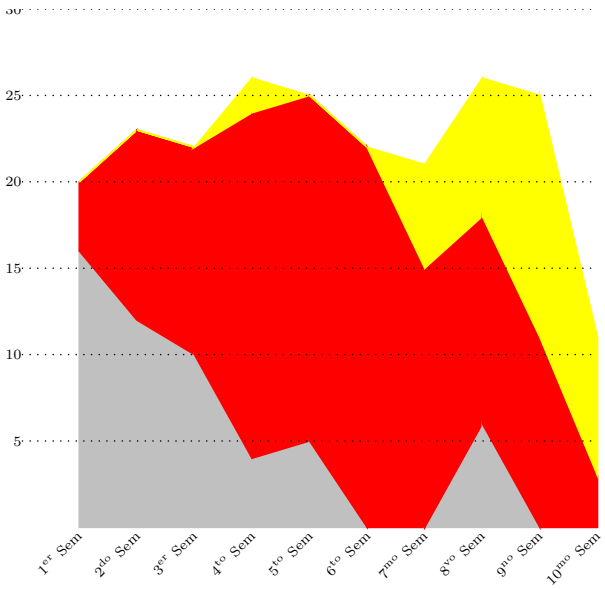


Figura 2.2: Créditos por área por semestre

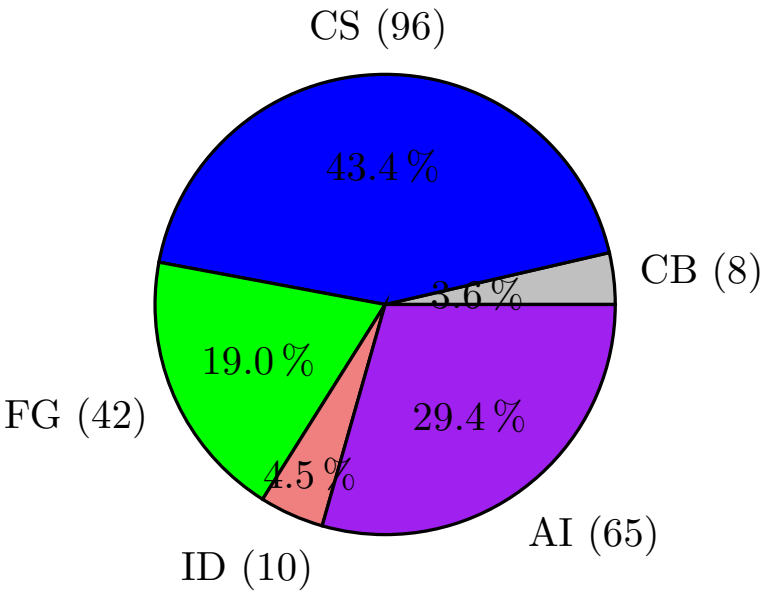


Figura 2.3: Distribución de cursos por áreas considerando creditaje.

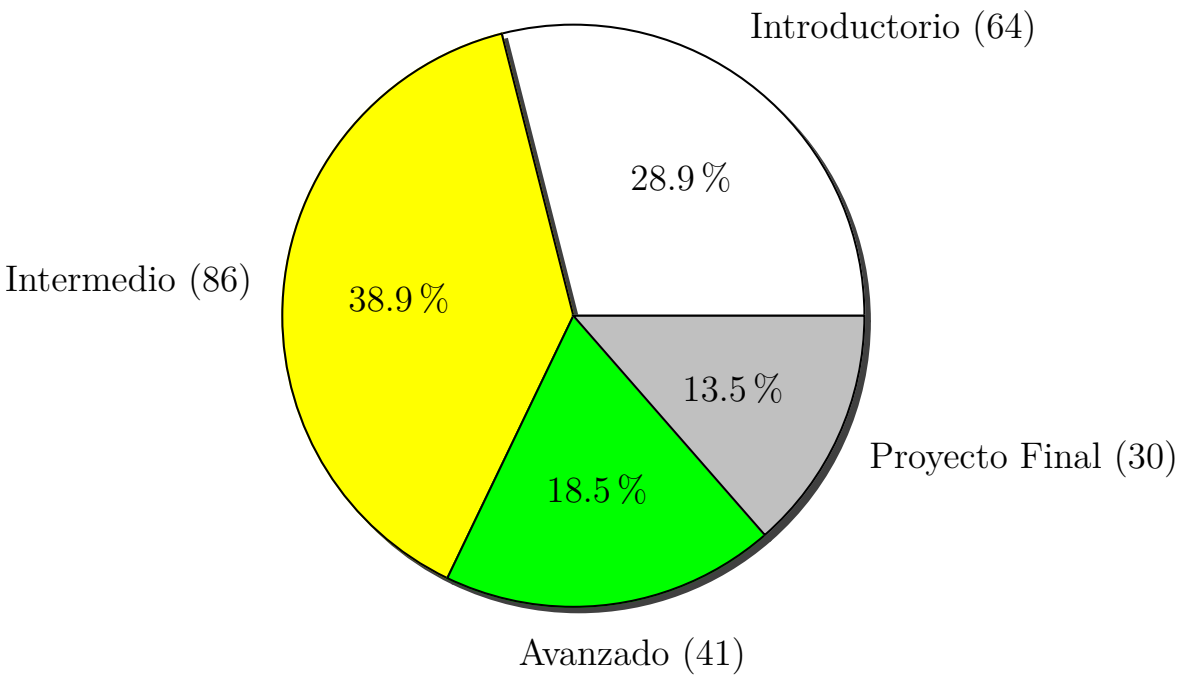


Figura 2.4: Distribución de créditos por niveles de cursos.

2.7. Departamentos

2.7.1. DAISI

La lista de cursos de este departamento es:

- CS100 Introducción a la Ciencia de la Computación (2do Sem)
- CS1D1 Estructuras Discretas (2do Sem)
- CS2B1 Desarrollo Basado en Plataformas (3er Sem)
- CS210 Algoritmos y Estructuras de Datos (4to Sem)

- CS211 Teoría de la Computación (4to Sem)
- CS212 Análisis y Diseño de Algoritmos (5to Sem)
- CS261 Inteligencia Artificial (5to Sem)
- CS291 Ingeniería de Software I (5to Sem)
- AI267 Interaccion Humano-IA (6to Sem)
- CS311 Algoritmos para Problemas Complejos (6to Sem)
- FG211 Ética Profesional (6to Sem)
- CS231 Redes y Comunicación (7mo Sem)
- CB309 Bioinformática (9no Sem)
- CS3I1 Seguridad en Computación (9no Sem)
- CS3P1 Computación Paralela y Distribuída (9no Sem)
- AI369 Robótica (10mo Sem)
- CS281 Computación en la Sociedad (10mo Sem)
- CS351 Tópicos en Computación Gráfica (10mo Sem)
- CS392 Tópicos en Ingeniería de Software (10mo Sem)
- CS393 Sistemas de Infomación (10mo Sem)
- CS3P3 Internet de las Cosas (10mo Sem)

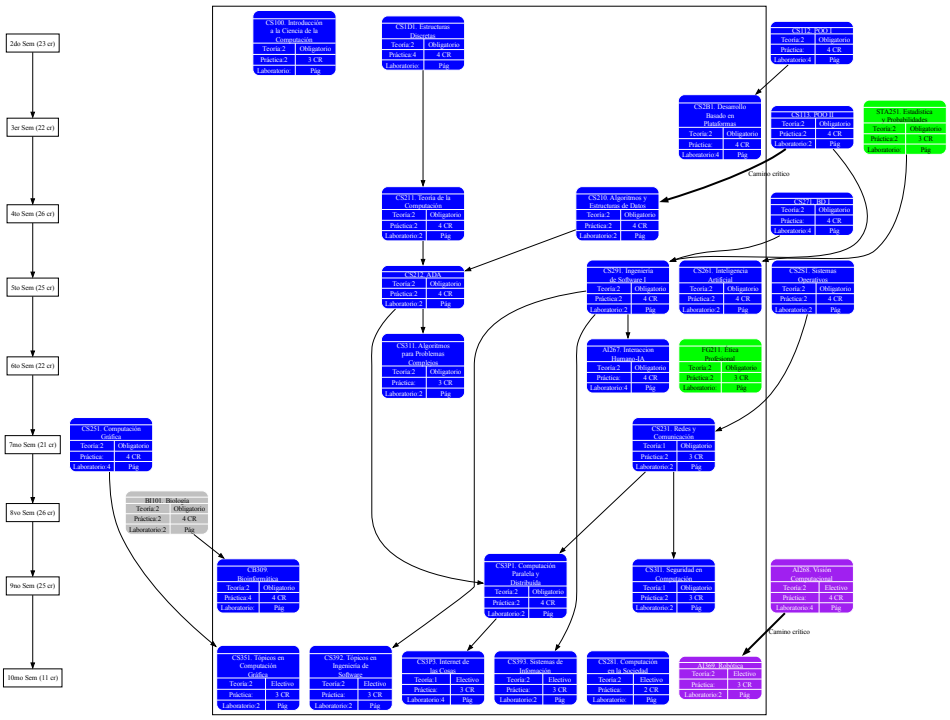


Figura 2.5: DAISI

2.7.2. DLL

La lista de cursos de este departamento es:

- BEI101 Inglés I (1er Sem)
- BEI102 Inglés II (2do Sem)
- BEI201 Inglés III (3er Sem)

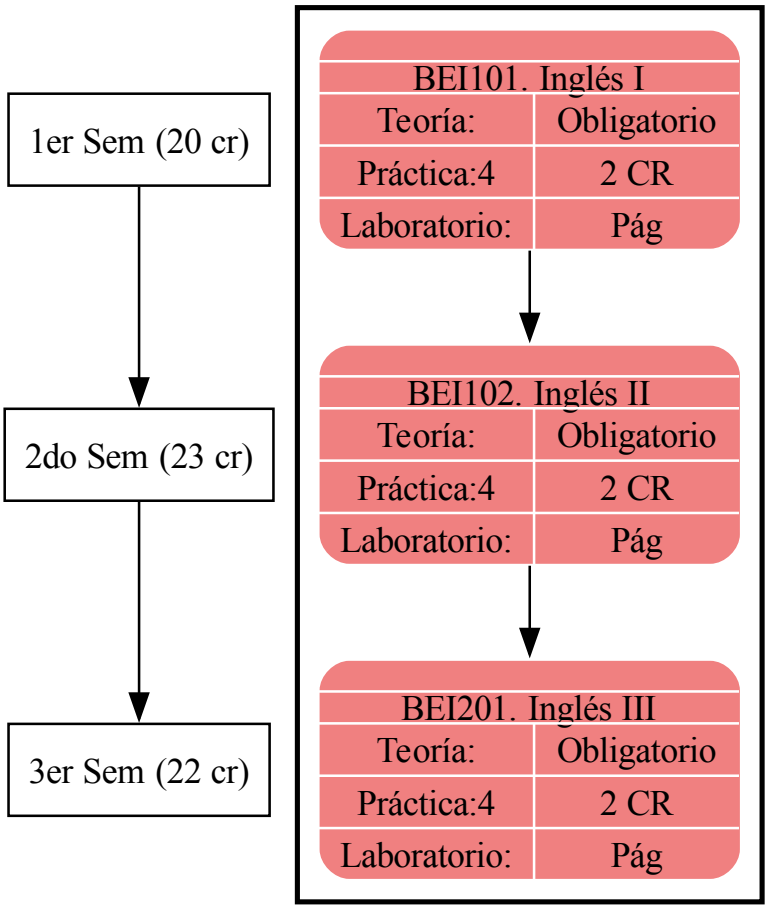


Figura 2.6: DLL

2.8. Compatibilidad de la carrera con relación a estandares internacionales

En esta sección presentamos la distribución de cursos por áreas de concentración en contraste con las propuestas internacionales de las carreras de la *Computing Curricula* de IEEE-CS/ACM.

Es necesario notar que en algunos casos las materias podrían aparecen en más de un eje pues tienen contenido de más de una área. Por ejemplo, la materia de sistemas operativos contiene unidades de aplicación que pueden ser clasificadas en Tecnología de Información pero al mismo tiempo contiene fundamentos de como está estructurado un Sistema Operativo que es del eje de Inteligencia Artificial. En estos casos el creditaje ha sido divido entre los ejes correspondientes.

1. Hardware y Arquitectura (9.50 Créditos)
   - CS221 Arquitectura de Computadores (4to Sem-Pág 142)
   - AI369 Robótica (10mo Sem-Pág 302)
2. Ciencia de la Computación (253.67 Créditos)
   - BIC101 Introducción a la Computación (1er Sem-Pág 49)
   - CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)

- CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)
- CS113 Programación Orientada a Objetos II (3er Sem-Pág 99)
- AI161 IA Aplicada (3er Sem-Pág 109)
- CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 132)
- CS211 Teoría de la Computación (4to Sem-Pág 137)
- CS401 Metodología de la Investigación (4to Sem-Pág 152)
- AI261 Probabilidad y Estadistica para IA (4to Sem-Pág 156)
- CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)
- CS261 Inteligencia Artificial (5to Sem-Pág 171)
- CS2S1 Sistemas Operativos (5to Sem-Pág 184)
- AI162 Política y gobernanza de la IA (5to Sem-Pág 189)
- AI262 Procesamiento de Lenguaje Natural (6to Sem-Pág 203)
- AI263 Introducción a Machine Learning (6to Sem-Pág 206)
- AI267 Interaccion Humano-IA (6to Sem-Pág 209)
- CS231 Redes y Comunicación (7mo Sem-Pág 220)
- CS251 Computación Gráfica (7mo Sem-Pág 225)
- AI264 Aprendizaje Profundo (7mo Sem-Pág 228)
- AI265 Aprendizaje por Refuerzo (7mo Sem-Pág 231)
- AI266 IA para Ingenieria de Software (7mo Sem-Pág 234)
- AI400 Metodología de la Investigación (7mo Sem-Pág 236)
- CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)
- AI362 Inteligencia Artificial para edge computing (8vo Sem-Pág 243)
- AI364 Inteligencia Artificial para Ciberseguridad I (8vo Sem-Pág 245)
- AI366 Sistemas de IA Distribuidos (8vo Sem-Pág 247)
- AI401 Proyecto de tesis 1 (8vo Sem-Pág 249)
- CS3P1 Computación Paralela y Distribuída (9no Sem-Pág 264)
- CS403 Proyecto de tesis 2 (9no Sem-Pág 276)
- AI402 Proyecto de tesis 2 (9no Sem-Pág 280)
- CB309 Bioinformática (9no Sem-Pág 284)
- AI268 Visión Computacional (9no Sem-Pág 288)
- AI363 Sistemas Autónomos (9no Sem-Pág 290)
- AI365 Modelos Generativos Avanzados en IA (9no Sem-Pág 292)
- CS404 Taller de Investigación (10mo Sem-Pág 294)
- AI403 Taller de Investigación (10mo Sem-Pág 298)
- AI369 Robótica (10mo Sem-Pág 302)
- CS281 Computación en la Sociedad (10mo Sem-Pág 306)
- CS351 Tópicos en Computación Gráfica (10mo Sem-Pág 312)
- CS353 Computación Cuántica (10mo Sem-Pág 316)
- CS3P3 Internet de las Cosas (10mo Sem-Pág 330)

3. Inteligencia Artificial
   - Ninguno
4. Ingeniería de Software (24.67 Créditos)
   - CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)
   - CS113 Programación Orientada a Objetos II (3er Sem-Pág 99)
   - CS291 Ingeniería de Software I (5to Sem-Pág 178)
   - AI267 Interaccion Humano-IA (6to Sem-Pág 209)
   - CS392 Tópicos en Ingeniería de Software (10mo Sem-Pág 320)
   - CS393 Sistemas de Infomación (10mo Sem-Pág 327)
5. Sistemas de Información

- Ninguno

6. Tecnología de Información (24.17 Créditos)
   - CS2B1 Desarrollo Basado en Plataformas (3er Sem-Pág 105)
   - CS271 Bases de Datos I (4to Sem-Pág 147)
   - CS2S1 Sistemas Operativos (5to Sem-Pág 184)
   - AI267 Interaccion Humano-IA (6to Sem-Pág 209)
   - CS231 Redes y Comunicación (7mo Sem-Pág 220)
   - CS281 Computación en la Sociedad (10mo Sem-Pág 306)
   - CS3P3 Internet de las Cosas (10mo Sem-Pág 330)
7. Ciencia de Datos (4.00 Créditos)
   - CS271 Bases de Datos I (4to Sem-Pág 147)
8. Ciberseguridad (6.00 Créditos)
   - CS3I1 Seguridad en Computación (9no Sem-Pág 258)
9. Matemática para Computación (22.00 Créditos)
   - CS1D1 Estructuras Discretas (2do Sem-Pág 85)
   - CS311 Algoritmos para Problemas Complejos (6to Sem-Pág 200)
   - MA307 Matemática aplicada a la computación (6to Sem-Pág 214)
10. Ciencias Básicas (68.00 Créditos)
    - BMA101 Álgebra Lineal (1er Sem-Pág 54)
    - BMA102 Cálculo Diferencial (1er Sem-Pág 57)
    - BQU101 Química I (1er Sem-Pág 60)
    - BMA103 Cálculo Integral (2do Sem-Pág 88)
    - BFI101 Física I (2do Sem-Pág 90)
    - STA251 Estadística y Probabilidades (3er Sem-Pág 113)
    - MA202 Métodos Numéricos (5to Sem-Pág 191)
    - BI101 Biología (8vo Sem-Pág 253)
11. Aspectos Organizacionales (4.00 Créditos)
    - FG350 Liderazgo y Desempeño (4to Sem-Pág 159)
12. Contenido Empresarial
    - Ninguno
13. Humanidades
    - Ninguno
14. Formación General (40.00 Créditos)
    - BEI101 Inglés I (1er Sem-Pág 65)
    - BEI102 Inglés II (2do Sem-Pág 95)
    - FG106 Teatro (3er Sem-Pág 118)
    - BEI201 Inglés III (3er Sem-Pág 122)
    - EC101 Economía General (3er Sem-Pág 126)
    - FG120 Realidad Nacional (3er Sem-Pág 129)
    - BEI202 Inglés IV (4to Sem-Pág 162)
    - BEI203 Inglés V (5to Sem-Pág 196)
    - FG211 Ética Profesional (6to Sem-Pág 217)
    - EX301 Actividades Extracurriculares (8vo Sem-Pág 256)
    - CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)

Considerando esta distribución, las figuras 2.7 a 2.19 nos permiten tener una vista gráfica de este currículo en comparación con las propuestas de programas presentadas por IEEE-CS/ACM en la *Computing Curricula*.

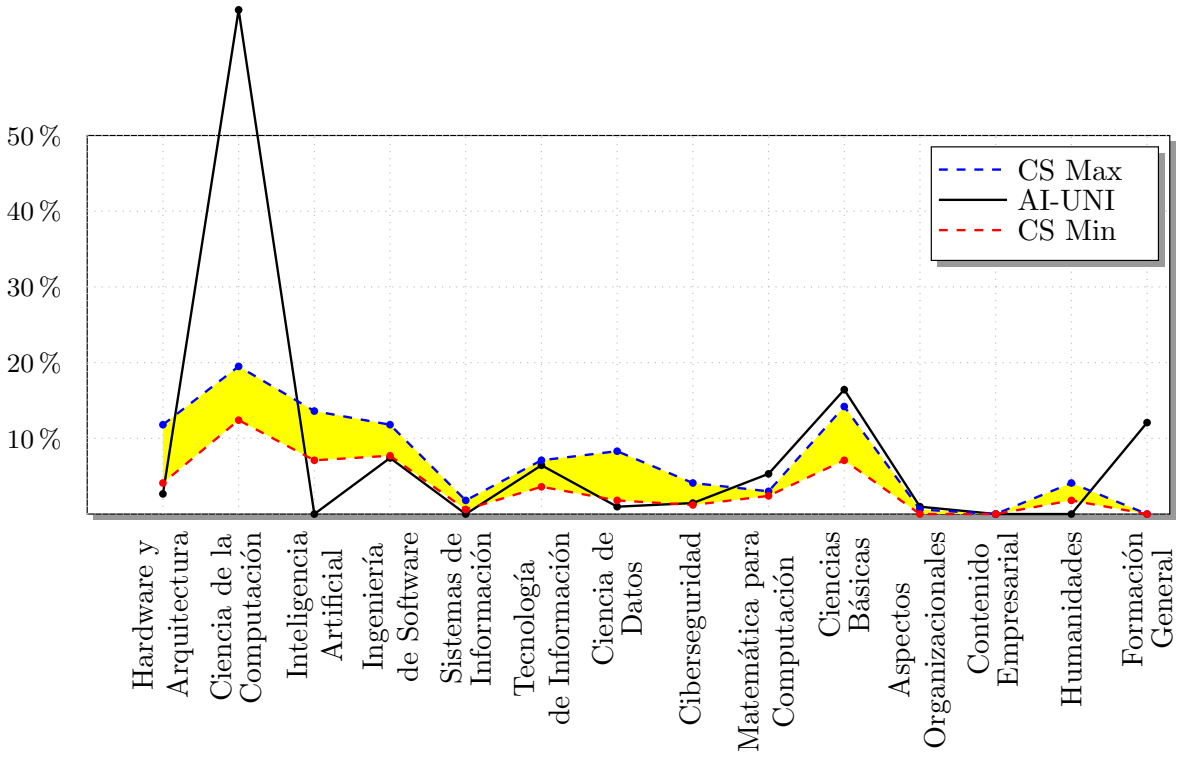


Figura 2.7: Comparación de AI-UNI con CS de ACM/IEEE-CS (Gráfico de curvas).

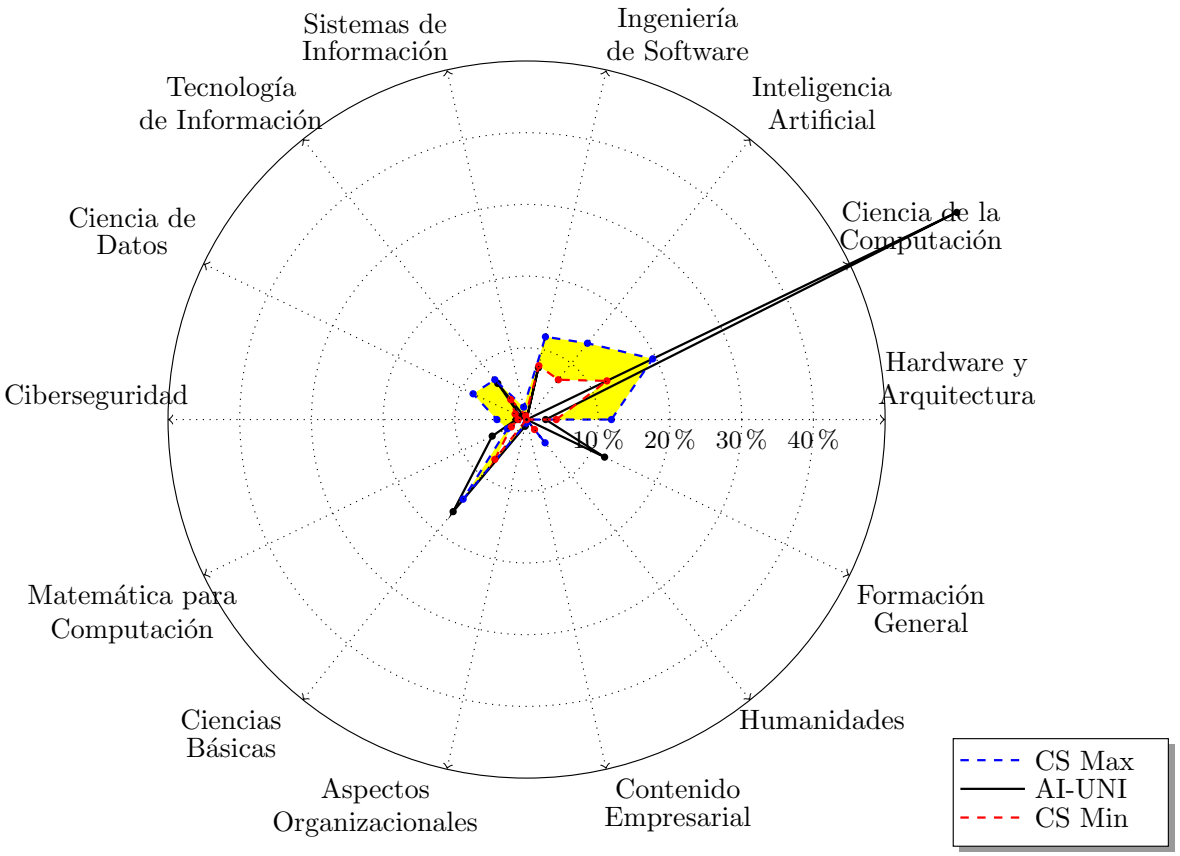


Figura 2.8: Comparación de AI-UNI con CS de ACM/IEEE-CS (Gráfico de araña).

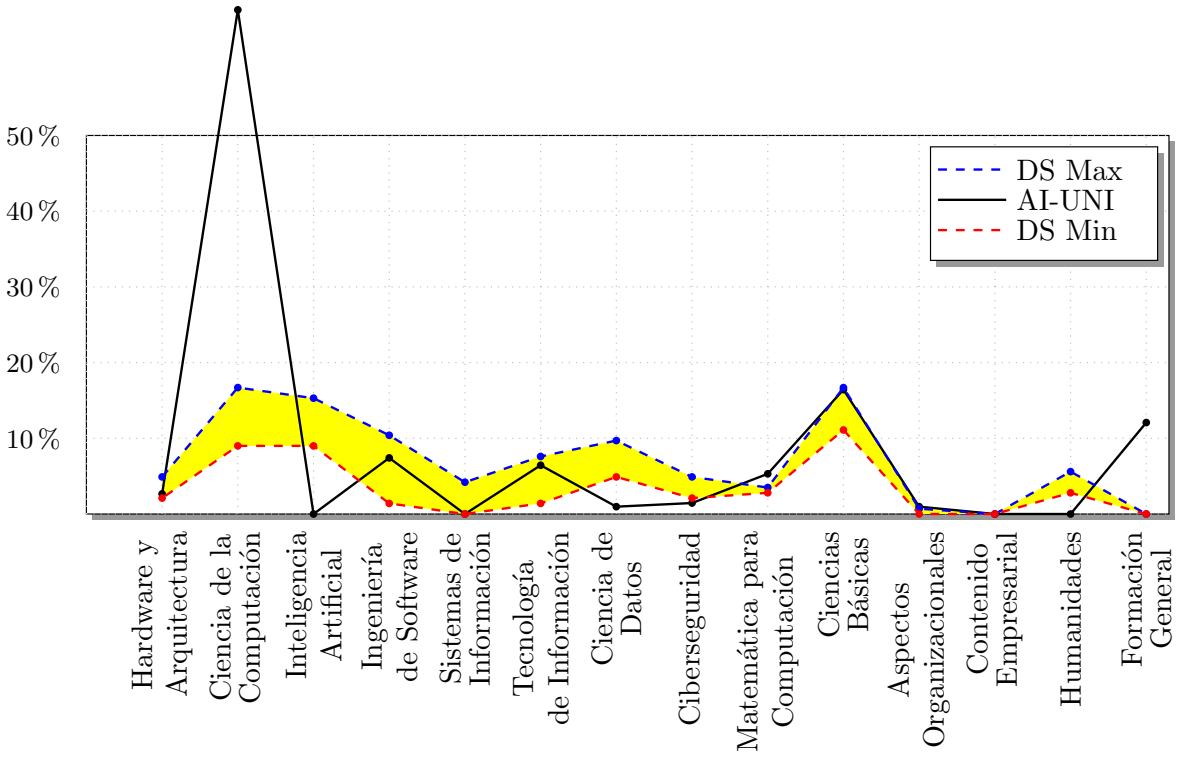


Figura 2.9: Comparación de AI-UNI con DS de ACM/IEEE-CS (Gráfico de curvas).

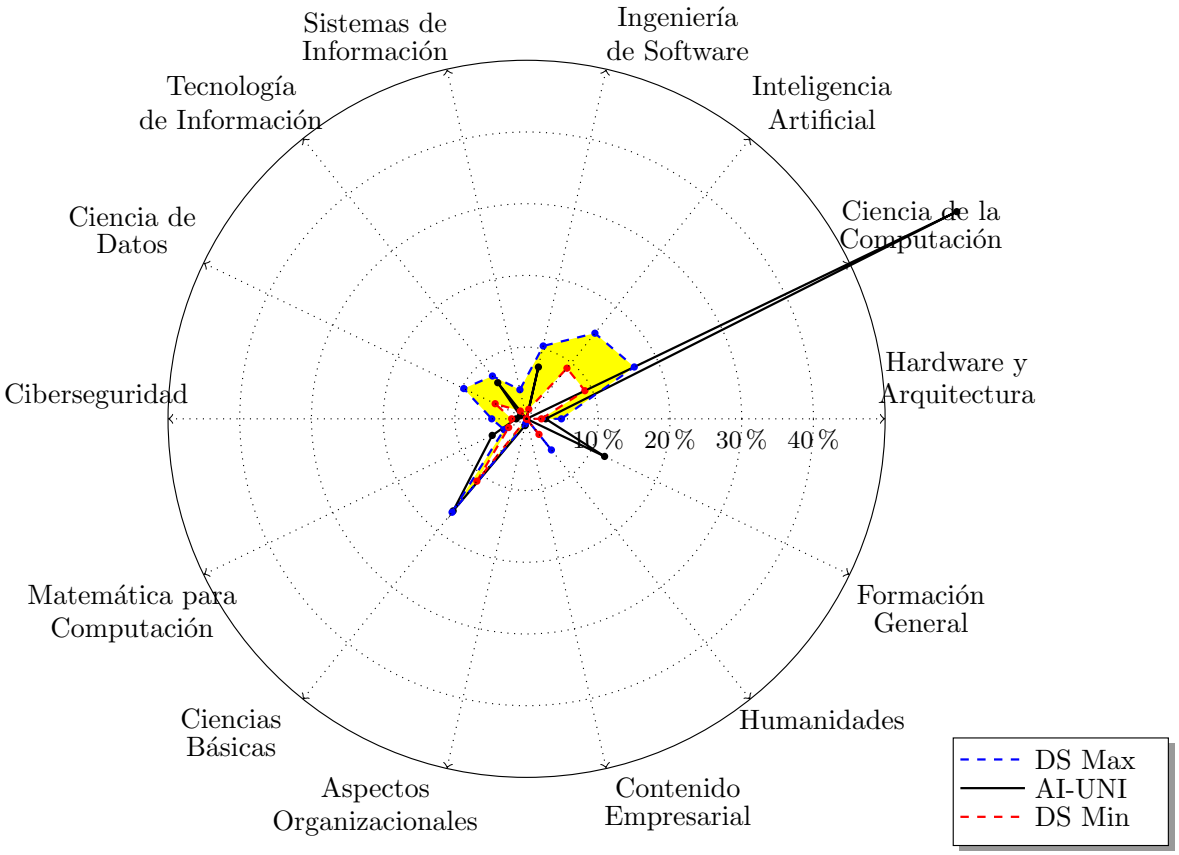


Figura 2.10: Comparación de AI-UNI con DS de ACM/IEEE-CS (Gráfico de araña).

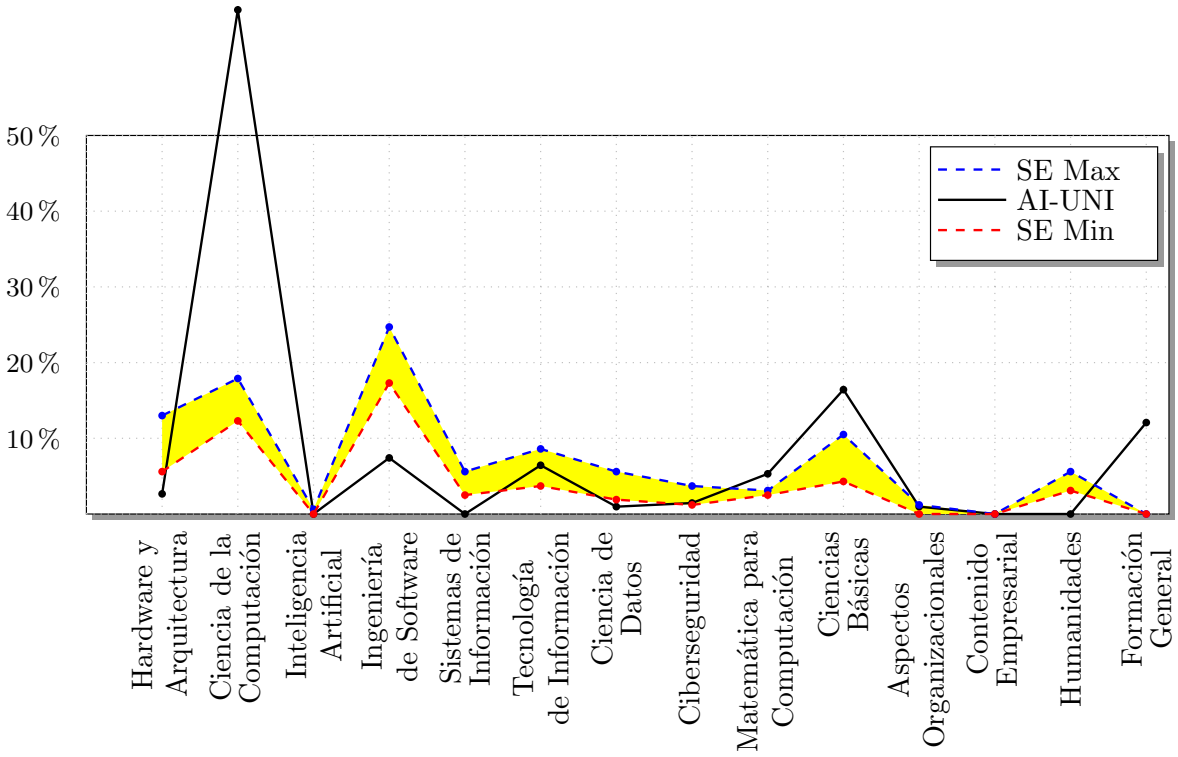


Figura 2.11: Comparación de AI-UNI con SE de ACM/IEEE-CS (Gráfico de curvas).

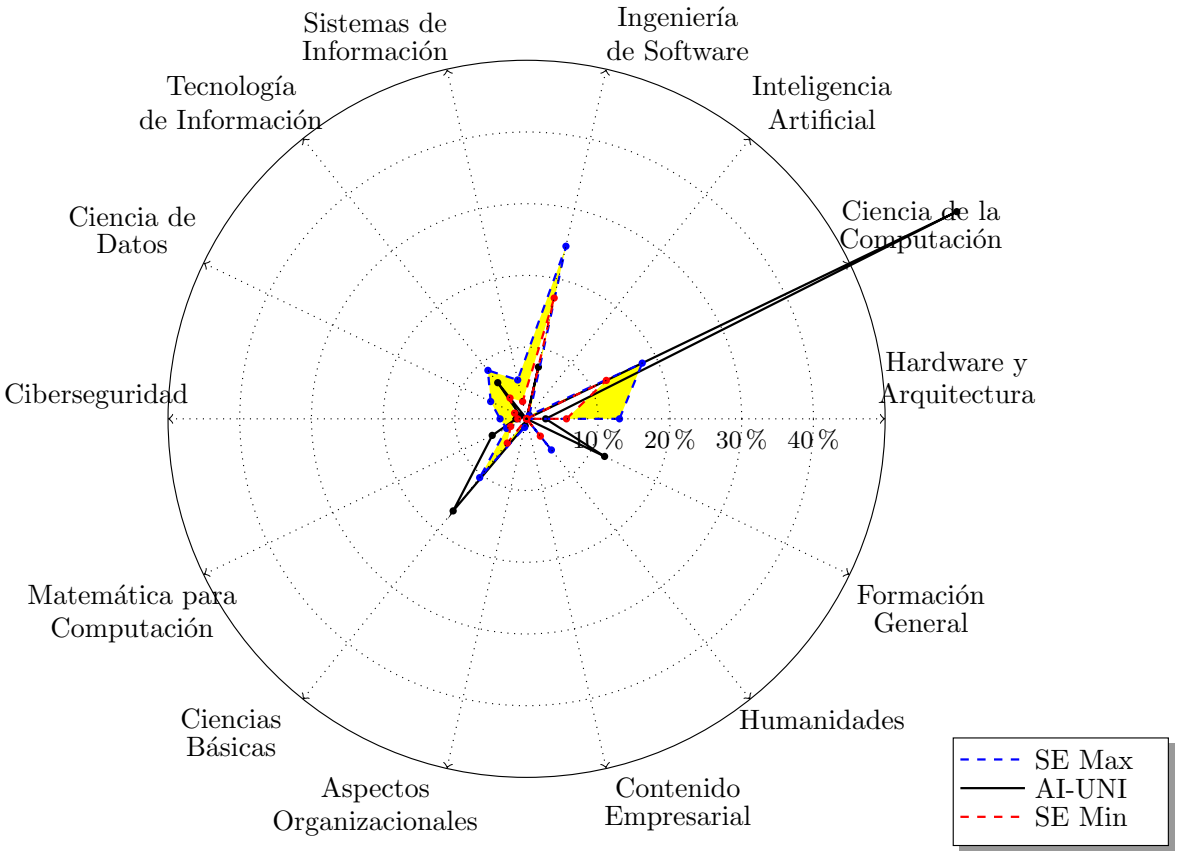


Figura 2.12: Comparación de AI-UNI con SE de ACM/IEEE-CS (Gráfico de araña).

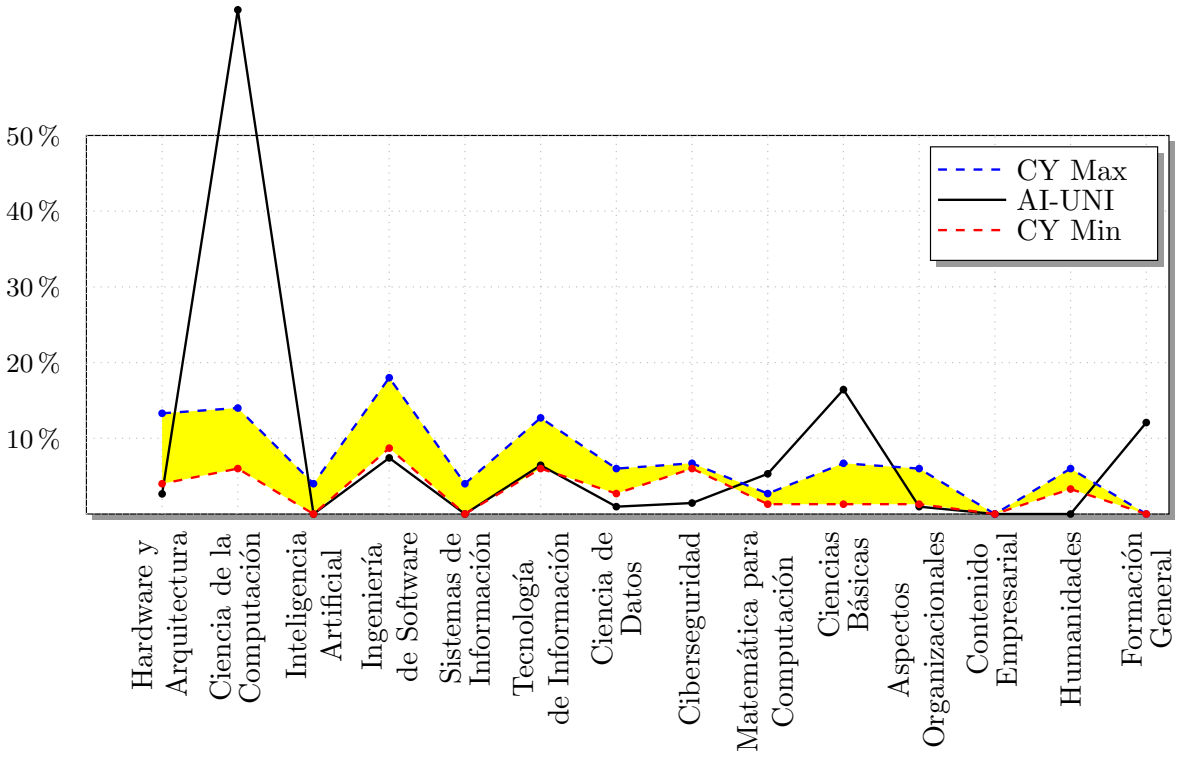


Figura 2.13: Comparación de AI-UNI con CY de ACM/IEEE-CS (Gráfico de curvas).

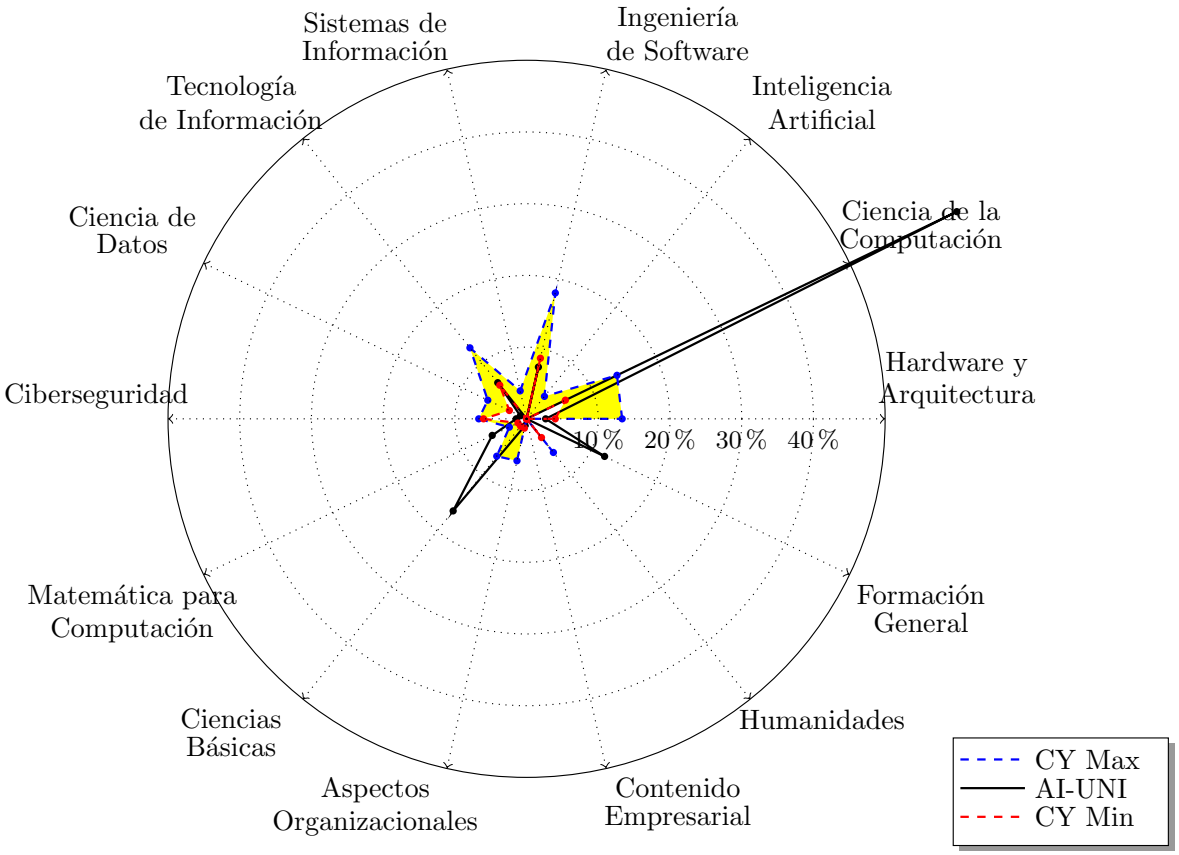


Figura 2.14: Comparación de AI-UNI con CY de ACM/IEEE-CS (Gráfico de araña).

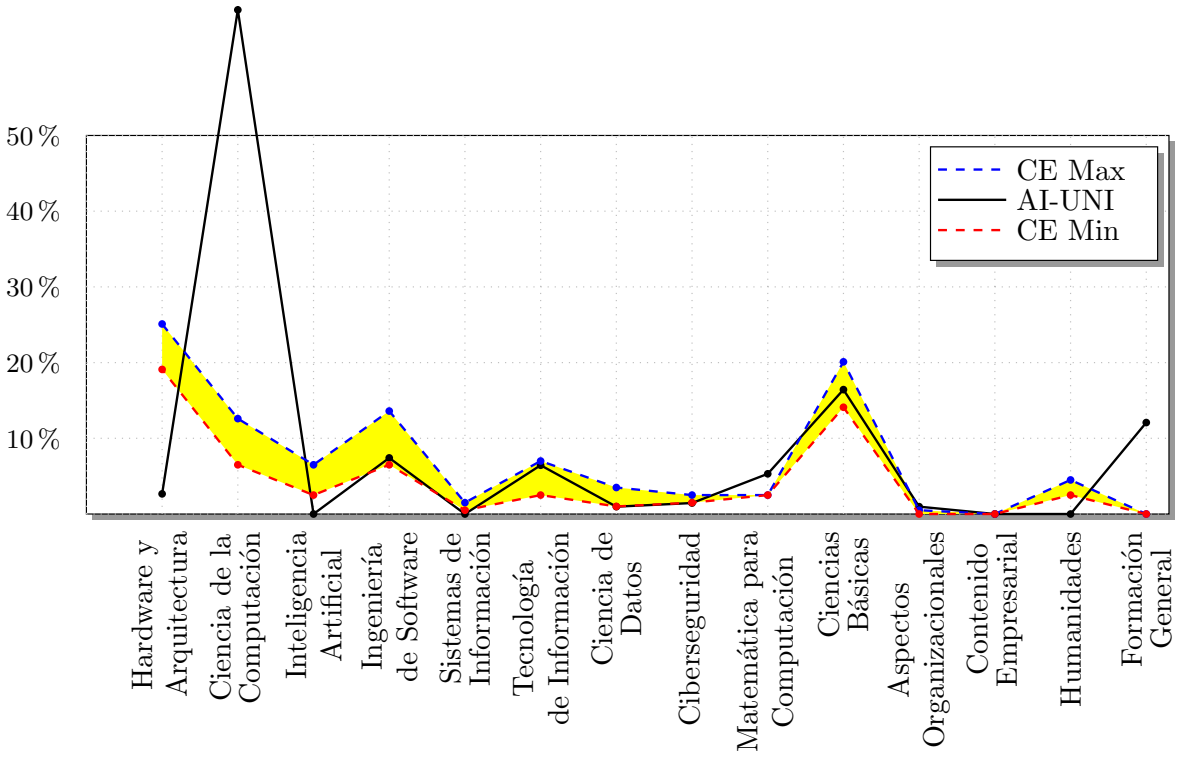


Figura 2.15: Comparación de AI-UNI con CE de ACM/IEEE-CS (Gráfico de curvas).

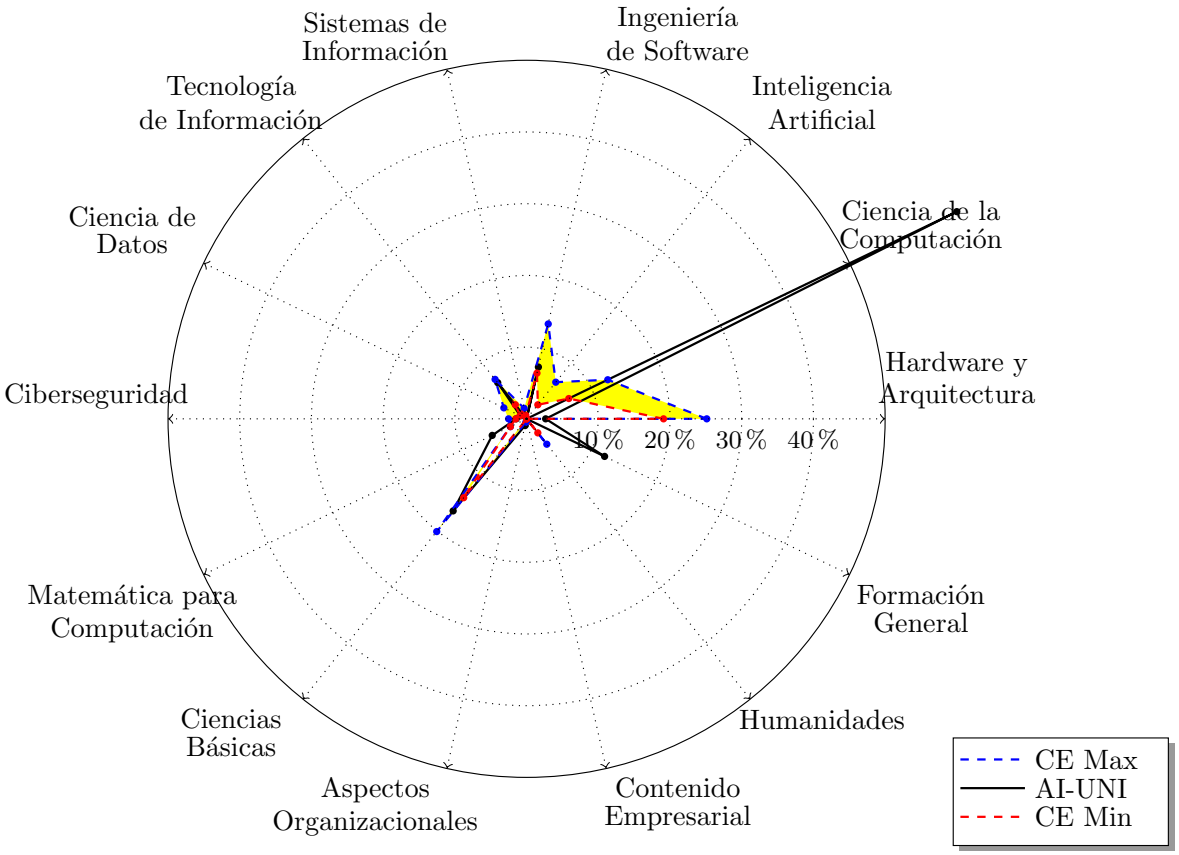


Figura 2.16: Comparación de AI-UNI con CE de ACM/IEEE-CS (Gráfico de araña).

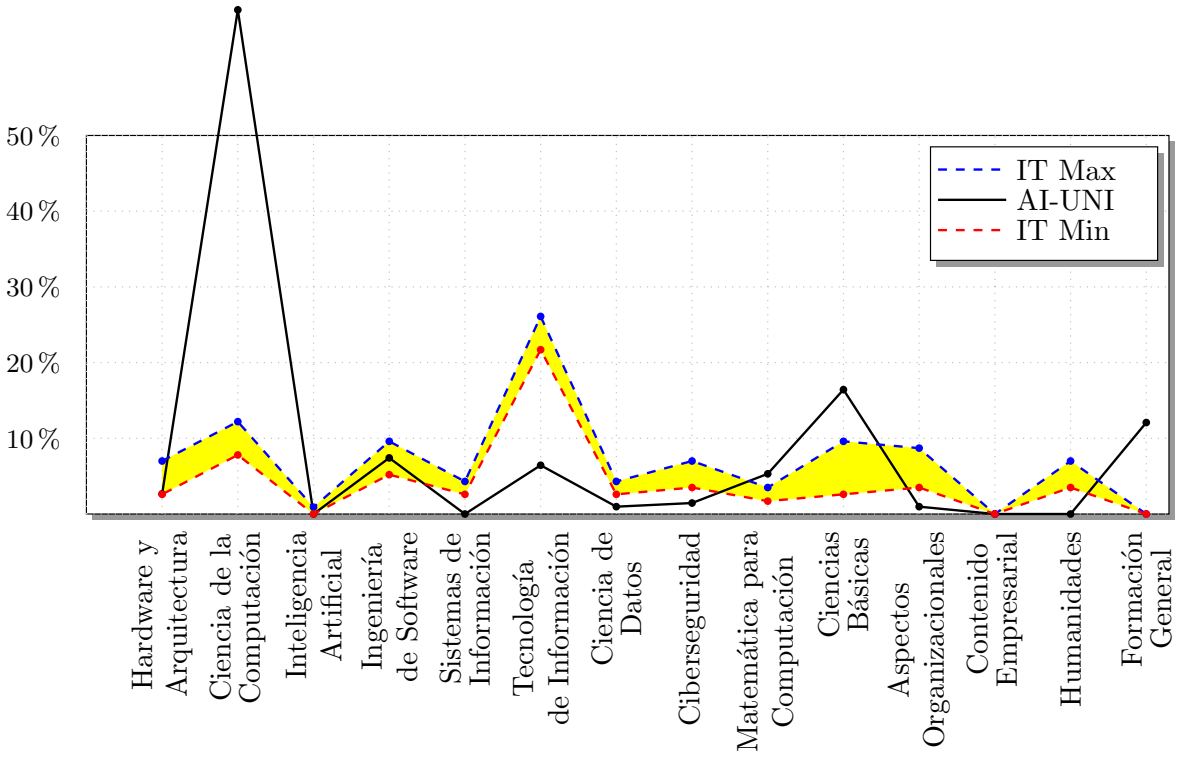


Figura 2.17: Comparación de AI-UNI con IT de ACM/IEEE-CS (Gráfico de curvas).

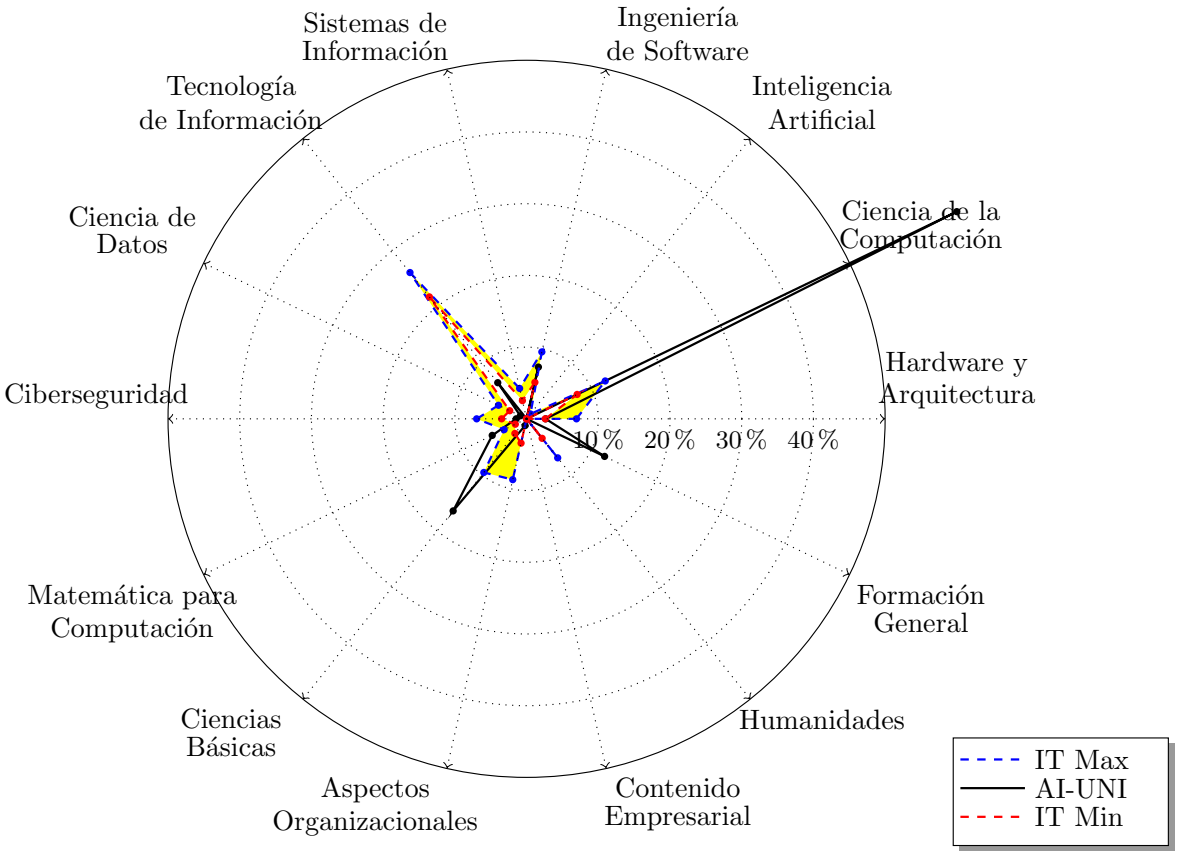


Figura 2.18: Comparación de AI-UNI con IT de ACM/IEEE-CS (Gráfico de araña).

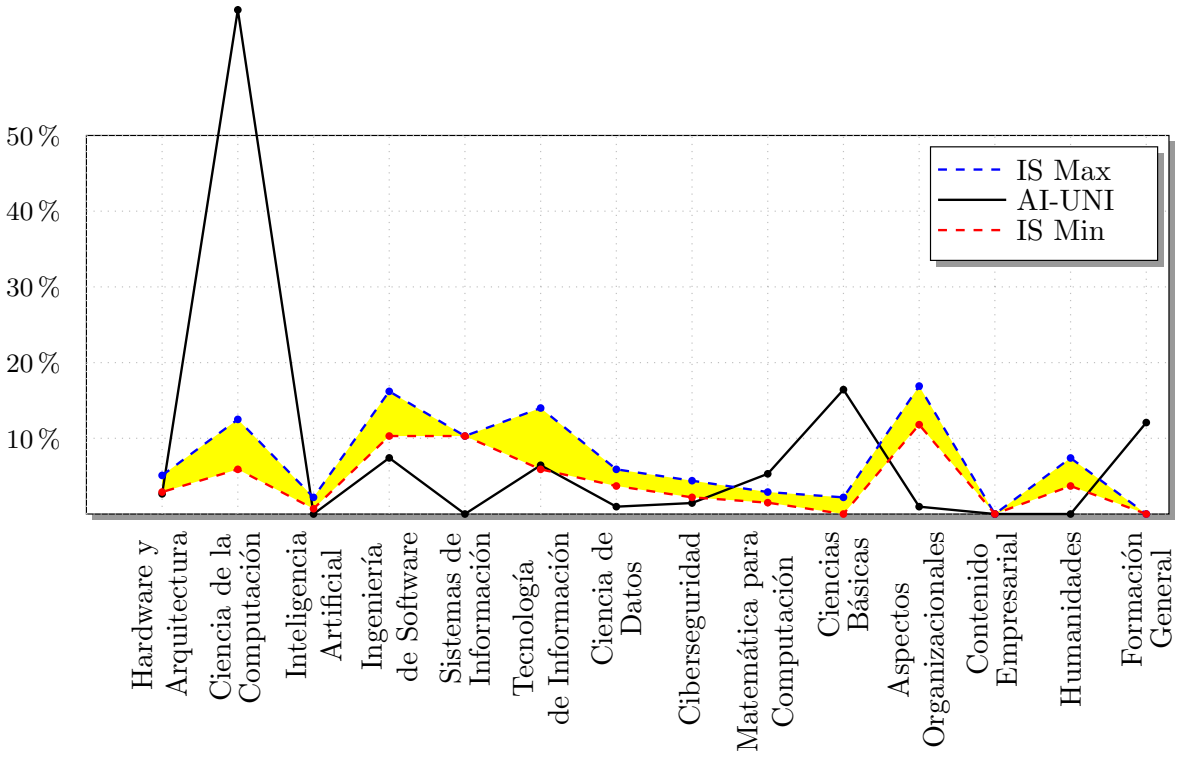


Figura 2.19: Comparación de AI-UNI con IS de ACM/IEEE-CS (Gráfico de curvas).

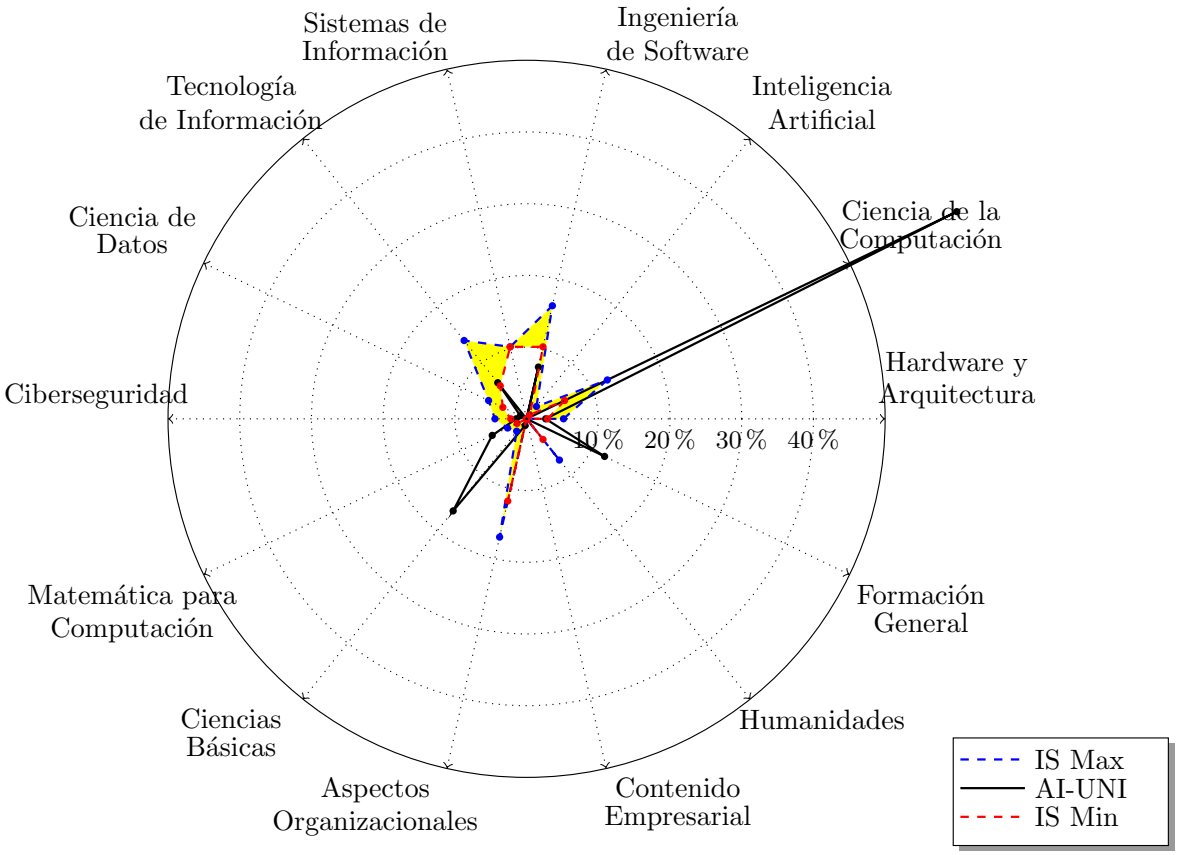


Figura 2.20: Comparación de AI-UNI con IS de ACM/IEEE-CS (Gráfico de araña).

Referencias Bibliográficas

[ACM and IEEE-CS, 2017] ACM and IEEE-CS (2017). Information technology curricula 2017. Technical report, ACM and IEEE-CS.

[ACM Data Science Task Force, 2021] ACM Data Science Task Force (2021). Computing competencies for undergraduate data science curricula. Technical report, Association for Computing Machinery (ACM).

[ACM/IEEE-CS, 2020] ACM/IEEE-CS (2020). Computing curricula 2020. Technical report, ACM Press and IEEE Computer Society Press.

[ACM/IEEE-CS Joint Task Force on Cybersecurity Education, 2017] ACM/IEEE-CS Joint Task Force on Cybersecurity Education (2017). Cybersecurity curricula 2017. Technical report, ACM Press and IEEE Computer Society Press and Association for Information Systems Special Interest Group on Information Security and Privacy (AIS SIGSEC) and International Federation for Information Processing Technical Committee on Information Security Education (IFIP WG 11.8).

[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.

[Díaz-Herrera and Hilburn, 2004] Díaz-Herrera, J. L. and Hilburn, T. B. (2004). Software engineering: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Technical report, ACM, IEEE.

[LeBlanc et al., 2015] LeBlanc, R., Sobel, A., Ben-Menachem, M., Lethbridge, T. C., Díaz-Herrera, J. L., Hilburn, T. B., McGettrick, A., Atlee, J. M., Hawthorne, E. K., Leaney, J., Budgen, D., Matsumoto, Y., Thompson, J. B., Ardis, M., Hislop, G., Offutt, J., Sebern, M., and Visser, W. (2015).

Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Technical report, IEEE Computer Society, ACM.

[Shakelford et al., 2005] Shakelford, R., Cross, J. H., Davies, G., Impagliazzo, J., Kamali, R., LeBlanc, R., Lunt, B., McGettrick, A., Sloan, R., and Topi, H. (2005). Computing curricula 2005. Technical report, ACM/IEEE.

[Soldan et al., 2016] Soldan, D., Aylor, J., Clements, A., Engel, G., Hoelzeman, R., Hughes, E. A., Hughes, J. L., Impagliazzo, J., Jaeger, R. C., Klenke, R., Lyon, D. A., McGettrick, A., Nelson, V. P., Neebel, D. J., Page, I., Peterson, G. D., Ranganathan, N., Sloan, R., Srimani, P. K., Theys, M. D., Wolf, W., and Varanasi, M. (2016). Computer engineering curricula 2016. Technical report, ACM and IEEE-CS.

[The Joint ACM/AIS IS2020 Task Force, 2020] The Joint ACM/AIS IS2020 Task Force (2020). A competency model for undergraduate programs in information systems. Technical report, ACM Press and AIS Press.

Capítulo 3

Contenido detallado por curso

Primer Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BIC101 Introducción a la Computación (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**: Ninguno

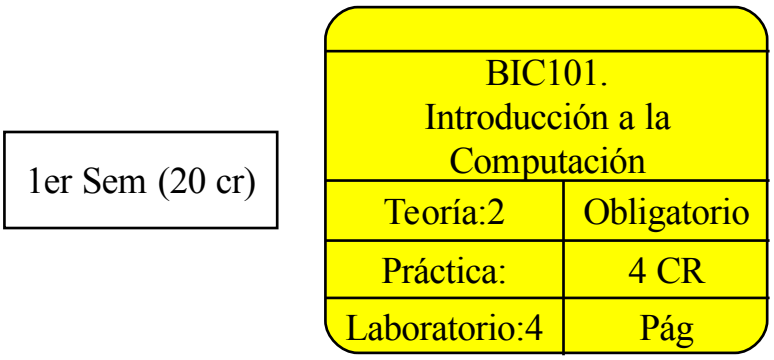


Figura 3.1: Mapa de Conexión. BIC101 Introducción a la Computación

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*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Familiarizarse)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Introducción a la Programación y su Historia, 4 hr(s)

Competencias: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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: AG-C08,AG-C12

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BMA101 Álgebra Lineal (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 3 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

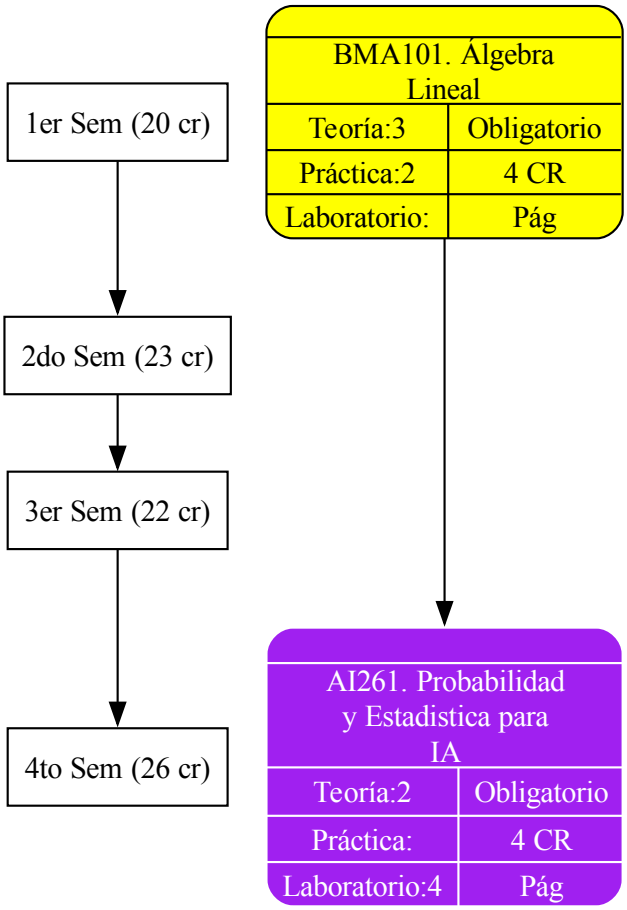


Figura 3.2: Mapa de Conexión. BMA101 Álgebra Lineal

2. Fundamentación

El álgebra lineal es una herramienta matemática fundamental compartida por la ciencia de la computación y las disciplinas de ingeniería, proporcionando las bases para el análisis estructural, las redes de circuitos, los sistemas de control, el procesamiento de datos y el diseño de algoritmos. Este curso proporciona una base sólida en los conceptos y técnicas del álgebra lineal elemental y avanzada, con énfasis en su aplicación en contextos científicos y de ingeniería.

3. Objetivos del curso.

- Comprender los conceptos fundamentales del álgebra lineal, incluyendo espacios vectoriales, transformaciones lineales y teoría de matrices.
- Aplicar las técnicas del álgebra lineal para resolver sistemas de ecuaciones y problemas de valores/vectores propios provenientes de contextos científicos y de ingeniería.
- Desarrollar habilidades de razonamiento abstracto y pensamiento lógico para abordar problemas matemáticos.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Matrices y Sistemas de Ecuaciones Lineales, 19 hr(s)

Competencias: AG-C07

Tópicos: **I:** Álgebra de matrices: suma, multiplicación por escalar, producto de matrices, transpuesta, tipos de matrices e inversa de una matriz **II:** Resolución de sistemas de ecuaciones lineales mediante eliminación gaussiana y Gauss-Jordan; forma escalonada, rango y consistencia de sistemas **III:** Determinantes: definición, propiedades, regla de Cramer e interpretación geométrica **IV:** Descomposiciones matriciales: LU, QR y descomposición en valores singulares (SVD)

Objetivos de Aprendizaje **I:** Identificar tipos de matrices y realizar operaciones básicas de álgebra matricial, incluyendo el cálculo de la inversa de una matriz [Familiarizarse] **II:** Resolver sistemas de ecuaciones lineales mediante eliminación gaussiana o Gauss-Jordan, determinando el rango y la consistencia del sistema [Usar] **III:** Aplicar la regla de Cramer y las descomposiciones LU, QR y SVD para resolver sistemas lineales y factorizar matrices [Evaluar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

5.2 Espacios Vectoriales y Transformaciones Lineales, 14 hr(s)

Competencias: AG-C07

Tópicos: **I:** Espacios y subespacios vectoriales; bases, dimensión e independencia lineal **II:** Transformaciones lineales, núcleo e imagen, teorema rango-nulidad y representaciones matriciales

Objetivos de Aprendizaje **I:** Identificar si un conjunto con operaciones dadas forma un espacio vectorial y determinar una base [Familiarizarse] **II:** Determinar el núcleo, la imagen y la representación matricial de una aplicación lineal, verificando el teorema de rango-nulidad [Usar] **III:** Aplicar transformaciones lineales para modelar y resolver problemas de cambio de base y composición de aplicaciones [Evaluar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

5.3 Valores Propios, Producto Interno y Formas Cuadráticas, 15 hr(s)

Competencias: AG-C07

Tópicos: **I:** Valores y vectores propios; diagonalización y el teorema espectral **II:** Espacios con producto interno, ortogonalidad, proceso de Gram-Schmidt y mínimos cuadrados **III:** Formas cuadráticas y definitud: clasificación en positiva, negativa e indefinida mediante valores propios

Objetivos de Aprendizaje **I:** Identificar los valores propios y vectores propios de una matriz y determinar si es diagonalizable [Familiarizarse] **II:** Calcular la diagonalización de matrices mediante el teorema espectral y aplicar el proceso de Gram-Schmidt para obtener bases ortogonales [Usar]

III: Aplicar los métodos de mínimos cuadrados y la clasificación de formas cuadráticas mediante valores propios para resolver problemas de aproximación y optimización [Evaluar]

Bibliografía: [Strang, 2016, Lay et al., 2016]

6. Bibliografía

[Lay et al., 2016] Lay, D. C., Lay, S. R., and McDonald, J. J. (2016). *Linear Algebra and Its Applications*. Pearson.

[Strang, 2016] Strang, G. (2016). *Introduction to Linear Algebra*. Wellesley-Cambridge Press.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BMA102 Cálculo Diferencial (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

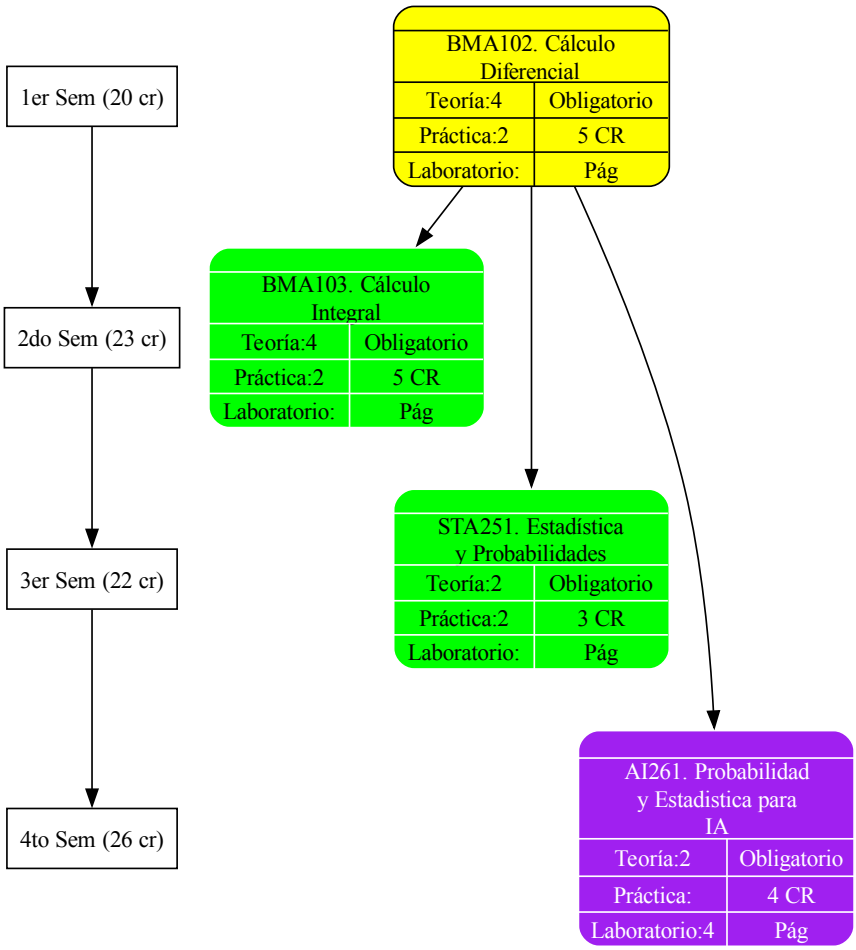


Figura 3.3: Mapa de Conexión. BMA102 Cálculo Diferencial

2. Fundamentación

El cálculo diferencial es una herramienta matemática fundamental compartida por la ciencia de la computación y las disciplinas de ingeniería, proporcionando las bases para el estudio riguroso del cambio, el análisis de razones de cambio, la optimización de sistemas y funciones, y el modelado de fenómenos continuos. Este curso proporciona una base sólida en los conceptos y técnicas del cálculo diferencial de una variable, con énfasis en su aplicación en contextos científicos y de ingeniería.

3. Objetivos del curso.

- Comprender rigurosamente los conceptos de límite, continuidad y derivada de funciones de una variable.
- Aplicar las reglas de diferenciación para calcular derivadas de funciones algebraicas, compuestas e implícitas provenientes de contextos científicos y de ingeniería.
- Utilizar el cálculo diferencial para resolver problemas de optimización, razones de cambio relacionadas y análisis del comportamiento de funciones.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Límites y Continuidad, 12 hr(s)

Competencias: AG-C07

Tópicos: **I:** Definición épsilon-delta de un límite y técnicas de demostración formal **II:** Leyes de los límites, límites laterales y límites en el infinito **III:** Continuidad: tipos, propiedades y el Teorema del Valor Intermedio **IV:** Regla de L'Hôpital y formas indeterminadas **V:** Límites de sucesiones: convergencia, Teorema del Sándwich y Bolzano-Weierstrass

Objetivos de Aprendizaje **I:** Enunciar la definición épsilon-delta de un límite y explicar su significado geométrico [Familiarizarse] **II:** Evaluar límites usando leyes de límites, la regla de L'Hôpital y análisis asintótico [Usar] **III:** Demostrar continuidad o discontinuidad de una función y aplicar el Teorema del Valor Intermedio [Evaluar]

Bibliografía: [Stewart, 2015, Larson and Edwards, 2014]

5.2 Cálculo Diferencial, 36 hr(s)

Competencias: AG-C07

Tópicos: **I:** La derivada como límite; interpretaciones geométricas y físicas; diferenciabilidad **II:** Reglas del producto, cociente y cadena; diferenciación implícita; tasas relacionadas **III:** Teorema de Rolle, Teorema del Valor Medio y criterios de monotonía **IV:** Derivadas de orden superior, polinomios de Taylor y estimación del residuo **V:** Extremos: pruebas de la primera y segunda derivada, trazado de curvas y optimización aplicada

Objetivos de Aprendizaje **I:** Interpretar la derivada como tasa de cambio y como pendiente de la recta tangente [Familiarizarse] **II:** Aplicar las reglas de diferenciación y la diferenciación implícita para calcular derivadas de funciones compuestas e implícitas [Usar] **III:** Resolver problemas de optimización y trazar curvas usando el Teorema del Valor Medio y las pruebas de derivadas [Evaluar]

Bibliografía: [Stewart, 2015, Larson and Edwards, 2014]

5.3 Ecuaciones Paramétricas y Coordenadas Polares, 6 hr(s)

Competencias: AG-C07

Tópicos: **I:** Curvas paramétricas y su cálculo diferencial, aplicadas a trayectorias en cinemática **II:** Coordenadas polares y su conversión a coordenadas cartesianas

Objetivos de Aprendizaje **I:** Calcular la velocidad y aceleración de una trayectoria descrita paramétricamente [Usar] **II:** Convertir entre coordenadas polares y cartesianas al modelar geometrías de ingeniería [Usar]

Bibliografía: [Stewart, 2016, Thomas et al., 2014]

6. Bibliografía

[Larson and Edwards, 2014] Larson, R. and Edwards, B. H. (2014). *Calculus*. Cengage Learning.

[Stewart, 2015] Stewart, J. (2015). *Calculus: Early Transcendentals*. Cengage Learning.

[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 de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BQU101 Química I (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**: Ninguno

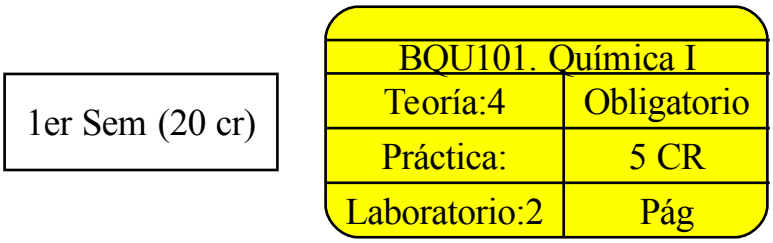


Figura 3.4: Mapa de Conexión. BQU101 Química I

2. Fundamentación

La química proporciona la base para comprender la composición, estructura y propiedades de la materia – conocimiento requerido en los programas de computación e ingeniería por igual, como parte de una sólida formación en ciencias naturales. Este curso introduce los principios básicos de la química general y los extiende al equilibrio químico y la electroquímica, junto con la química aplicada del agua, los polímeros y el cemento, temas que reaparecen en la ciencia de materiales, el análisis ambiental y la práctica de la ingeniería.

3. Objetivos del curso.

- Comprender la estructura de la materia a nivel atómico y molecular.
- Aplicar los principios de estequiometría para realizar cálculos químicos.
- Comprender los diferentes tipos de enlaces químicos y su influencia en las propiedades de las sustancias.
- Aplicar el equilibrio químico, incluidos los equilibrios ácido-base y de solubilidad, para predecir el comportamiento de los sistemas acuosos.
- Aplicar principios electroquímicos para analizar la corrosión de metales y del concreto armado.
- Relacionar la composición y propiedades del agua, los polímeros y el cemento con sus aplicaciones en la ciencia de materiales y en la ingeniería civil.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Materia y Energía, 8 hr(s)

Competencias: AG-C07

Tópicos: **I:** Clasificación de la materia: elementos, compuestos, mezclas homogéneas y heterogéneas **II:** Propiedades físicas y químicas, y cambios físicos y químicos de la materia **III:** Sistema Internacional de unidades, cifras significativas y análisis dimensional **IV:** Formas de energía, calor y temperatura; ley de conservación de la masa y la energía **V:** El método científico aplicado a la química: observación, hipótesis y experimentación

Objetivos de Aprendizaje **I:** Clasificar una muestra de materia como elemento, compuesto o mezcla, y distinguir mezclas homogéneas de heterogéneas [Familiarizarse] **II:** Aplicar el análisis dimensional y las reglas de cifras significativas en cálculos de mediciones químicas [Usar] **III:** Relacionar los cambios de energía observados con procesos físicos y químicos específicos, aplicando la ley de conservación de la energía [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.2 Fundamentos de la Estructura Atómica, 10 hr(s)

Competencias: AG-C07

Tópicos: **I:** Evolución histórica del modelo atómico: Dalton, Thomson, Rutherford y Bohr **II:** Partículas subatómicas, número atómico, número de masa e isótopos **III:** Configuración electrónica, números cuánticos y el principio de Aufbau **IV:** Organización de la tabla periódica: grupos, periodos y bloques **V:** Tendencias periódicas: radio atómico, energía de ionización, afinidad electrónica y electronegatividad

Objetivos de Aprendizaje **I:** Describir la evolución histórica del modelo atómico y las contribuciones de cada modelo a la comprensión actual del átomo [Familiarizarse] **II:** Determinar la configuración electrónica de un átomo o ion a partir de su posición en la tabla periódica [Usar] **III:** Predecir tendencias periódicas (radio atómico, energía de ionización) para comparar dos o más elementos [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.3 Enlace Químico, 10 hr(s)

Competencias: AG-C07

Tópicos: **I:** Enlace iónico: formación, energía reticular y propiedades de los compuestos iónicos **II:** Enlace covalente, estructuras de Lewis y el concepto de resonancia **III:** Geometría molecular mediante la teoría de repulsión de pares de electrones de valencia (RPECV) **IV:** Electronegatividad, polaridad de enlaces y polaridad molecular **V:** Fuerzas intermoleculares: dipolo-dipolo, puente de hidrógeno y fuerzas de dispersión de London

Objetivos de Aprendizaje **I:** Dibujar estructuras de Lewis correctas, incluyendo casos de resonancia, para moléculas e iones poliatómicos [Usar] **II:** Predecir la geometría molecular y la polaridad de una molécula aplicando la teoría RPECV [Usar] **III:** Relacionar el tipo de fuerza intermolecular presente en una sustancia con sus propiedades físicas observadas (punto de ebullición, solubilidad) [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.4 Estequiometría, 10 hr(s)

Competencias: AG-C07

Tópicos: **I:** Fórmulas químicas, masa molar y el concepto de mol **II:** Balanceo de ecuaciones químicas y su interpretación cuantitativa **III:** Cálculos estequiométricos mol-mol, masa-masa y masa-mol **IV:** Reactivo limitante, reactivo en exceso y rendimiento porcentual **V:** Molaridad, preparación de disoluciones y estequiometría de disoluciones

Objetivos de Aprendizaje I: Balancear ecuaciones químicas por el método de tanteo, verificando la conservación de la masa [Familiarizarse] **II:** Calcular las cantidades de reactivos y productos en una reacción química usando relaciones molares [Usar] **III:** Determinar el reactivo limitante y el rendimiento porcentual de una reacción a partir de datos experimentales [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.5 Estados de la Materia, 10 hr(s)

Competencias: AG-C07

Tópicos: I: Leyes de los gases: Boyle, Charles, Avogadro y la ecuación del gas ideal **II:** Teoría cinético-molecular de los gases y sus aplicaciones **III:** Cambios de fase, curvas de calentamiento y diagramas de fase **IV:** Sólidos cristalinos y amorfos; tipos de sólidos cristalinos **V:** Propiedades coligativas de las disoluciones: disminución de la presión de vapor, elevación ebulloscópica, descenso crioscópico y presión osmótica

Objetivos de Aprendizaje I: Aplicar la ecuación del gas ideal y las leyes de los gases para resolver problemas de presión, volumen y temperatura [Usar] **II:** Interpretar un diagrama de fase para identificar los estados y transiciones de una sustancia a distintas condiciones [Familiarizarse] **III:** Calcular las propiedades coligativas de una disolución (presión de vapor, punto de ebullición, punto de congelación) a partir de su concentración [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.6 Química Ambiental del Agua, 4 hr(s)

Competencias: AG-C07

Tópicos: I: Composición química del agua natural y parámetros de calidad (pH, dureza, oxígeno disuelto, DBO) **II:** Fundamentos de tratamiento de agua: coagulación, floculación y desinfección **III:** Principales contaminantes del aire y del agua y sus fuentes **IV:** Ciclos biogeoquímicos del carbono, nitrógeno y fósforo **V:** Química atmosférica y el efecto invernadero

Objetivos de Aprendizaje I: Interpretar parámetros de calidad del agua para evaluar su aptitud de uso [Usar] **II:** Describir los procesos básicos de tratamiento de agua potable y residual [Familiarizarse] **III:** Identificar las principales fuentes y efectos de los contaminantes del aire y del agua [Familiarizarse] **IV:** Explicar el mecanismo químico del efecto invernadero [Usar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.7 Polímeros y Geosintéticos, 4 hr(s)

Competencias: AG-C07

Tópicos: I: Clasificación de polímeros: termoplásticos, termoestables y elastómeros **II:** Reacciones de polimerización: adición y condensación **III:** Propiedades mecánicas y químicas de los polímeros relevantes para ingeniería **IV:** Geosintéticos: geotextiles, geomembranas y geomallas, y sus aplicaciones **V:** Degradación y durabilidad de polímeros en condiciones ambientales

Objetivos de Aprendizaje I: Clasificar un polímero según su estructura y comportamiento térmico [Usar] **II:** Identificar si una reacción de polimerización es por adición o por condensación [Familiarizarse] **III:** Seleccionar el tipo de geosintético adecuado según la aplicación de ingeniería civil requerida [Usar] **IV:** Evaluar la durabilidad de un polímero frente a la exposición ambiental [Evaluar]

Bibliografía: [Chang and Goldsby, 2016, Mamlouk and Zaniewski, 2017]

5.8 Equilibrio Químico, 4 hr(s)

Competencias: AG-C07

Tópicos: **I:** La constante de equilibrio K_c y K_p, y su relación con la estequiometría de la reacción **II:** El cociente de reacción Q y la predicción del sentido de una reacción **III:** Principio de Le Chatelier: efecto de la concentración, la presión y la temperatura **IV:** Tablas ICE para el cálculo de concentraciones en el equilibrio

Objetivos de Aprendizaje **I:** Escribir la expresión de la constante de equilibrio para una reacción dada [Usar] **II:** Calcular concentraciones en el equilibrio usando tablas ICE [Usar] **III:** Predecir el desplazamiento de un equilibrio ante cambios de concentración, presión o temperatura usando el principio de Le Chatelier [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.9 Equilibrio Ácido-Base, 12 hr(s)

Competencias: AG-C07

Tópicos: **I:** Teorías ácido-base de Arrhenius, Brønsted-Lowry y Lewis **II:** Cálculo de pH y pOH para ácidos y bases fuertes **III:** Equilibrio de ácidos y bases débiles: K_a, K_b y grado de disociación **IV:** Hidrólisis de sales y predicción del carácter ácido, básico o neutro de una disolución **V:** Disoluciones amortiguadoras (buffer) y la ecuación de Henderson-Hasselbalch

Objetivos de Aprendizaje **I:** Calcular el pH de disoluciones de ácidos y bases fuertes y débiles [Usar] **II:** Predecir el carácter ácido-base de una disolución de una sal a partir de la hidrólisis de sus iones [Evaluar] **III:** Calcular el pH de una disolución amortiguadora usando la ecuación de Henderson-Hasselbalch [Usar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.10 Equilibrio de Solubilidad, 6 hr(s)

Competencias: AG-C07

Tópicos: **I:** El producto de solubilidad K_{sp} y su relación con la solubilidad molar **II:** Predicción de precipitación mediante la comparación del cociente de solubilidad con K_{sp} **III:** Efecto del ion común sobre la solubilidad

Objetivos de Aprendizaje **I:** Calcular K_{sp} a partir de la solubilidad molar de una sal, y viceversa [Usar] **II:** Predecir si se formará un precipitado al mezclar dos disoluciones dadas sus concentraciones y el K_{sp} correspondiente [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.11 Electroquímica, 8 hr(s)

Competencias: AG-C07

Tópicos: **I:** Reacciones de oxidación-reducción y balanceo por el método del ion-electrón **II:** Celdas galvánicas, potenciales estándar de electrodo y la ecuación de Nernst **III:** Celdas electrolíticas y electrólisis; leyes de Faraday **IV:** Corrosión electroquímica de metales y métodos de protección **V:** Corrosión del acero de refuerzo en concreto armado

Objetivos de Aprendizaje **I:** Balancear ecuaciones redox por el método del ion-electrón en medio ácido o básico [Usar] **II:** Calcular el potencial de una celda galvánica y predecir la espontaneidad de una reacción redox [Usar] **III:** Aplicar las leyes de Faraday para calcular la masa depositada o el tiempo en un proceso de electrólisis [Usar] **IV:** Analizar las condiciones que favorecen la corrosión de un metal o del acero de refuerzo en concreto, y proponer métodos de protección [Evaluar]

Bibliografía: [Brown et al., 2017, Chang and Goldsby, 2016]

5.12 Química del Cemento, 4 hr(s)

Competencias: AG-C07

Tópicos: I: Composición química del cemento Portland: silicatos, aluminatos y ferroaluminatos de calcio **II:** Reacciones de hidratación del cemento y desarrollo de resistencia **III:** Rol de cada compuesto (C_3S, C_2S, C_3A, C_4AF) en las propiedades del cemento **IV:** Materiales cementantes suplementarios: puzolanas, escoria y cenizas volantes

Objetivos de Aprendizaje I: Describir la composición química del cemento Portland y sus principales compuestos [Familiarizarse] **II:** Explicar el proceso de hidratación del cemento y su relación con el desarrollo de resistencia [Usar] **III:** Evaluar el efecto de materiales cementantes suplementarios sobre las propiedades del concreto [Evaluar]

Bibliografía: [Mamlouk and Zaniewski, 2017, Chang and Goldsby, 2016]

6. Bibliografía

[Brown et al., 2017] Brown, T. L., Jr., H. E. L., Bursten, B. E., Murphy, C. J., and Woodward, P. M. (2017). *Chemistry: The Central Science*. Pearson.

[Chang and Goldsby, 2016] Chang, R. and Goldsby, K. A. (2016). *Chemistry*. McGraw-Hill Education.

[Mamlouk and Zaniewski, 2017] Mamlouk, M. S. and Zaniewski, J. P. (2017). *Materials for Civil and Construction Engineers*. Pearson, 4th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BEI101 Inglés I (Obligatorio)

1. Información general

- **Semestre**: 1er Sem. **Créditos**: 2
- **Horas del curso**: **Práctica**: 4 horas;
- **Prerrequisitos**: Ninguno

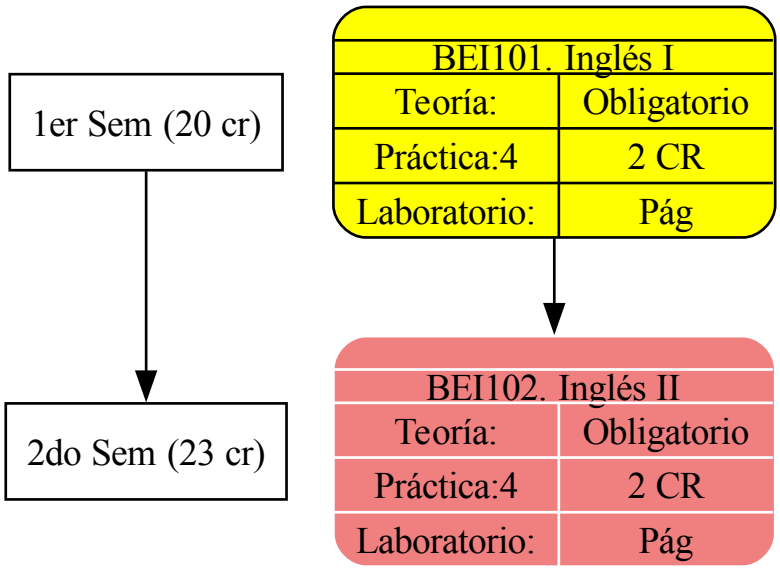


Figura 3.5: Mapa de Conexión. BEI101 Inglés I

2. Fundamentación

El inglés es el idioma principal de la ciencia, la ingeniería y la tecnología en el mundo. Para los estudiantes de computación e ingeniería, dominarlo significa acceder directamente a documentación técnica, artículos de investigación y colaboración profesional sin barreras de traducción. Este curso desarrolla las habilidades fundamentales del inglés utilizando contextos extraídos de sistemas computacionales, hardware y el trabajo cotidiano de ingenieros y científicos.

3. Objetivos del curso.

- Adquirir estructuras gramaticales básicas del inglés aplicadas a contextos de computación e ingeniería.
- Identificar y utilizar vocabulario técnico relacionado con hardware, software y conceptos científicos.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Familiarizarse)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Bienvenido a la Computación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Verbo To Be: Afirmativo, Negativo e Interrogativo. **II:** Pronombres personales y adjetivos posesivos. **III:** Expresiones numéricas (1-100): Puertos, direcciones y fundamentos binarios. **IV:** Expresiones sociales en entornos académicos y de laboratorio.

Objetivos de Aprendizaje I: Al finalizar la unidad, el estudiante utiliza el verbo 'to be' para presentarse como estudiante de computación y describir propiedades básicas de sistemas computacionales (e.g., "The processor is fast. It is a 64-bit CPU.").

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.2 Partes y Sistemas, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Posesivo 's y sustantivos plurales aplicados a componentes (the CPU's speed, the system's memory). **II:** Componentes de hardware y sus relaciones. **III:** Estructuras Adjetivo + Sustantivo para descripciones técnicas. **IV:** Interacciones en el laboratorio: Preguntar sobre equipos y herramientas.

Objetivos de Aprendizaje I: El estudiante describe componentes de hardware y sus relaciones utilizando posesivos y frases nominales (e.g., "the motherboard's slots", "the server's processors").

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.3 Cómo Funciona, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Presente Simple I (Tercera persona: He/She/It) para describir procesos. **II:** Vocabulario: Cómo operan programas, algoritmos y dispositivos. **III:** Expresar la hora e interpretar relojes y horarios de sistemas. **IV:** Verbos de operación: runs, compiles, executes, stores, transmits.

Objetivos de Aprendizaje I: El estudiante explica cómo funciona un programa o dispositivo utilizando el presente simple en tercera persona (e.g., "The compiler translates source code. It generates an executable file.").

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.4 Lo que Hacen los Ingenieros, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Presente Simple II (I/You/We/They) para describir rutinas profesionales. **II:** Preguntas y negaciones con 'Do/Does' en contextos técnicos. **III:** Roles en ingeniería: developer, analyst, researcher, systems engineer. **IV:** Adverbios de frecuencia para describir ciclos de trabajo y pruebas.

Objetivos de Aprendizaje I: El estudiante describe las tareas diarias y los roles profesionales de ingenieros y científicos computacionales utilizando el presente simple con precisión.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.5 Dentro de una Computadora, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Existencial: There is / There are aplicado a descripciones de arquitecturas. **II:** Preposiciones de lugar: componentes en una placa madre, archivos en un directorio. **III:** Vocabulario: memoria, almacenamiento, buses, ranuras, puertos, capas. **IV:** Descripción de topología de red e infraestructura física.

Objetivos de Aprendizaje I: El estudiante describe la organización espacial de componentes de hardware y sistemas de archivos, permitiéndole interpretar y producir descripciones técnicas de arquitecturas.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.6 ¿Pueden Pensar las Computadoras?, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Verbo modal Can/Can't para describir capacidades y limitaciones computacionales. **II:** Adverbios de modo aplicados al rendimiento de sistemas. **III:** Pasado del verbo 'To Be' (Was/Were): sistemas computacionales tempranos y pioneros. **IV:** Expresiones básicas para soporte técnico y resolución de problemas.

Objetivos de Aprendizaje I: El estudiante describe lo que los sistemas computacionales pueden y no pueden hacer, e identifica hechos básicos sobre la historia temprana de la computación utilizando estructuras de pasado simple.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.7 Historia de la Computación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Pasado Simple: Verbos regulares e irregulares para narrar la historia tecnológica. **II:** Expresiones de tiempo pasado: invented, developed, published, launched. **III:** Figuras clave: Turing, Babbage, Lovelace, von Neumann. **IV:** Vocabulario: Generaciones de computadoras, hitos, inventos.

Objetivos de Aprendizaje I: El estudiante narra eventos clave en la historia de la computación y la ciencia utilizando el pasado simple correctamente, identificando las contribuciones de figuras fundacionales.

Bibliografía: [Soars and John, 2002d, Cambridge, 2006, MacGrew, 1999]

5.8 Mis Metas Profesionales en Computación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Introducción al inglés técnico para computación e ingeniería. **II:** Lectura y comprensión de especificaciones técnicas simples. **III:** Trabajo en equipos de ingeniería multiculturales: normas de comunicación. **IV:** Presentación oral: "Mi rol en el campo de la computación".

Objetivos de Aprendizaje I: El estudiante presenta sus metas profesionales en computación utilizando vocabulario técnico básico y las relaciona con los roles y responsabilidades de profesionales en software y hardware.

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.

Segundo Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

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

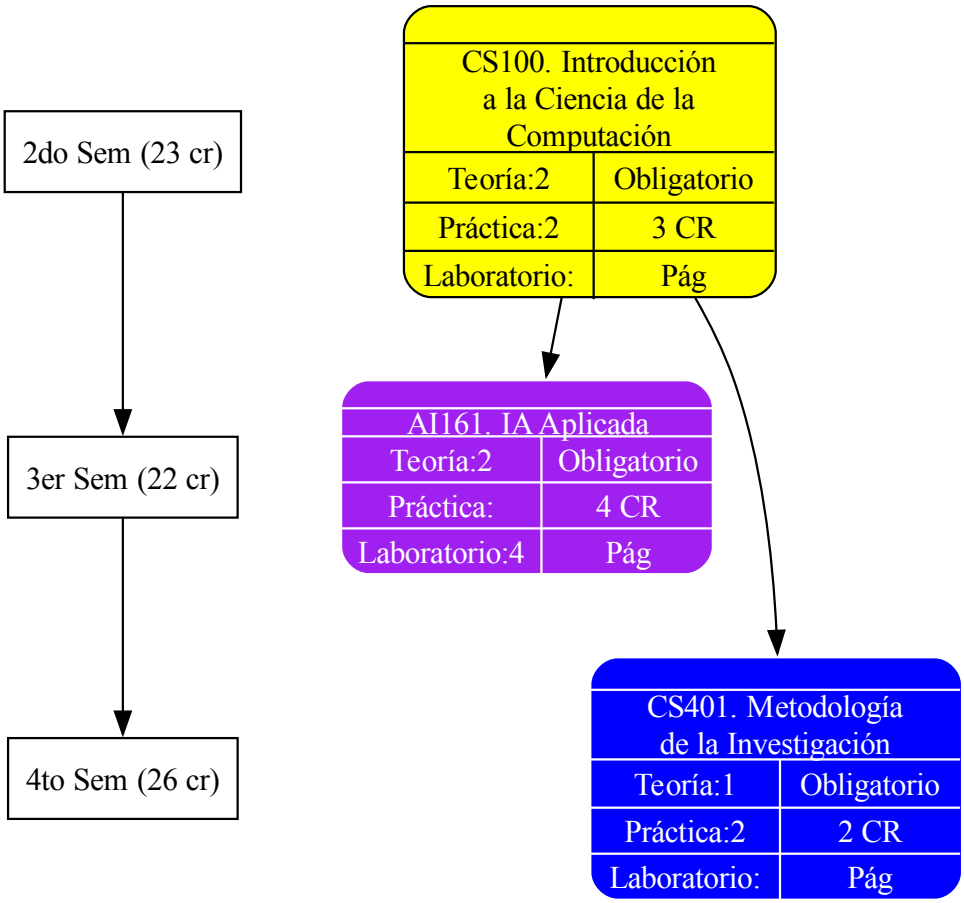


Figura 3.6: Mapa de Conexión. CS100 Introducción a la Ciencia de la Computación

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*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Familiarizarse)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Familiarizarse)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Familiarizarse)

5. Contenido

5.1 Pensamiento Computacional (Parte I), 4 hr(s)

Competencias: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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: AG-C02,AG-C08,AG-C12

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

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**: Ninguno

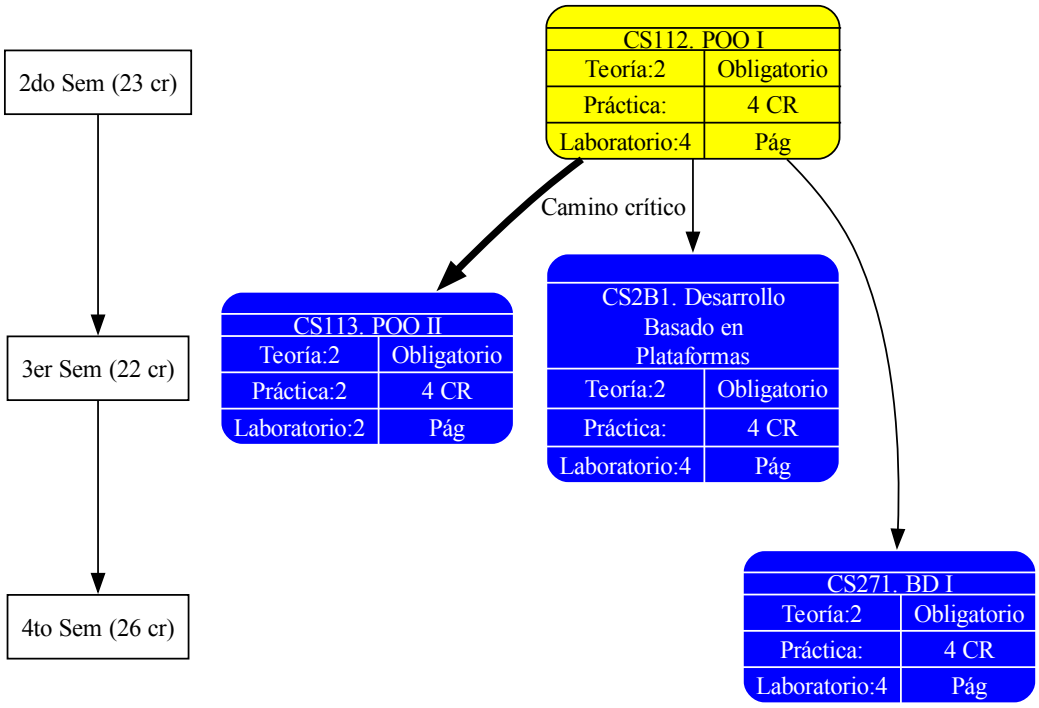


Figura 3.7: Mapa de Conexión. CS112 Programación Orientada a Objetos I

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Pragmática del Lenguaje, 2 hr(s)

Competencias: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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++ How to Program*. 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 de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS1D1 Estructuras Discretas (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**: Ninguno

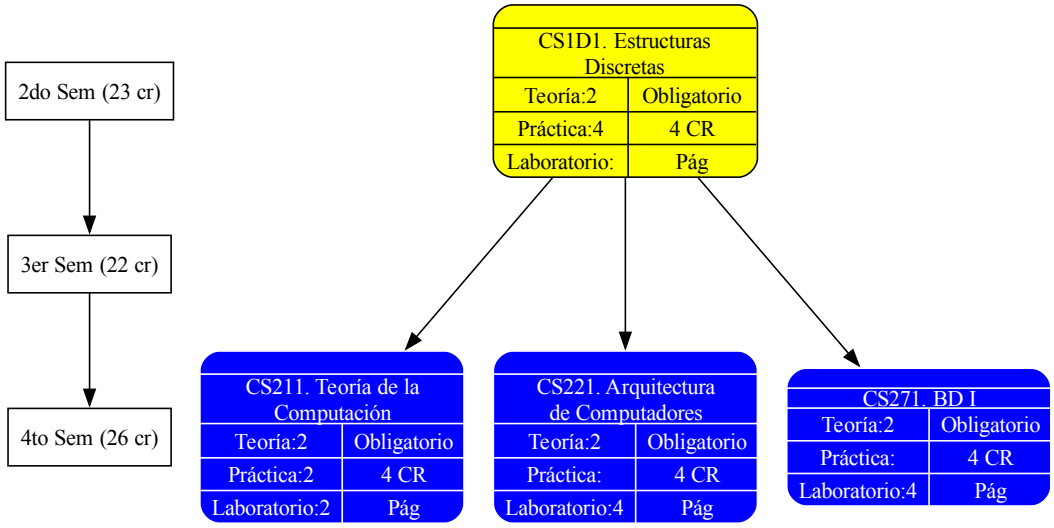


Figura 3.8: Mapa de Conexión. CS1D1 Estructuras Discretas

2. Fundamentación

Las estructuras discretas proporcionan el fundamento teórico para la ciencia de la computación. Este curso introduce las herramientas matemáticas esenciales utilizadas para modelar problemas computacionales, incluyendo lógica, teoría de conjuntos y funciones. Comprender estos conceptos es crítico para desarrollar habilidades de razonamiento riguroso y para el estudio posterior de algoritmos, bases de datos y lenguajes formales.

3. Objetivos del curso.

- Comprender los principios de la lógica matemática y las técnicas de demostración.
- Realizar operaciones con conjuntos, funciones y relaciones.
- Aplicar técnicas básicas de conteo y relaciones de recurrencia a problemas.
- Desarrollar la capacidad de pensar de manera abstracta y resolver problemas formalmente.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Lógica y Fundamentos, 12 hr(s)

Competencias: AG-C07,AG-C12

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, 12 hr(s)

Competencias: AG-C07,AG-C12

Tópicos: **I:** Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones **II:** Lógica avanzada: sistemas de demostración, completitud, decidibilidad **III:** Teoría avanzada de conjuntos: conjuntos infinitos, argumento diagonal de Cantor **IV:** Relaciones avanzadas: clausuras, retículos **V:** Funciones avanzadas: funciones recursivas, computabilidad

Objetivos de Aprendizaje **I:** Construir demostraciones lógicas simples [Aplicar] **II:** Explicar conceptos de conjuntos infinitos y cardinalidad [Explicar] **III:** Aplicar operaciones de clausura a relaciones [Aplicar] **IV:** Determinar si una relación es una relación de equivalencia [Aplicar]

Bibliografía: [Rosen, 2019, Velleman, 2019]

5.3 Relaciones, Funciones y Grafos, 12 hr(s)

Competencias: AG-C07,AG-C12

Tópicos: **I:** Relaciones: propiedades, relaciones de equivalencia, órdenes parciales **II:** Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa **III:** Grafos: definiciones, caminos, ciclos, árboles, conectividad **IV:** Teoría avanzada de grafos: planaridad, coloración, redes

Objetivos de Aprendizaje **I:** Determinar si una relación es una relación de equivalencia [Aplicar] **II:** Componer funciones y encontrar inversas [Aplicar] **III:** Definir conceptos básicos de grafos [Explicar] **IV:** Resolver problemas básicos de grafos [Aplicar] **V:** Analizar propiedades avanzadas de grafos [Evaluar]

Bibliografía: [Rosen, 2019, Scheinerman, 2012]

5.4 Conteo y Recurrencia, 12 hr(s)

Competencias: AG-C07,AG-C12

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 [Explicar] **II:** Aplicar técnicas de conteo para resolver problemas [Aplicar] **III:** Definir relaciones de recurrencia [Explicar] **IV:** Resolver relaciones de recurrencia simples [Aplicar] **V:** Desarrollar funciones generadoras para problemas de conteo [Crear]

Bibliografía: [Rosen, 2019, Grimaldi, 2003]

6. Bibliografía

[Grimaldi, 2003] Grimaldi, R. P. (2003). *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson, 5th 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.

[Scheinerman, 2012] Scheinerman, E. A. (2012). *Mathematics: A Discrete Introduction*. Cengage Learning, 3rd edition.

[Velleman, 2019] Velleman, D. J. (2019). *How to Prove It: A Structured Approach*. Cambridge University Press, 3rd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BMA103 Cálculo Integral (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - BMA102 Cálculo Diferencial (1er Sem-Pág 57)

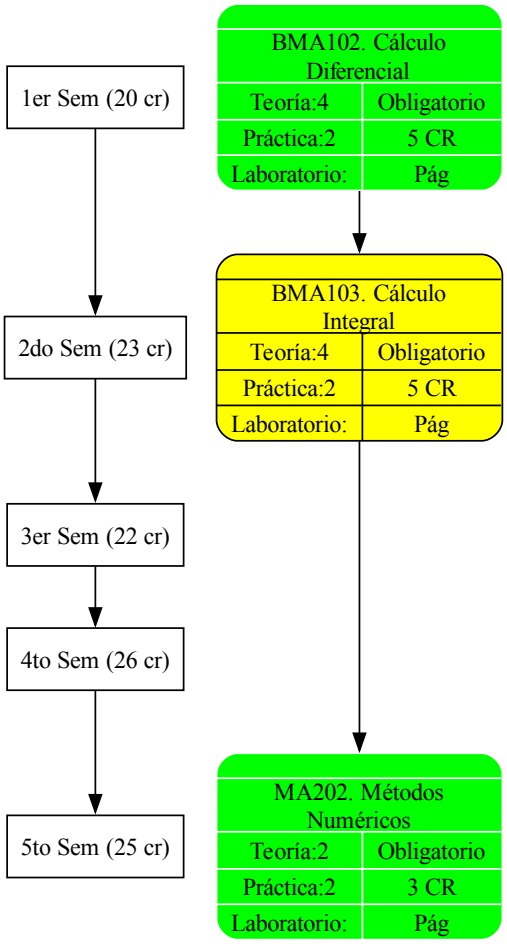


Figura 3.9: Mapa de Conexión. BMA103 Cálculo Integral

2. Fundamentación

El cálculo integral es una herramienta matemática fundamental compartida por la ciencia de la computación y las disciplinas de ingeniería, esencial para modelar y resolver problemas que involucran acumulación, cambio y áreas bajo curvas. Este curso proporciona las bases del cálculo integral, incluyendo técnicas de integración y sus aplicaciones en contextos científicos y de ingeniería.

3. Objetivos del curso.

- Comprender el concepto de integral definida e indefinida.

- Aplicar diversas técnicas de integración para resolver problemas.
- Utilizar el cálculo integral para modelar y resolver problemas en contextos científicos y de ingeniería.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Antiderivadas y Técnicas de Integración, 20 hr(s)

Competencias: AG-C07

Tópicos: I: El Teorema Fundamental del Cálculo y la antiderivada **II:** Técnicas de integración: sustitución, partes, fracciones parciales y sustitución trigonométrica

Objetivos de Aprendizaje I: Enunciar el Teorema Fundamental del Cálculo y explicar la relación entre la diferenciación y la integración [Familiarizarse] **II:** Calcular integrales definidas e indefinidas usando técnicas estándar, incluyendo sustitución e integración por partes [Usar]

Bibliografía: [Stewart, 2015, Larson and Edwards, 2014]

5.2 Integrales Impropias, 8 hr(s)

Competencias: AG-C07

Tópicos: I: Integrales impropias: criterios de convergencia y pruebas de comparación

Objetivos de Aprendizaje I: Determinar la convergencia de integrales impropias usando criterios de comparación y otras pruebas de convergencia [Evaluar]

Bibliografía: [Stewart, 2015, Larson and Edwards, 2014]

5.3 Sucesiones, Series y Series de Potencias, 20 hr(s)

Competencias: AG-C07

Tópicos: I: Sucesiones y series: pruebas de convergencia (cociente, raíz, comparación, integral) **II:** Series de potencias, radio de convergencia, series de Taylor y Maclaurin

Objetivos de Aprendizaje I: Determinar la convergencia de series de potencias y construir series de Taylor para funciones estándar [Evaluar]

Bibliografía: [Stewart, 2015, Larson and Edwards, 2014]

6. Bibliografía

[Larson and Edwards, 2014] Larson, R. and Edwards, B. H. (2014). *Calculus*. Cengage Learning.

[Stewart, 2015] Stewart, J. (2015). *Calculus: Early Transcendentals*. Cengage Learning.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BFI101 Física I (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 5
- **Horas del curso**: **Teoría**: 4 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**: Ninguno

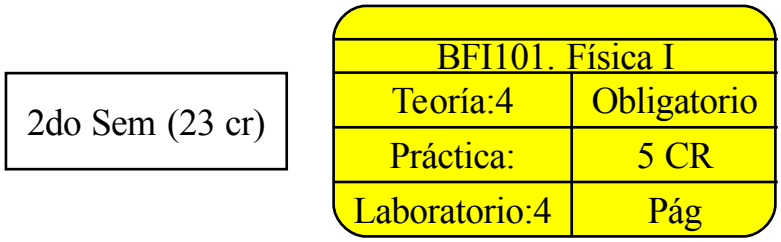


Figura 3.10: Mapa de Conexión. BFI101 Física I

2. Fundamentación

Física I introduce los fundamentos de la mecánica clásica como una disciplina matemática y experimental. A través del estudio de la cinemática, la dinámica newtoniana, los principios de conservación, el movimiento rotacional, la elasticidad, la mecánica de fluidos, las vibraciones y ondas, y una introducción a los fenómenos de transporte, los estudiantes adquieren las herramientas analíticas necesarias para modelar y resolver problemas del mundo físico. Este curso proporciona la base indispensable para el estudio de los fenómenos oscilatorios, ondulatorios y electromagnéticos que se abordan en cursos posteriores.

3. Objetivos del curso.

- Describir el movimiento de partículas en diferentes sistemas de coordenadas y formular las ecuaciones del movimiento a partir de las leyes de Newton.
- Aplicar los principios de trabajo, energía potencial y conservación de la energía mecánica a sistemas con fuerzas conservativas y no conservativas.
- Resolver problemas de dinámica de sistemas de partículas utilizando la conservación del momento lineal y el análisis de colisiones.
- Analizar el movimiento rotacional y la dinámica del momento angular en sistemas de partículas y cuerpos extendidos.
- Aplicar los módulos elásticos y la ley de Hooke para relacionar el esfuerzo y la deformación en sólidos.
- Aplicar los principios de la hidrostática y la hidrodinámica, incluyendo las ecuaciones de continuidad y de Bernoulli, a problemas de flujo de fluidos.
- Analizar el movimiento armónico simple y la propagación de ondas mecánicas, incluyendo los fenómenos de superposición y resonancia.
- Aplicar las leyes de Fourier y Fick para describir procesos básicos de conducción de calor y difusión de masa.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Cinemática y Dinámica de Partículas, 22 hr(s)

Competencias: AG-C07

Tópicos: **I:** Descripción vectorial de posición, velocidad y aceleración; las tres leyes de Newton y marcos de referencia inerciales **II:** Formulación y solución analítica de ecuaciones de movimiento para fuerzas constantes y variables **III:** Diagramas del espacio de fases (posición vs. velocidad) para visualizar el estado mecánico, la estabilidad y la periodicidad **IV:** Integración numérica de ecuaciones de movimiento usando métodos de Euler, Verlet y Runge-Kutta **V:** Movimiento con restricciones y ligaduras holonómicas relevantes para eslabones robóticos y cuerpos articulados

Objetivos de Aprendizaje **I:** Identificar todas las fuerzas que actúan sobre una partícula en un escenario mecánico dado y trazar el diagrama correcto de cuerpo libre [Familiarizarse] **II:** Implementar un algoritmo de integración numérica como el método de Runge-Kutta para calcular la trayectoria de una partícula bajo fuerzas variables [Usar] **III:** Construir e interpretar diagramas del espacio de fases para caracterizar la estabilidad y periodicidad de un sistema mecánico simulado [Evaluar]

Bibliografía: [Serway and Jewett, 2018, Young and Freedman, 2015]

5.2 Trabajo, Energía y Leyes de Conservación, 12 hr(s)

Competencias: AG-C07

Tópicos: **I:** Teorema trabajo-energía que relaciona el trabajo neto realizado con el cambio en la energía cinética **II:** Funciones de energía potencial y la relación fuerza-gradiente para campos de fuerza conservativos **III:** Conservación de la energía mecánica total y condiciones para su aplicabilidad **IV:** Diagramas de energía para identificar puntos de equilibrio, puntos de retorno y estabilidad **V:** Modelos de colisiones elásticas e inelásticas: intercambio de momento y energía en implementaciones de motores de física

Objetivos de Aprendizaje **I:** Explicar la distinción entre fuerzas conservativas y no conservativas y las condiciones bajo las cuales se conserva la energía mecánica [Familiarizarse] **II:** Aplicar métodos de conservación de energía para calcular velocidades y posiciones en problemas de movimiento unidimensional comunes en implementaciones de motores de física [Usar] **III:** Analizar eventos de colisión elástica e inelástica, calculando velocidades post-colisión y cuantificando la disipación de energía [Evaluar]

Bibliografía: [Serway and Jewett, 2018, Tipler and Mosca, 2007]

5.3 Sistemas de Partículas y Momento Lineal, 8 hr(s)

Competencias: AG-C07

Tópicos: **I:** Definición de centro de masa: $\vec{R}_{CM} = \frac{1}{M}\sum m_i \vec{r}_i$ **II:** Momento lineal de una partícula y de un sistema: $\vec{p} = m\vec{v}$ y $\vec{P} = M\vec{V}_{CM}$ **III:** Segunda ley de Newton para sistemas: $\vec{F}_{ext} = \frac{d\vec{P}}{dt}$ **IV:** Conservación del momento lineal para sistemas aislados **V:** Distinción entre fuerzas internas y externas en un sistema **VI:** Colisiones de dos cuerpos, incluyendo casos elásticos, inelásticos y completamente inelásticos **VII:** Coeficiente de restitución y su significado físico en las colisiones **VIII:** Sistemas con masa variable, como la ecuación del cohete: $M\frac{dV}{dt} = v_{ex}\frac{dM}{dt}$ **IX:** Sistemas de muchas partículas y su dinámica del centro de masa **X:** Distribuciones continuas de masa y el cálculo de su centro de masa mediante integración

Objetivos de Aprendizaje I: Localizar el centro de masa para un conjunto discreto de partículas en Una, Dos y Tres dimensiones [Usar] **II:** Determinar el momento lineal total de un sistema a partir del movimiento de su centro de masa [Usar] **III:** Aplicar la conservación del momento lineal para resolver problemas de colisión en Una y Dos dimensiones [Usar] **IV:** Clasificar colisiones como elásticas, inelásticas o completamente inelásticas basándose en la conservación de la energía cinética [Familiarizarse] **V:** Calcular el coeficiente de restitución a partir de las velocidades antes y después de la colisión [Usar] **VI:** Resolver la ecuación del cohete para determinar la velocidad en función de la masa expulsada [Usar] **VII:** Analizar cómo las fuerzas internas afectan a las partículas individuales pero no al momento total de un sistema [Evaluar] **VIII:** Integrar para hallar el centro de masa de cuerpos continuos simétricos como varillas, discos y esferas [Usar]

Bibliografía: [Young and Freedman, 2015, Tipler and Mosca, 2007]

5.4 Movimiento Rotacional y Momento Angular, 12 hr(s)

Competencias: AG-C07

Tópicos: I: Cinemática rotacional incluyendo desplazamiento angular, velocidad y aceleración (θ, ω, α) **II:** Definición de torque: $\vec{\tau} = \vec{r} \times \vec{F}$ y el cálculo de su magnitud **III:** Momento angular de una partícula: $\vec{L} = \vec{r} \times \vec{p}$ **IV:** Forma rotacional de la segunda ley de Newton: $\vec{\tau}_{net} = \frac{d\vec{L}}{dt}$ **V:** Momento de inercia $I = \sum m_i r_i^2$ y el teorema de los ejes paralelos **VI:** Energía cinética rotacional: $K_{rot} = \frac{1}{2} I\omega^2$ **VII:** Conservación del momento angular para sistemas aislados **VIII:** Movimiento giroscópico y el fenómeno de la precesión **IX:** Dinámica de rodadura sin deslizamiento: $v_{CM} = R\omega$ **X:** El efecto Coriolis como manifestación de la dinámica rotacional en marcos no inerciales

Objetivos de Aprendizaje I: Relacionar las magnitudes cinemáticas lineales y angulares para el movimiento circular [Usar] **II:** Calcular el torque respecto a un eje dado para una fuerza aplicada en un punto específico [Usar] **III:** Determinar el momento de inercia para geometrías simples como una varilla, un aro, un disco o una esfera [Usar] **IV:** Aplicar la conservación del momento angular para analizar cambios en la velocidad de rotación cuando cambia el momento de inercia [Evaluar] **V:** Calcular la energía cinética rotacional de un cuerpo rígido en rotación [Usar] **VI:** Analizar el movimiento de objetos que ruedan sin deslizar por un plano inclinado [Evaluar] **VII:** Predecir la dirección de la precesión para una peonza o un giroscopio en rotación [Evaluar] **VIII:** Explicar el efecto Coriolis en términos de marcos de referencia rotatorios y su efecto sobre los proyectiles [Familiarizarse]

Bibliografía: [Serway and Jewett, 2018, Kleppner and Kolenkow, 2013]

5.5 Elasticidad, 2 hr(s)

Competencias: AG-C07

Tópicos: I: Definición de esfuerzo (fuerza/área) y deformación unitaria $(\Delta L/L)$ **II:** Módulo de Young y ley de Hooke para deformación longitudinal: $\sigma = E\epsilon$ **III:** Módulo de corte (rigidez) y deformación por cizalladura **IV:** Módulo de volumen (compresibilidad) y deformación volumétrica **V:** Límite elástico, región plástica y punto de ruptura en el diagrama esfuerzo-deformación **VI:** Energía elástica almacenada en un material deformado

Objetivos de Aprendizaje I: Aplicar la ley de Hooke escalar para calcular la deformación de una barra bajo carga axial [Usar] **II:** Calcular el módulo de Young, de corte o de volumen a partir de datos experimentales esfuerzo-deformación [Usar] **III:** Identificar la región elástica, el límite elástico y el punto de fractura en una curva esfuerzo-deformación [Familiarizarse] **IV:** Calcular la energía potencial elástica almacenada en un resorte o barra deformada [Usar]

Bibliografía: [Serway and Jewett, 2018, Young and Freedman, 2015]

5.6 Fluidos: Hidrostática, Hidrodinámica y Viscosidad, 10 hr(s)

Competencias: AG-C07

Tópicos: **I:** Densidad, presión y su variación con la profundidad en un fluido en reposo: $p = p_0 + \rho gh$ **II:** Principio de Pascal y sus aplicaciones (prensa hidráulica) **III:** Principio de Arquímedes, empuje y flotación **IV:** Medición de presión con manómetros y barómetros **V:** Ecuación de continuidad para flujo incompresible: $A_1 v_1 = A_2 v_2$ **VI:** Ecuación de Bernoulli y conservación de energía en flujo ideal **VII:** Aplicaciones: tubo de Venturi, teorema de Torricelli, tubo de Pitot **VIII:** Viscosidad y ley de Poiseuille para flujo laminar en tuberías **IX:** Número de Reynolds y transición a flujo turbulento **X:** Ley de Stokes para la fuerza de arrastre sobre una esfera en un fluido viscoso

Objetivos de Aprendizaje **I:** Calcular la presión en un punto de un fluido en reposo dado la profundidad y densidad [Usar] **II:** Aplicar el principio de Arquímedes para determinar si un objeto flota, se hunde o permanece en equilibrio [Usar] **III:** Aplicar las ecuaciones de continuidad y de Bernoulli para resolver problemas de flujo en tuberías de sección variable [Usar] **IV:** Interpretar lecturas de manómetros y barómetros para determinar presiones absolutas y manométricas [Evaluar] **V:** Aplicar la ley de Poiseuille para calcular el caudal de un fluido viscoso en un tubo cilíndrico [Usar] **VI:** Estimar el número de Reynolds de un flujo y predecir si es laminar o turbulento [Evaluar]

Bibliografía: [Serway and Jewett, 2018, Young and Freedman, 2015]

5.7 Vibraciones y Ondas, 14 hr(s)

Competencias: AG-C07

Tópicos: **I:** Movimiento armónico simple (MAS): ecuación de movimiento, posición, velocidad y aceleración **II:** Sistemas oscilantes: resorte-masa y péndulo simple/físico **III:** Energía en el movimiento armónico simple **IV:** Oscilaciones amortiguadas y forzadas; introducción a la resonancia **V:** Ondas transversales y longitudinales; velocidad de propagación **VI:** Ecuación de onda y velocidad de una onda en una cuerda tensa: $v = \sqrt{T/\mu}$ **VII:** Principio de superposición y ondas estacionarias **VIII:** Ondas sonoras, intensidad y nivel de intensidad sonora (decibeles) **IX:** Efecto Doppler para ondas sonoras **X:** Pulsaciones (beats) e interferencia de ondas de frecuencias cercanas

Objetivos de Aprendizaje **I:** Deducir la ecuación de movimiento y el período de un sistema resorte-masa y de un péndulo simple [Usar] **II:** Calcular la energía cinética, potencial y total de un oscilador armónico simple en cualquier instante [Usar] **III:** Analizar la respuesta en amplitud de un oscilador forzado cerca de la frecuencia de resonancia [Evaluar] **IV:** Calcular la velocidad de propagación de una onda en una cuerda a partir de la tensión y la densidad lineal [Usar] **V:** Determinar las frecuencias y longitudes de onda de los modos normales de una onda estacionaria en una cuerda fija [Usar] **VI:** Calcular el nivel de intensidad sonora en decibeles a partir de la intensidad de una onda sonora [Usar] **VII:** Aplicar la fórmula del efecto Doppler para fuente y/o observador en movimiento [Usar]

Bibliografía: [Serway and Jewett, 2018, Young and Freedman, 2015]

5.8 Fenómeno de Transporte, 4 hr(s)

Competencias: AG-C07

Tópicos: **I:** Conducción de calor y la ley de Fourier: $\vec{q} = -k\nabla T$ **II:** Conductividad térmica de materiales y resistencia térmica en paredes compuestas **III:** Difusión molecular y la primera ley de Fick: $\vec{J} = -D\nabla C$ **IV:** Analogía entre el transporte de calor, masa y momento (números de Prandtl y Schmidt)

Objetivos de Aprendizaje I: Aplicar la ley de Fourier para calcular el flujo de calor a través de una pared plana o compuesta [Usar] **II:** Aplicar la primera ley de Fick para calcular el flujo difusivo de una especie en un gradiente de concentración [Usar] **III:** Identificar analogías matemáticas entre los procesos de transporte de calor, masa y momento [Familiarizarse]

Bibliografía: [Serway and Jewett, 2018, Tipler and Mosca, 2007]

6. Bibliografía

[Kleppner and Kolenkow, 2013] Kleppner, D. and Kolenkow, R. J. (2013). *An Introduction to Mechanics*. Cambridge University Press, 2nd edition.

[Serway and Jewett, 2018] Serway, R. A. and Jewett, J. W. (2018). *Physics for Scientists and Engineers with Modern Physics*. Cengage Learning, 10th edition.

[Tipler and Mosca, 2007] Tipler, P. A. and Mosca, G. (2007). *Physics for Scientists and Engineers*. W. H. Freeman and Company, 6th edition.

[Young and Freedman, 2015] Young, H. D. and Freedman, R. A. (2015). *Sears and Zemansky's University Physics with Modern Physics*. Pearson Education, 14th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BEI102 Inglés II (Obligatorio)

1. Información general

- **Semestre**: 2do Sem. **Créditos**: 2
- **Horas del curso**: **Práctica**: 4 horas;
- **Prerrequisitos**:
  - BEI101 Inglés I (1er Sem-Pág 65)

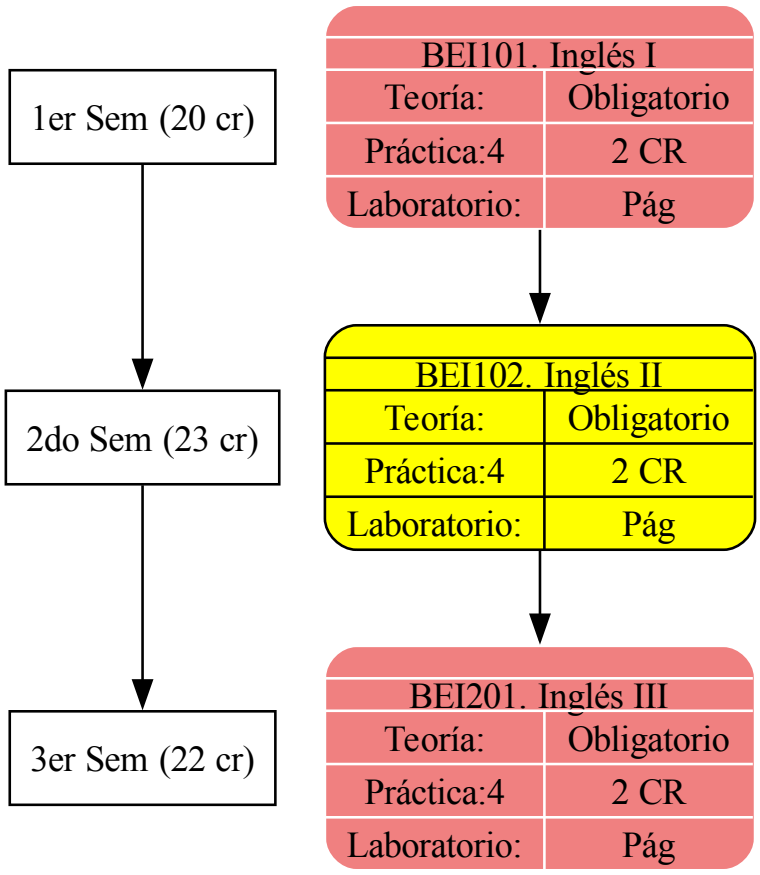


Figura 3.11: Mapa de Conexión. BEI102 Inglés II

2. Fundamentación

Sobre la base de las habilidades fundamentales del inglés, este curso desarrolla la capacidad del estudiante para comunicarse en contextos técnicos y científicos con mayor fluidez. Las estructuras gramaticales se practican mediante la lectura y producción de descripciones de sistemas tecnológicos, fenómenos científicos y procesos de ingeniería. Los estudiantes adquieren vocabulario relevante para la computación, los datos y las ciencias físicas.

3. Objetivos del curso.

- Desarrollar fluidez en inglés aplicado a la comunicación técnica y científica.
- Ampliar el vocabulario en computación, datos y dominios de ingeniería.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Familiarizarse)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Avances y Descubrimientos, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Repaso de Pasado Simple: Afirmativo y Negativo en contextos científicos. **II:** Expresiones de tiempo con 'ago': Cuándo se inventaron o lanzaron tecnologías. **III:** Conjunciones: and, so, but, because — estructuración de explicaciones técnicas. **IV:** Vocabulario: Hitos científicos, inventos y descubrimientos tecnológicos.

Objetivos de Aprendizaje I: Al finalizar la unidad, el estudiante narra avances tecnológicos pasados y conecta relaciones causa-efecto utilizando el pasado simple y conjunciones (e.g., "Scientists discovered X, so engineers developed Y").

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.2 Datos y Variables, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Sustantivos contables e incontables en computación: files, data, code, memory. **II:** Uso de some, any, a lot of para cuantificar recursos técnicos. **III:** Expresiones de cantidad: a little memory, a few errors, too many processes. **IV:** Vocabulario: Tipos de datos, variables, constantes y unidades de almacenamiento.

Objetivos de Aprendizaje I: El estudiante utiliza correctamente estructuras de sustantivos contables e incontables para describir datos, recursos y estados del sistema en contextos técnicos.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.3 El Futuro de la Tecnología, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Planes futuros: Verb + to + infinitive en la planificación de proyectos. **II:** Uso de 'Will' vs 'Going to' para predicciones y lanzamientos programados. **III:** Verbos auxiliares para pronosticar: tendencias de IA, computación en la nube, computación cuántica. **IV:** Vocabulario: Tecnologías emergentes y trayectorias profesionales en computación.

Objetivos de Aprendizaje I: El estudiante expresa intenciones tecnológicas y planes de proyecto, además de realizar predicciones basadas en evidencia sobre tendencias en computación e ingeniería.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.4 Comparando Tecnologías, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Adjetivos comparativos y superlativos aplicados a hardware y software. **II:** Comparación de lenguajes de programación, sistemas operativos y algoritmos. **III:** Sinónimos y antónimos en vocabulario técnico. **IV:** Lectura: Artículos cortos de comparación tecnológica e informes de benchmarks.

Objetivos de Aprendizaje I: El estudiante realiza comparaciones técnicas detalladas entre sistemas, lenguajes y herramientas, utilizando estructuras comparativas y superlativas con precisión.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.5 Procesos en Ejecución, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Presente Continuo para describir procesos técnicos activos y en curso. **II:** Uso actual: "The server is processing requests." **III:** Uso futuro con Presente Continuo: despliegues y experimentos programados. **IV:** Redacción: Actualizaciones de estado e informes de avance en proyectos técnicos.

Objetivos de Aprendizaje I: El estudiante describe procesos computacionales activos, experimentos en curso y actividades técnicas programadas utilizando la forma del presente continuo.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.6 Hitos del Software, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Introducción al Presente Perfecto para describir logros con relevancia actual. **II:** Palabras clave: ever, never, yet, just — aplicadas a versiones de software e investigación. **III:** Vocabulario: Historial de versiones, parches, actualizaciones, artículos publicados. **IV:** Lectura: Noticias tecnológicas y changelogs de software.

Objetivos de Aprendizaje I: El estudiante comunica logros técnicos e hitos de software con relevancia presente, sin especificar un tiempo exacto, utilizando el presente perfecto.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.7 Desafíos Técnicos, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Adjetivos terminados en -ed e -ing para describir experiencias y procesos técnicos. **II:** Verbos modales de obligación: must, have to — aplicados a requisitos de ingeniería. **III:** Vocabulario: Desafíos de ingeniería, restricciones, trade-offs y estándares. **IV:** Redacción: Planteamiento corto de un problema técnico.

Objetivos de Aprendizaje I: El estudiante describe los aspectos técnicos de los desafíos de ingeniería y expresa obligaciones y restricciones utilizando vocabulario especializado.

Bibliografía: [Soars and John, 2002, Cambridge, 2006, MacGrew, 1999]

5.8 Inglés para Tecnología II, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Terminología técnica: Dispositivos de entrada/salida, almacenamiento y periféricos. **II:** Lectura y seguimiento de instrucciones de instalación y configuración de software. **III:** Comunicación en soporte técnico: Descripción de síntomas y soluciones. **IV:** Presentación: Describir un sistema computacional a una audiencia no especializada.

Objetivos de Aprendizaje I: Al finalizar el curso, el estudiante interpreta instrucciones técnicas, describe componentes de hardware y comunica conceptos básicos de computación a audiencias diversas en inglés.

Bibliografía: [Soars and John, 2002, 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, 2002] Soars, L. and John (2002). *American Headway N 1 Student Book*. Editorial Oxford.

Tercer Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS113 Programación Orientada a Objetos II (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 80)

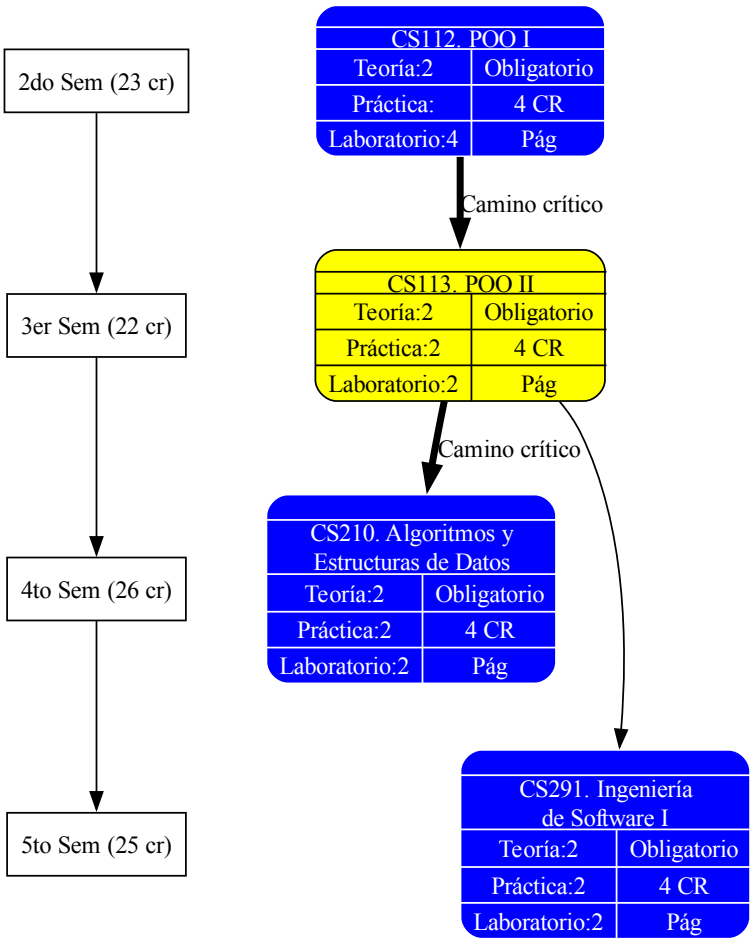


Figura 3.12: Mapa de Conexión. CS113 Programación Orientada a Objetos II

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Evaluar)

5. Contenido

5.1 Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión, 8 hr(s)

Competencias: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS2B1 Desarrollo Basado en Plataformas (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)

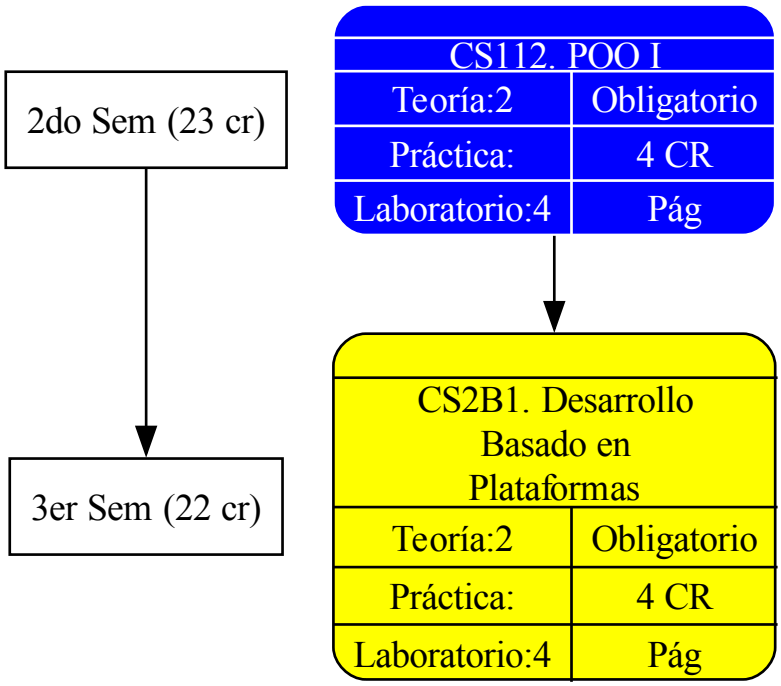


Figura 3.13: Mapa de Conexión. CS2B1 Desarrollo Basado en Plataformas

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*)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Aspectos Comunes: Plataformas, APIs y Restricciones, 5 hr(s)

Competencias: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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: AG-C04,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI161 IA Aplicada (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)

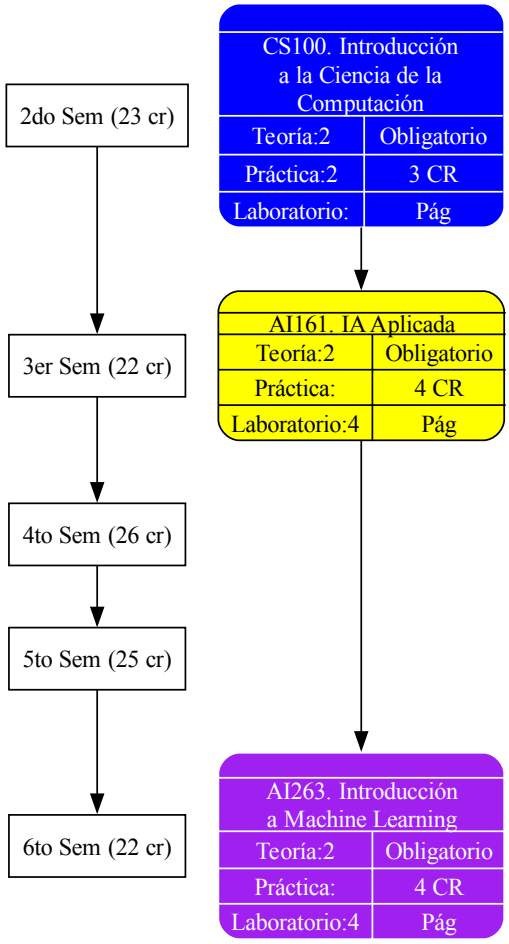


Figura 3.14: Mapa de Conexión. AI161 IA Aplicada

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*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Familiarizarse)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Fundamentos de IA y Panorama Global, 16 hr(s)

Competencias: AG-C01,AG-C11

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: AG-C01,AG-C11

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: AG-C01,AG-C11

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, Cloud, 2024b, Cloud, 2024a, OpenAI, 2024]

5.4 Ética Global en IA y Uso Responsable, 12 hr(s)

Competencias: AG-C01,AG-C11

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, European Union, 2024, Bender et al., 2021, Chinese Academy of Cyberspace Studies, 2024, Tencent AI Lab, 2023]

6. Bibliografía

[Bender et al., 2021] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pages 610–623. Association for Computing Machinery. Foundational paper on AI ethics with ongoing relevance.

[Chinese Academy of Cyberspace Studies, 2024] Chinese Academy of Cyberspace Studies (2024). Ai governance and ethics in china. Technical report, Cyberspace Administration of China. Official Chinese perspective on AI governance and ethical standards.

[Cloud, 2024a] Cloud, A. (2024a). AI Ethics and best practices in the chinese context. Chinese perspective on AI ethics and implementation guidelines.

[Cloud, 2024b] Cloud, G. (2024b). Introduction to responsible ai. Google's updated framework for responsible AI development and use.

[DeepSeek, 2024] DeepSeek (2024). Deepseek model documentation and best practices. Official documentation for DeepSeek AI models and usage guidelines.

[European Union, 2024] European Union (2024). AI Act: Regulatory framework. Technical report, European Commission. Comprehensive AI regulation framework effective in 2024.

[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*, 101(5):42–51. 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*. Portfolio. Practical guide for human-AI collaboration.

[Ng, 2019] Ng, A. (2019). AI for everyone. Technical report, DeepLearning.AI. Continuously updated online course, available on Coursera.

[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 the chinese technology industry. Technical report, Tencent AI Lab. Industry perspective on AI ethics from a leading Chinese technology 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.

[White et al., 2023] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. Peer-reviewed research on prompt engineering patterns.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

STA251 Estadística y Probabilidades (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - BMA102 Cálculo Diferencial (1er Sem-Pág 57)

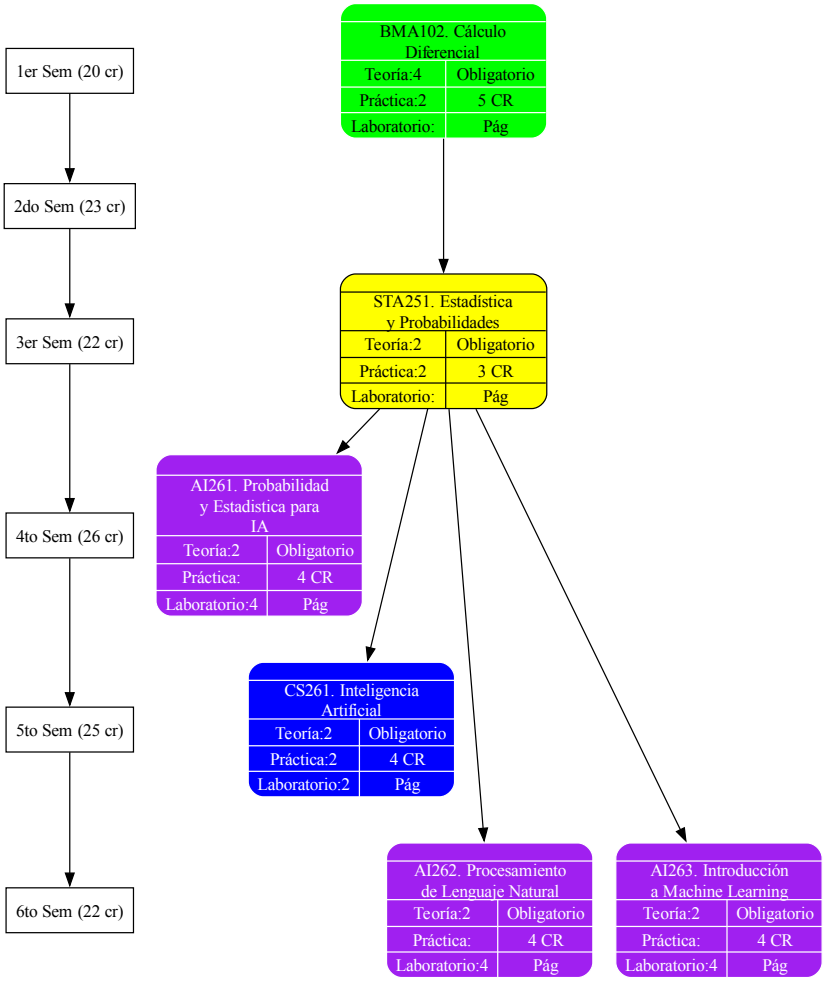


Figura 3.15: Mapa de Conexión. STA251 Estadística y Probabilidades

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*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

5. Contenido

5.1 Resúmenes Numéricos: Posición, Dispersión y Forma, 4 hr(s)

Competencias: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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: AG-C07

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

FG106 Teatro (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

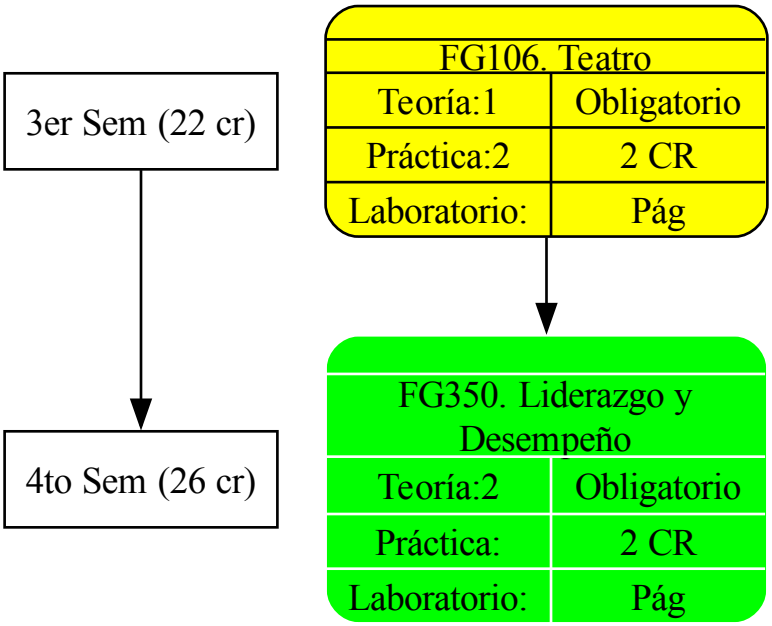


Figura 3.16: Mapa de Conexión. FG106 Teatro

2. Fundamentación

El teatro como arte escénica integral invita al estudiante a reconocerse como ser humano creativo y social. A través de la práctica teatral, el estudiante desarrolla su expresión corporal, verbal y emocional; fortalece su confianza y seguridad personal; y cultiva habilidades de comunicación, empatía y trabajo en equipo. Este curso proporciona herramientas prácticas de expresión artística que contribuyen a la formación integral del estudiante universitario, estimulando su sensibilidad, creatividad e imaginación. Al mismo tiempo, lo integra a su comunidad académica a través de experiencias vivenciales compartidas, abriendo caminos de autoconocimiento y de comunicación auténtica con los demás.

3. Objetivos del curso.

- Desarrollar la conciencia corporal y las habilidades comunicativas no verbales del estudiante mediante técnicas de expresión teatral.
- Fomentar la creatividad, la espontaneidad y el trabajo en equipo a través del juego dramático y la improvisación.

- Conocer el recorrido histórico del teatro desde sus orígenes hasta sus manifestaciones contemporáneas, incluyendo el teatro peruano.
- Valorar el arte teatral como herramienta de integración social, autoconocimiento y expresión de la identidad personal y colectiva.
- Aplicar las técnicas teatrales aprendidas en la creación y presentación pública de una propuesta escénica grupal.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 El Arte, la Creatividad y el Teatro, 6 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** ¿Qué es el Arte? Una experiencia vivencial y personal. **II:** La llave maestra: la creatividad. **III:** La importancia del teatro en la formación personal y profesional. **IV:** Utilidad y enfoque del arte teatral. **V:** El teatro como herramienta pedagógica y de desarrollo personal.

Objetivos de Aprendizaje **I:** Reconocer la vigencia del arte y la creatividad en el desarrollo personal y social [Usar]. **II:** Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usar]. **III:** Identificar nociones básicas del teatro y su función en la vida universitaria [Usar].

Bibliografía: [Majorana, 1958, Pavis, 1998, Miravalles, 1998]

5.2 El Juego: el quehacer del actor, 6 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** Juego, luego existo. El juego como lenguaje universal. **II:** El juego del niño y el juego dramático. **III:** Juegos de integración grupal y juegos de creatividad. **IV:** La improvisación espontánea. **V:** La secuencia teatral y el calentamiento corporal.

Objetivos de Aprendizaje **I:** Reconocer el juego como herramienta fundamental del teatro [Usar]. **II:** Interiorizar y revalorar el juego como aprendizaje creativo [Usar]. **III:** Acercarse de manera espontánea y natural a la vivencia teatral [Usar].

Bibliografía: [Pavis, 1998, Miravalles, 1998]

5.3 La expresión corporal y el uso dramático del objeto, 9 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** Toma de conciencia del cuerpo: el cuerpo como instrumento. **II:** Toma de conciencia del espacio escénico. **III:** Toma de conciencia del tiempo y el ritmo. **IV:** Creación de secuencias individuales y colectivas: cuerpo, espacio y tiempo. **V:** El uso dramático del objeto: el juego teatral. **VI:** Presentaciones teatrales con el uso del objeto.

Objetivos de Aprendizaje **I:** Experimentar nuevas formas de expresión y comunicación corporal [Usar]. **II:** Desarrollar mecanismos de control y manejo del cuerpo en el espacio escénico [Usar]. **III:** Cultivar la imaginación y la capacidad de respuesta a estímulos auditivos, rítmicos y visuales [Usar]. **IV:** Explorar estados emocionales y climas colectivos a través del movimiento [Usar].

Bibliografía: [Pavis, 1998, Grotowski, 1970]

5.4 Comunicación No Verbal en el Teatro, 9 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Relajación, concentración y respiración. **II:** Desinhibición e interacción con el grupo. **III:** La improvisación: técnicas y ejercicios. **IV:** Equilibrio, peso, tiempo y ritmo corporal. **V:** Análisis del movimiento: tipos y calidades. **VI:** La presencia escénica y la energía del actor. **VII:** La danza y la coreografía teatral. **VIII:** El gesto expresivo y el uso de la máscara.

Objetivos de Aprendizaje I: Ejercitarse en el manejo de destrezas comunicativas no verbales [Usar]. **II:** Practicar juegos y ejercicios de lenguaje corporal individual y grupal [Usar]. **III:** Expresar libre y creativamente emociones y sentimientos a través de representaciones originales [Usar]. **IV:** Distinguir los diferentes tipos de actuación y sus fundamentos técnicos [Usar].

Bibliografía: [Pavis, 1998, Stanislavski, 1936, Grotowski, 1970]

5.5 Huellas del Teatro en el Tiempo, 6 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: El origen del teatro: rito y representación en Grecia y Roma. **II:** El teatro medieval y la Commedia dell'Arte. **III:** Teatro del Siglo de Oro español y teatro isabelino inglés. **IV:** Romanticismo, Naturalismo y Realismo: Ibsen y Chéjov. **V:** El teatro épico de Brecht y el teatro psicológico de Stanislavski. **VI:** El teatro del absurdo, el teatro contemporáneo y el teatro total. **VII:** Teatro en el Perú: Yuyachkani, La Tarumba, Pataclaun y otros.

Objetivos de Aprendizaje I: Conocer la influencia que la sociedad ha ejercido en el teatro a lo largo de la historia [Usar]. **II:** Apreciar el valor y el aporte de dramaturgos y directores teatrales fundamentales [Usar]. **III:** Analizar el contexto social e histórico del arte teatral en distintas épocas [Usar]. **IV:** Reflexionar sobre el teatro peruano y sus expresiones contemporáneas [Usar].

Bibliografía: [Pavis, 1998, Campbell, 2001]

5.6 El Montaje Teatral, 12 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Apreciación teatral: asistencia y análisis crítico de obras. **II:** El espacio escénico: tipos de escenario y su uso dramático. **III:** El texto dramático: lectura, análisis y adaptación. **IV:** Construcción del personaje: técnica y psicología. **V:** Dirección y puesta en escena: fundamentos básicos. **VI:** Elementos de producción: vestuario, maquillaje, escenografía y utilería. **VII:** Presentación pública de ejercicios teatrales en grupo.

Objetivos de Aprendizaje I: Emplear la creación teatral como manifestación de ideas y sentimientos ante la sociedad [Usar]. **II:** Aplicar las técnicas practicadas en una propuesta escénica concreta y significativa [Usar]. **III:** Intercambiar experiencias y realizar presentaciones teatrales breves en grupo frente al público [Usar].

Bibliografía: [Pavis, 1998, Miravalles, 1998, Brook, 1968, Campbell, 2001]

6. Bibliografía

[Brook, 1968] Brook, P. (1968). *The Empty Space: A Book About the Theatre*. Penguin Books.

[Campbell, 2001] Campbell, A. B. (2001). *Historia general del teatro en el Perú*. Universidad de San Martín de Porres.

[Grotowski, 1970] Grotowski, J. (1970). *Towards a Poor Theatre*. Simon and Schuster.

[Majorana, 1958] Majorana, A. (1958). *El arte de hablar en público*. La España Moderna.

[Miravalles, 1998] Miravalles, L. (1998). *Iniciación al teatro: teoría y práctica*. San Pablo.

[Pavis, 1998] Pavis, P. (1998). *Dictionary of the Theatre: Terms, Concepts, and Analysis*. University of Toronto Press.

[Stanislavski, 1936] Stanislavski, C. (1936). *An Actor Prepares*. Theatre Arts Books.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BEI201 Inglés III (Obligatorio)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 2
- **Horas del curso**: **Práctica**: 4 horas;
- **Prerrequisitos**:
  - BEI102 Inglés II (2do Sem-Pág 95)

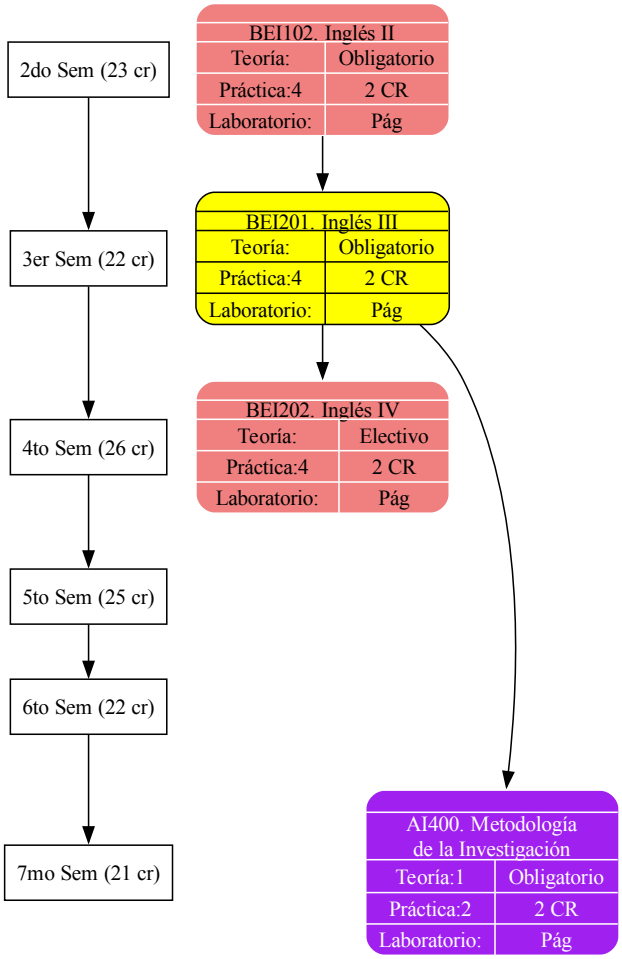


Figura 3.17: Mapa de Conexión. BEI201 Inglés III

2. Fundamentación

En este nivel, los estudiantes utilizan el inglés como medio principal para acceder y producir contenido técnico en computación, ingeniería y ciencias. Las estructuras gramaticales se aplican a la lectura de artículos científicos, la redacción de informes técnicos, la descripción de procesos de ingeniería y la participación en discusiones profesionales. El curso conecta la fluidez cotidiana en inglés con las exigencias de la comunicación técnica especializada.

3. Objetivos del curso.

- Capacitar a los estudiantes para leer, comprender y producir comunicación técnica en inglés.
- Proporcionar herramientas lingüísticas para conectar ideas en la escritura descriptiva y analítica sobre computación e ingeniería.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Familiarizarse)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Comunicación Técnica, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Repaso de Tiempos: Presente, Pasado y Futuro en documentación técnica. **II:** Oraciones interrogativas con Wh- questions complejas para el análisis de sistemas. **III:** Palabras con más de un significado en contextos técnicos (Polisemia). **IV:** Expresiones para describir actividades profesionales e intereses de investigación.

Objetivos de Aprendizaje **I:** Al finalizar la unidad, el estudiante utiliza diversos tiempos verbales para intercambiar información técnica detallada y formular preguntas precisas sobre sistemas computacionales y procesos de ingeniería.

Bibliografía: [? line:46 , ? line:46 , ? line:46]

5.2 Comparando Arquitecturas, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Uso de "What's it like?": Descripción de características de sistemas e interfaces. **II:** Adjetivos comparativos y superlativos avanzados en análisis de benchmarks. **III:** Colocaciones (collocations) en computación: high-performance, low-level, fault-tolerant. **IV:** Lectura: Informes de comparación de arquitecturas (RISC vs. CISC, SQL vs. NoSQL).

Objetivos de Aprendizaje **I:** El estudiante analiza y describe diferencias entre arquitecturas, marcos de trabajo y sistemas computacionales utilizando estructuras comparativas variadas con precisión técnica.

Bibliografía: [? line:59 , ? line:59 , ? line:59]

5.3 Antes y Ahora en la Computación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Presente Perfecto vs. Pasado Simple: Contraste entre eventos históricos y estado actual. **II:** Expresiones temporales: for, since, ever, never aplicadas a la evolución tecnológica. **III:** Adverbios de tiempo y modo en la narrativa científica. **IV:** Vocabulario: Generaciones de tecnología, sistemas heredados (legacy), plataformas actuales.

Objetivos de Aprendizaje **I:** El estudiante diferencia y aplica correctamente el presente perfecto y el pasado simple para narrar la evolución de las tecnologías computacionales y contrastar logros históricos con el estado actual del campo.

Bibliografía: [? line:72 , ? line:72 , ? line:72]

5.4 Estándares y Protocolos de Ingeniería, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Modales de obligación y permiso: Have to, Must, Can en normas profesionales. **II:** Recomendaciones técnicas: Should, Ought to — revisiones de código y diseño. **III:** Vocabulario: Estándares IEEE/ACM, convenciones de codificación, políticas de seguridad. **IV:** Redacción formal: Memorandos técnicos y recomendaciones de ingeniería.

Objetivos de Aprendizaje **I:** El estudiante expresa normas de ingeniería, estándares y recomendaciones profesionales utilizando verbos modales apropiados en escritura técnica formal y discusión entre pares.

Bibliografía: [? line:85, ? line:85, ? line:85]

5.5 ¿Cómo se construye el Software?, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Voz Pasiva: Presente y Pasado Simple para describir procesos de ingeniería. **II:** Vocabulario: Fabricación, despliegue, integración y pruebas. **III:** Expresiones para dar opiniones técnicas y discutir trade-offs de diseño. **IV:** Redacción: Descripción de proceso para un sistema de software o hardware.

Objetivos de Aprendizaje **I:** El estudiante utiliza la voz pasiva para describir cómo se diseñan, construyen, prueban y despliegan los sistemas, participando en discusiones técnicas sobre procesos de ingeniería.

Bibliografía: [? line:98, ? line:98, ? line:98]

5.6 Depuración y Análisis, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Pasado Continuo vs. Pasado Simple en reportes de incidentes y fallos. **II:** Conectores de narrativa y secuencia: while, when, suddenly, as a result. **III:** Vocabulario: Bugs, excepciones, crashes, logs, análisis de causa raíz. **IV:** Redacción: Informe de incidente de un fallo de software o de red.

Objetivos de Aprendizaje **I:** El estudiante narra sesiones de depuración, fallos de sistemas e incidentes experimentales utilizando tiempos narrativos y conectores de secuencia técnicos con fluidez.

Bibliografía: [? line:111, ? line:111, ? line:111]

5.7 Investigación e Innovación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Formas de Futuro: Will, Going to, Presente Continuo para proyección de investigación. **II:** Expresiones de probabilidad y certeza en contextos científicos. **III:** Vocabulario: Ciclos de innovación, propuestas de investigación, prospectiva tecnológica. **IV:** Presentaciones orales: .[El] futuro de la computación en mi campo".

Objetivos de Aprendizaje **I:** El estudiante proyecta metas de investigación y discute direcciones futuras en computación e ingeniería utilizando diversas estructuras de futuro con grados apropiados de certeza.

Bibliografía: [? line:124, ? line:124, ? line:124]

5.8 Inglés Técnico III, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Terminología de Programación y Desarrollo Web: funciones, APIs, frameworks. **II:** Descripciones de Interfaz de Usuario (UI/UX) y conceptos de usabilidad. **III:** Vocabulario básico de gestión de proyectos: sprints, milestones, deliverables. **IV:** Redacción: Especificación técnica simple o documento README.

Objetivos de Aprendizaje I: Al finalizar el curso, el estudiante utiliza vocabulario especializado de desarrollo de software para describir componentes lógicos y funcionales, y produce documentación técnica básica en inglés.

Bibliografía: [**?** line:137 , **?** line:137 , **?** line:137]

6. Bibliografía

line:142 line:142 line:142

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

EC101 Economía General (Electivo)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

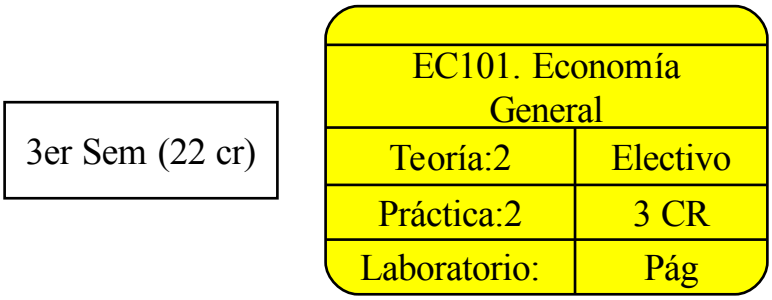


Figura 3.18: Mapa de Conexión. EC101 Economía General

2. Fundamentación

La economía es el estudio de cómo las sociedades asignan recursos escasos. Para los estudiantes de ciencias de la computación, la comprensión de los principios económicos es valiosa para la toma de decisiones en contextos empresariales, la evaluación de proyectos tecnológicos y la comprensión del impacto económico de las innovaciones. Este curso proporciona una introducción a los principios microeconómicos y macroeconómicos.

3. Objetivos del curso.

- Comprender los principios fundamentales de la microeconomía y la macroeconomía.
- Analizar la oferta, la demanda y la formación de precios en los mercados.
- Evaluar políticas económicas y su impacto en la sociedad.

4. Competencias Generales (*Outcomes*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Familiarizarse)

AG-C06) Aprendizaje a lo largo de la vida: Reconoce la importancia del aprendizaje continuo y la adaptación a nuevas tecnologías. (Familiarizarse)

5. Contenido

5.1 Introducción a la Economía, 6 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: **I:** "¿Qué es la economía?" **II:** Principios básicos de la economía. **III:** Microeconomía vs. Macroeconomía. **IV:** Pensamiento económico.

Objetivos de Aprendizaje I: Definir los conceptos básicos de la economía. [Familiarizarse] **II:** Diferenciar entre microeconomía y macroeconomía. [Usar] **III:** Aplicar el pensamiento económico para analizar situaciones cotidianas. [Evaluar]

Bibliografía: [Mankiw, 2018, Samuelson and Nordhaus, 2010]

5.2 Microeconomía: Oferta y Demanda, 8 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: I: La curva de demanda. **II:** La curva de oferta. **III:** Equilibrio de mercado. **IV:** Elasticidad.

Objetivos de Aprendizaje I: Explicar la ley de la demanda y la oferta. [Familiarizarse] **II:** Determinar el equilibrio de mercado. [Usar] **III:** Calcular e interpretar la elasticidad precio de la demanda. [Evaluar]

Bibliografía: [Mankiw, 2018, Samuelson and Nordhaus, 2010]

5.3 Microeconomía: Empresas y Mercados, 10 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: I: Costos de producción. **II:** Competencia perfecta. **III:** Monopolio. **IV:** Oligopolio.

Objetivos de Aprendizaje I: Analizar los costos de producción de una empresa. [Familiarizarse] **II:** Describir las características de los diferentes tipos de mercado. [Usar] **III:** Comparar y contrastar la competencia perfecta con el monopolio y el oligopolio. [Evaluar]

Bibliografía: [Mankiw, 2018, Samuelson and Nordhaus, 2010]

5.4 Macroeconomía: Medición de la Actividad Económica, 8 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: I: El producto interno bruto (PIB). **II:** Inflación y desempleo.

Objetivos de Aprendizaje I: Definir y calcular el PIB. [Familiarizarse] **II:** Explicar cómo se miden la inflación y el desempleo. [Usar] **III:** Analizar la relación entre el PIB, la inflación y el desempleo. [Evaluar]

Bibliografía: [Mankiw, 2018, Samuelson and Nordhaus, 2010]

5.5 Macroeconomía: Política Monetaria y Fiscal, 8 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: I: Política monetaria. **II:** Política fiscal.

Objetivos de Aprendizaje I: Explicar cómo funciona la política monetaria. [Familiarizarse] **II:** Describir los instrumentos de la política fiscal. [Usar] **III:** Analizar el impacto de las políticas monetarias y fiscales en la economía. [Evaluar]

Bibliografía: [Mankiw, 2018, Samuelson and Nordhaus, 2010]

5.6 Economía de la Innovación y la Tecnología, 8 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: I: El papel de la innovación en el crecimiento económico. **II:** Impacto económico de las nuevas tecnologías. **III:** Mercados digitales.

Objetivos de Aprendizaje I: Explicar cómo la innovación impulsa el crecimiento económico. [Familiarizarse] **II:** Analizar el impacto económico de las nuevas tecnologías. [Usar] **III:** Comprender las características de los mercados digitales. [Evaluar]

Bibliografía: [Samuelson and Nordhaus, 2010]

6. Bibliografía

[Mankiw, 2018] Mankiw, N. G. (2018). *Principles of Economics*. Cengage Learning.

[Samuelson and Nordhaus, 2010] Samuelson, P. A. and Nordhaus, W. D. (2010). *Economics*. McGraw-Hill Education.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

FG120 Realidad Nacional (Electivo)

1. Información general

- **Semestre**: 3er Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

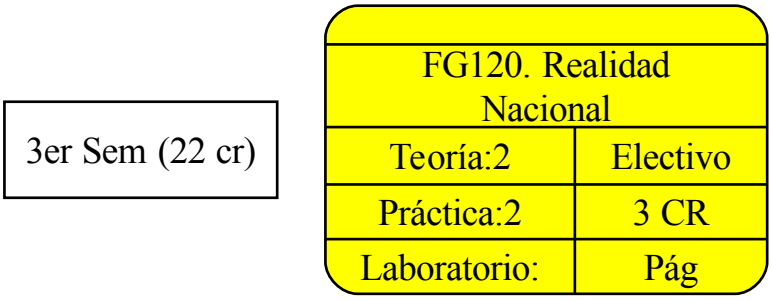


Figura 3.19: Mapa de Conexión. FG120 Realidad Nacional

2. Fundamentación

Comprender la realidad nacional peruana es esencial para los futuros profesionales, incluyendo aquellos en ciencias de la computación, para que puedan contribuir al desarrollo del país de manera informada y responsable. Este curso analiza los aspectos históricos, sociales, económicos y políticos del Perú, con énfasis en los desafíos y oportunidades presentes.

3. Objetivos del curso.

- Analizar los procesos históricos que han configurado la realidad nacional peruana.
- Comprender los principales desafíos sociales, económicos y políticos que enfrenta el Perú.
- Identificar las oportunidades de desarrollo y el rol de la tecnología en la construcción de un futuro mejor para el país.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Historia del Perú, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** Perú prehispánico. **II:** Época colonial. **III:** El Perú republicano: siglo XIX y XX.

Objetivos de Aprendizaje I: Describir los principales periodos de la historia peruana. [Familiarizarse] **II:** Analizar los eventos clave que han moldeado la identidad nacional. [Usar] **III:** Explicar la influencia del pasado en la realidad actual del Perú. [Evaluar]

Bibliografía: [Klarén, 2012, Cotler, 2005]

5.2 Diversidad Cultural y Social, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Diversidad cultural del Perú. **II:** Grupos étnicos y lingüísticos. **III:** Desigualdad social y exclusión.

Objetivos de Aprendizaje I: Identificar los diferentes grupos culturales que conforman el Perú. [Familiarizarse] **II:** Analizar las causas y consecuencias de la desigualdad social. [Usar] **III:** Proponer soluciones para promover la inclusión social. [Evaluar]

Bibliografía: [Portocarrero, 2001, Manrique, 1988]

5.3 Economía Peruana, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Estructura económica del Perú. **II:** Sectores económicos: minería, agricultura, industria, servicios. **III:** Comercio internacional. **IV:** Desafíos económicos y desarrollo sostenible.

Objetivos de Aprendizaje I: Describir la estructura económica del Perú. [Familiarizarse] **II:** Analizar la importancia de los diferentes sectores económicos. [Usar] **III:** Evaluar las políticas económicas y su impacto en el desarrollo sostenible. [Evaluar]

Bibliografía: [Óscar Dancourt, 2013, Hunt, 2018]

5.4 Sistema Político Peruano, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Organización del Estado peruano. **II:** Democracia y gobernabilidad. **III:** Corrupción y transparencia.

Objetivos de Aprendizaje I: Describir la organización del Estado peruano. [Familiarizarse] **II:** Analizar los desafíos para la democracia y la gobernabilidad en el Perú. [Usar] **III:** Evaluar la importancia de la transparencia y la lucha contra la corrupción. [Evaluar]

Bibliografía: [Tanaka, 2018, Quiroz, 2013]

5.5 Tecnología y Desarrollo en el Perú, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: El rol de la tecnología en el desarrollo del Perú. **II:** Brecha digital e inclusión digital. **III:** Innovación y emprendimiento tecnológico.

Objetivos de Aprendizaje I: Analizar la importancia de la tecnología para el desarrollo económico y social del Perú. [Familiarizarse] **II:** Proponer soluciones para reducir la brecha digital. [Usar] **III:** Evaluar el potencial de la innovación y el emprendimiento tecnológico en el Perú. [Evaluar]

Bibliografía: [de Economía y Finanzas Perú, 2019]

5.6 Desafíos y Oportunidades para el Futuro, 8 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Desarrollo sostenible. **II:** Visión del Perú al 2050. **III:** El rol de los profesionales en la construcción del futuro.

Objetivos de Aprendizaje I: Analizar los desafíos que enfrenta el Perú para lograr un desarrollo sostenible. [Familiarizarse] **II:** Comprender la visión a largo plazo para el desarrollo del país. [Usar] **III:** Reflexionar sobre el rol de los profesionales en la construcción de un futuro mejor para el Perú. [Evaluar]

Bibliografía: [CEPLAN, 2021]

6. Bibliografía

[CEPLAN, 2021] CEPLAN (2021). Bases del plan estratégico de desarrollo nacional: Perú al 2050.

[Cotler, 2005] Cotler, J. (2005). *Clases, Estado y Nación en el Perú*. IEP.

[de Economía y Finanzas Perú, 2019] de Economía y Finanzas Perú, M. (2019). *Plan Nacional de Competitividad y Productividad 2019-2030*.

[Hunt, 2018] Hunt, S. (2018). *El Perú en el siglo XXI*. IEP.

[Klarén, 2012] Klarén, P. F. (2012). *Nación y sociedad en la historia del Perú*. IEP.

[Manrique, 1988] Manrique, N. (1988). *Yawar Fiesta: una ceremonia de sangre*. DESCO.

[Portocarrero, 2001] Portocarrero, G. (2001). *Rostros criollos del Perú: cultura, sociedad, política*. PUCP.

[Quiroz, 2013] Quiroz, A. W. (2013). *Historia de la corrupción en el Perú*. IEP.

[Tanaka, 2018] Tanaka, M. (2018). *Democracia y política en el Perú*. IEP.

[Óscar Dancourt, 2013] Óscar Dancourt (2013). *La economía peruana: siglos XIX y XX*. PUCP.

Cuarto Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

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 99)

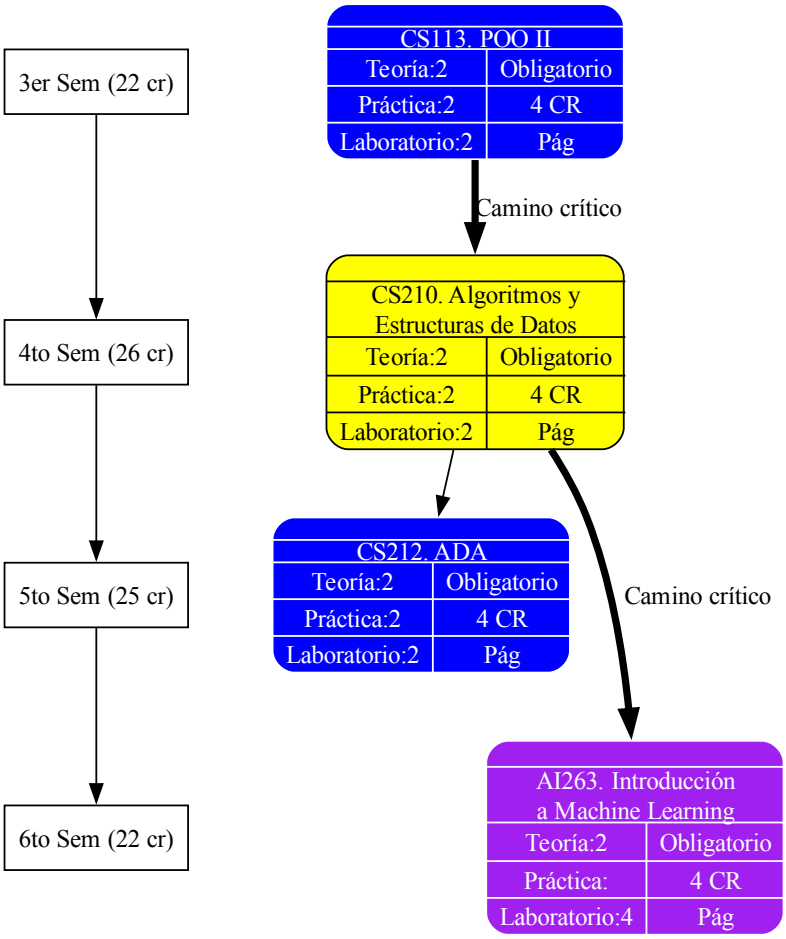


Figura 3.20: Mapa de Conexión. CS210 Algoritmos y Estructuras de Datos

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*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Fundamentos avanzados para estructuras de datos, 8 hr(s)

Competencias: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS211 Teoría de la Computación (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS1D1 Estructuras Discretas (2do Sem-Pág 85)

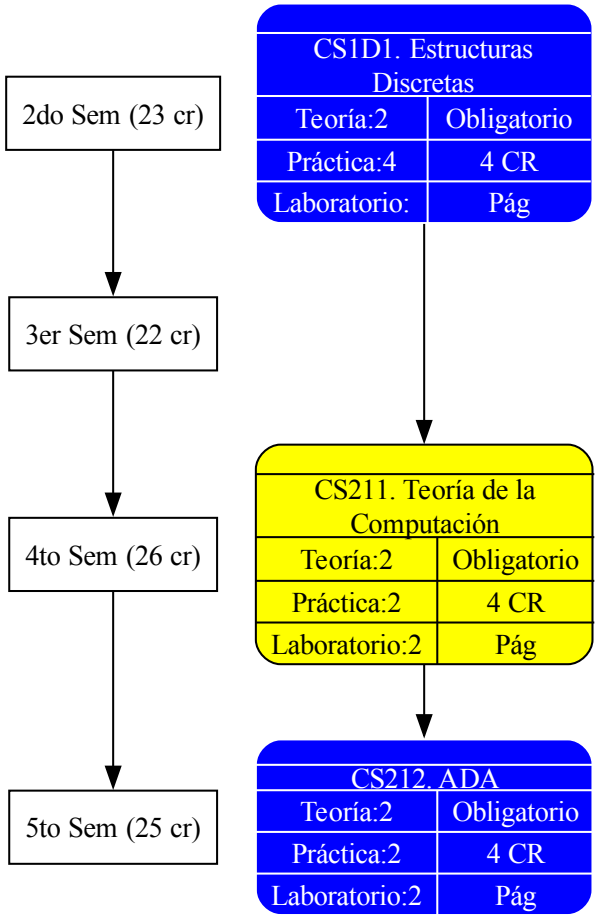


Figura 3.21: Mapa de Conexión. CS211 Teoría de la Computación

2. Fundamentación

La Teoría de la Computación proporciona los fundamentos matemáticos para entender qué puede ser computado y con qué eficiencia. Introduce lenguajes formales, autómatas y los límites de la resolubilidad algorítmica. Este conocimiento es fundamental para comprender la construcción de compiladores, la verificación formal y la complejidad inherente de los problemas computacionales.

3. Objetivos del curso.

- Dominar los conceptos de autómatas finitos y expresiones regulares.

- Analizar gramáticas libres de contexto y autómatas de pila.
- Comprender la Máquina de Turing Universal como modelo de computación.
- Comprender los límites de la computabilidad y el Problema de la Parada.
- Diferenciar entre clases de complejidad como P y NP.

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Autómatas Finitos y Lenguajes Regulares, 18 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Autómatas formales: *a*) Estado Finito *b*) Pila *c*) Linealmente Acotado *d*) Máquina de Turing **II:** Lenguajes formales, gramáticas y Jerarquía de Chomsky: *a*) Regulares (Tipo-3): 1) Expresiones Regulares *b*) Libres de Contexto (Tipo-2) *c*) Sensibles al Contexto (Tipo-1) *d*) Recursivamente Enumerables (Tipo-0) **III:** Autómatas deterministas y no deterministas **IV:** Demostraciones del Lema de Bombeo: *a*) Demostración de la limitación de Estado Finito/Lenguaje Regular *b*) Limitación de Autómatas de Pila/Lenguaje Libre de Contexto

Objetivos de Aprendizaje I: 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] **II:** Dado un problema, desarrollar un autómata apropiado que aborde el problema [Crear] **III:** Desarrollar una expresión regular para un lenguaje regular dado expresado en lenguaje natural [Crear] **IV:** Convertir entre notaciones equivalentemente poderosas para un lenguaje, incluyendo entre DFAs, NFAs y expresiones regulares, y entre PDAs y CFGs [Aplicar] **V:** Aplicar lemas de bombeo, o medios alternativos, para demostrar las limitaciones de los autómatas de Estado Finito y Pila [Aplicar]

Bibliografía: [Sipser, 2012, Hopcroft et al., 2013]

5.2 Lenguajes Libres de Contexto y PDA, 16 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Autómatas formales: *a*) Estado Finito *b*) Pila *c*) Linealmente Acotado *d*) Máquina de Turing **II:** Lenguajes formales, gramáticas y Jerarquía de Chomsky: *a*) Regulares (Tipo-3): 1) Expresiones Regulares *b*) Libres de Contexto (Tipo-2) *c*) Sensibles al Contexto (Tipo-1) *d*) Recursivamente Enumerables (Tipo-0) **III:** Formas Normales: Chomsky y Greibach **IV:** Demostraciones del Lema de Bombeo: *a*) Demostración de la limitación de Estado Finito/Lenguaje Regular *b*) Limitación de Autómatas de Pila/Lenguaje Libre de Contexto **V:** Relaciones entre autómatas formales, lenguajes y gramáticas

Objetivos de Aprendizaje I: 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] **II:** Dado un problema, desarrollar un autómata apropiado que aborde el problema [Crear] **III:** Convertir entre notaciones equivalentemente poderosas para un lenguaje, incluyendo entre DFAs, NFAs y expresiones regulares, y entre PDAs y CFGs [Aplicar] **IV:** Aplicar lemas de bombeo, o medios alternativos, para demostrar las limitaciones de los autómatas

de Estado Finito y Pila [Aplicar] **V:** Construir GLCs para descripciones de lenguajes formales [Evaluar]

Bibliografía: [Sipser, 2012, Hopcroft et al., 2013]

5.3 Máquinas de Turing y Computabilidad, 16 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Autómatas formales: *a*) Estado Finito *b*) Pila *c*) Linealmente Acotado *d*) Máquina de Turing **II:** Lenguajes formales, gramáticas y Jerarquía de Chomsky: *a*) Regulares (Tipo-3): 1) Expresiones Regulares *b*) Libres de Contexto (Tipo-2) *c*) Sensibles al Contexto (Tipo-1) *d*) Recursivamente Enumerables (Tipo-0) **III:** Decidabilidad, (in)computabilidad y parada **IV:** La tesis de Church-Turing **V:** Reducibilidad y reducciones **VI:** Modelos equivalentes de computación algorítmica: *a*) Máquinas de Turing y Variaciones (por ejemplo, multicinta, no deterministas) *b*) Cálculo Lambda *c*) Funciones Mu-Recursivas

Objetivos de Aprendizaje I: 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] **II:** Explicar una Máquina de Turing universal y su operación [Explicar] **III:** 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] **IV:** Explicar la Tesis de Church-Turing y su importancia para la computación algorítmica [Explicar] **V:** Aplicar aritmetización y diagonalización para demostrar que el Problema de la Parada para Máquinas de Turing es Indecidible [Aplicar] **VI:** Dado un lenguaje indecidible conocido, aplicar una reducción por mapeo o historia computacional para demostrar que otro lenguaje es indecidible [Aplicar]

Bibliografía: [Sipser, 2012, Kozen, 2006]

5.4 Jerarquía de Chomsky y Modelos Equivalentes, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Lenguajes formales, gramáticas y Jerarquía de Chomsky: *a*) Regulares (Tipo-3): 1) Expresiones Regulares *b*) Libres de Contexto (Tipo-2) *c*) Sensibles al Contexto (Tipo-1) *d*) Recursivamente Enumerables (Tipo-0) **II:** Relaciones entre autómatas formales, lenguajes y gramáticas **III:** Modelos equivalentes de computación algorítmica: *a*) Máquinas de Turing y Variaciones (por ejemplo, multicinta, no deterministas) *b*) Cálculo Lambda *c*) Funciones Mu-Recursivas **IV:** Lenguajes Sensibles al Contexto (Tipo-1) y aplicaciones

Objetivos de Aprendizaje I: 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] **II:** Explicar las funciones Recursivas Primitivas y Generales (cero, sucesor, selección, recursión primitiva, composición y Mu), su importancia e implementaciones en Máquina de Turing [Explicar] **III:** Explicar cómo se realiza la computación en el Cálculo Lambda (por ejemplo, conversión alfa y reducción beta) [Explicar] **IV:** Explicar el teorema de Rice y su importancia [Explicar]

Bibliografía: [Sipser, 2012, Kozen, 2006]

5.5 Marco de Análisis de Complejidad, 3 hr(s)

Competencias: AG-C08,AG-C12

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

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:** Evaluar informalmente la clase de complejidad fundamental de algoritmos simples [Evaluar]

Bibliografía: [Sipser, 2012, Cormen et al., 2022]

5.6 Notación Asintótica y Clases de Complejidad, 4 hr(s)

Competencias: AG-C08,AG-C12

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:** 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: [Sipser, 2012, Cormen et al., 2022]

5.7 Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas, 7 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Notaciones Little-o, Little-Omega y Little Theta **II:** Análisis recursivo formal

Objetivos de Aprendizaje I: Usar relaciones de recurrencia para evaluar la complejidad de tiempo de algoritmos definidos recursivamente [Aplicar] **II:** Aplicar relaciones de recurrencia elementales usando una forma del Teorema Maestro [Aplicar]

Bibliografía: [Sipser, 2012, Cormen et al., 2022]

5.8 Teoría de la Complejidad Computacional, 6 hr(s)

Competencias:

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

Objetivos de Aprendizaje I: Explicar la importancia de la NP-Completitud [Explicar] **II:** Explicar cómo NP-Duro es una cota inferior y NP es una cota superior para la NP-Completitud [Explicar] **III:** Explicar ejemplos de problemas NP-completos [Explicar] **IV:** Explicar el Teorema de Cook-Levin y la NP-Completitud de SAT [Explicar] **V:** Explicar las clases P y NP [Explicar] **VI:** Demostrar que un problema es NP-Completo reduciendo un problema NP-C clásico conocido a él (por ejemplo, 3SAT y Clique) [Crear] **VII:** Explicar la clase P-espacio y su relación con la clase EXP [Explicar]

Bibliografía: [Sipser, 2012, Cormen et al., 2022]

6. Bibliografía

[Cormen et al., 2022] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to Algorithms*. MIT Press, 4th edition.

[Hopcroft et al., 2013] Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2013). *Introduction to Automata Theory, Languages, and Computation*. Pearson, 3rd edition.

[Kozen, 2006] Kozen, D. C. (2006). *Theory of Computation*. Springer.

[Sipser, 2012] Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS221 Arquitectura de Computadores (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS1D1 Estructuras Discretas (2do Sem-Pág 85)

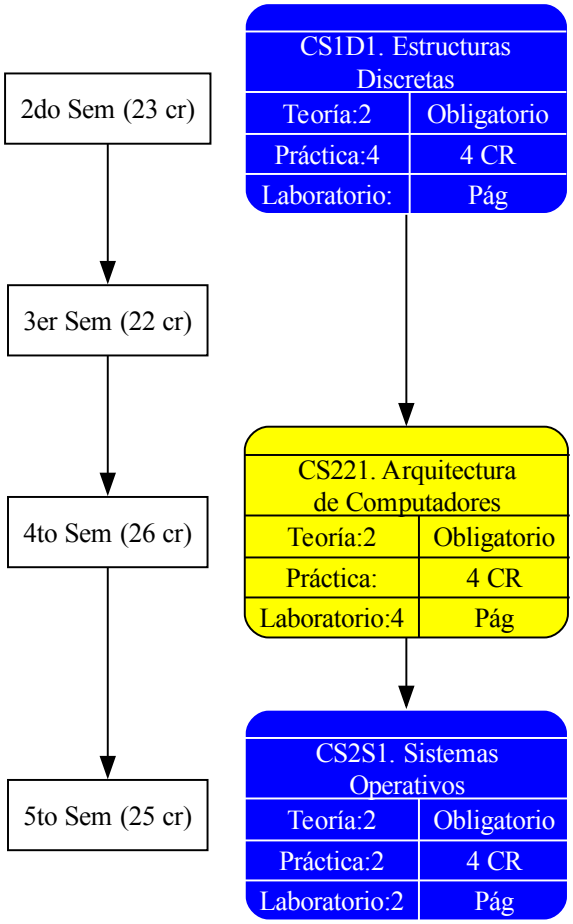


Figura 3.22: Mapa de Conexión. CS221 Arquitectura de Computadores

2. Fundamentación

Una comprensión profunda de la arquitectura de computadoras es esencial para desarrollar software de alto rendimiento y entender la interfaz hardware-software. Este curso explora cómo las computadoras ejecutan instrucciones, gestionan memoria y se comunican con dispositivos externos. Al estudiar la organización interna de los procesadores y la jerarquía de los sistemas de memoria, los estudiantes adquieren la capacidad de optimizar código para arquitecturas de hardware específicas y comprender las compensaciones en el diseño de sistemas modernos.

3. Objetivos del curso.

- Comprender los principios de la lógica digital y la representación de datos.
- Comprender la organización de la Unidad Central de Procesamiento (CPU).
- Dominar los conceptos básicos del lenguaje ensamblador y la arquitectura del conjunto de instrucciones.
- Analizar la jerarquía de memoria, incluyendo memoria caché y memoria virtual.
- Evaluar mejoras de rendimiento mediante segmentación y paralelismo.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Organización de Máquina a Nivel de Ensamblador, 20 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Arquitectura de máquina von Neumann **II:** Unidad de control: captación, decodificación y ejecución de instrucciones **III:** Introducción a SIMD vs MIMD y la taxonomía de Flynn **IV:** Organización de multiprocesadores/multinúcleo de memoria compartida **V:** Arquitectura del conjunto de instrucciones (ISA) (por ejemplo, x86, ARM y RISC-V): *a*) Conjuntos de instrucciones de ancho fijo vs variable *b*) Formatos de instrucción *c*) Manipulación de datos, control, E/S *d*) Modos de direccionamiento *e*) Programación en lenguaje de máquina *f*) Programación en lenguaje ensamblador **VI:** Mecanismos de llamada y retorno de subrutinas **VII:** E/S e interrupciones **VIII:** Segmentos de montículo, estáticos, pila y código

Objetivos de Aprendizaje **I:** Discutir cómo las unidades funcionales clásicas de von Neumann se implementan en sistemas embebidos, particularmente en memoria dentro y fuera del chip [Debatir] **II:** Describir cómo se ejecutan las instrucciones en una máquina von Neumann clásica, con extensiones para hilos, sincronización de multiprocesadores y ejecución SIMD [Describir] **III:** 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)] **IV:** Discutir cómo se representan las instrucciones a nivel de máquina y en el contexto de un ensamblador simbólico [Debatir] **V:** Mapear un ejemplo de patrones de lenguaje de alto nivel a notaciones de lenguaje ensamblador/máquina [Mapear/Mapa conceptual] **VI:** Contrastar diferentes formatos de instrucción considerando aspectos como direcciones por instrucción y formatos de longitud variable vs longitud fija [Contrastar] **VII:** Analizar un diagrama de subrutina para comentar cómo se manejan las llamadas de subrutina a nivel de ensamblador [Analizar] **VIII:** Describir conceptos básicos de interrupciones y operaciones de E/S [Describir] **IX:** Escribir un programa simple en lenguaje ensamblador para procesamiento y manipulación de cadenas/arreglos [Escribir]

Bibliografía: [Patterson and Hennessy, 2020, Harris and Harris, 2012]

5.2 Jerarquía de Memoria, 18 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Jerarquía de memoria: la importancia de la localidad temporal y espacial **II:** Organización y operaciones de memoria principal **III:** Memoria persistente (por ejemplo, SSD, discos estándar) **IV:** Latencia, tiempo de ciclo, ancho de banda e intercalado **V:** Memorias caché: *a*) Mapeo de direcciones *b*) Tamaño de bloque *c*) Política de reemplazo y almacenamiento *d*) Prebúsqueda **VI:** Coherencia de caché multiprocesador **VII:** Memoria virtual (soporte de hardware) **VIII:** Manejo de fallas y confiabilidad **IX:** Confiabilidad: *a*) Codificación de error *b*) Compresión de datos *c*) Integridad de datos **X:** Procesamiento en Memoria (PIM)

Objetivos de Aprendizaje I: 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] **II:** Medir el efecto de la latencia de memoria en el tiempo de ejecución [Evaluar] **III:** Enumerar las funciones de un sistema con gestión de memoria virtual [Enumerar] **IV:** 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]

Bibliografía: [Patterson and Hennessy, 2020, Stallings, 2015]

5.3 Interconexión y Comunicación, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Fundamentos de E/S: *a*) Protocolo de enlace y almacenamiento en búfer *b*) E/S programada *c*) E/S basada en interrupciones **II:** Estructuras de interrupción: vectorizadas y priorizadas, acuse de recibo de interrupción **III:** Dispositivos de E/S (por ejemplo, ratón, teclado, pantalla, cámara, sensores, actuadores) **IV:** Almacenamiento externo, organización física y unidades **V:** Fundamentos de buses: *a*) Protocolos de bus *b*) Arbitraje *c*) Acceso directo a memoria (DMA)

Objetivos de Aprendizaje I: 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] **II:** Enumerar varios tipos de buses en un sistema informático [Enumerar] **III:** Enumerar las ventajas de los discos magnéticos y contrastarlas con las de los discos de estado sólido [Listar/Enumerar]

Bibliografía: [Patterson and Hennessy, 2020, Tanenbaum and Austin, 2012]

5.4 Organización Funcional, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Implementación de rutas de datos simples, incluyendo segmentación de instrucciones, detección de riesgos y resolución (por ejemplo, paradas, reenvío) **II:** Unidad de control: *a*) Implementación cableada *b*) Realización microprogramada **III:** Segmentación de instrucciones **IV:** Introducción al paralelismo a nivel de instrucción (ILP)

Objetivos de Aprendizaje I: Comparar implementaciones alternativas de rutas de datos en arquitecturas de computadoras modernas [Comparar] **II:** Producir un conjunto de señales de control para sumar dos enteros usando implementaciones cableadas y microprogramadas [Producir] **III:** Discutir el paralelismo a nivel de instrucción usando segmentación y los riesgos significativos que pueden ocurrir [Debatir] **IV:** Diseñar un procesador completo, incluyendo ruta de datos y control [Diseñar] **V:** Calcular los ciclos promedio por instrucción para una implementación dada de procesador y sistema de memoria [Computar/Calcular]

Bibliografía: [Harris and Harris, 2012, Patterson and Hennessy, 2020]

5.5 Arquitecturas Heterogéneas, 10 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Arquitecturas SIMD y MIMD (por ejemplo, GPUs de Propósito General, TPUs y NPUs) **II:** Sistemas de memoria heterogéneos: *a*) Memoria compartida versus memoria distribuida *b*) Memoria volátil vs no volátil *c*) Protocolos de coherencia **III:** Arquitecturas Específicas de Dominio (DSAs): *a*) Acelerador de Aprendizaje Automático *b*) Computación en red *c*) Sistemas embebidos para aplicaciones emergentes *d*) Computación neuromórfica *e*) Dispositivos de computación perimetral **IV:** Soluciones de empaquetado e integración como 3DIC y chiplets **V:** Aprendizaje automático en diseño de arquitectura: *a*) Algoritmos de IA para análisis de carga de trabajo *b*) Optimización de configuraciones de arquitectura para rendimiento y eficiencia

energética **VI:** Diseño Orientado a Datos (DOD): organización de datos en memoria (arreglo de estructuras vs. estructura de arreglos) para maximizar localidad de caché y aprovechar vectorización SIMD; contraste y complementariedad con el paradigma orientado a objetos.

Objetivos de Aprendizaje I: Analizar un diagrama de sistema con arquitecturas paralelas alternativas, por ejemplo, SIMD y MIMD, e identificar las diferencias clave [Evaluar] **II:** Discutir qué problemas de gestión de memoria se encuentran en multiprocesadores que no están presentes en uniprocesadores y cómo estos problemas podrían resolverse [Evaluar] **III:** 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 [Familiarizarse] **IV:** Discutir cómo determinarías cuándo usar un acelerador específico de dominio en lugar de una CPU de propósito general [Evaluar] **V:** Enumerar diferencias clave en los principios de diseño arquitectónico entre una unidad de procesamiento basada en vector y basada en escalar [Familiarizarse] **VI:** Enumerar las ventajas y desventajas de una arquitectura PIM [Familiarizarse] **VII:** Reestructurar una colección de objetos (AoS) a un diseño DOD (SoA) y medir el impacto en rendimiento [Usar].

Bibliografía: [Patterson and Hennessy, 2020, Stallings, 2015]

5.6 Arquitecturas de Procesador Seguras, 8 hr(s)

Competencias: AG-C03,AG-C09

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

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

Bibliografía: [Patterson and Hennessy, 2020, Anderson, 2020]

6. Bibliografía

[Anderson, 2020] Anderson, R. J. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 3rd edition.

[Harris and Harris, 2012] Harris, D. and Harris, S. (2012). *Digital Design and Computer Architecture*. Morgan Kaufmann, 2nd edition.

[Patterson and Hennessy, 2020] Patterson, D. A. and Hennessy, J. L. (2020). *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*. Morgan Kaufmann, 2nd edition.

[Stallings, 2015] Stallings, W. (2015). *Computer Organization and Architecture: Designing for Performance*. Pearson, 10th edition.

[Tanenbaum and Austin, 2012] Tanenbaum, A. S. and Austin, T. (2012). *Structured Computer Organization*. Pearson, 6th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS271 Bases de Datos I (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS112 Programación Orientada a Objetos I (2do Sem-Pág 80)
  - CS1D1 Estructuras Discretas (2do Sem-Pág 85)

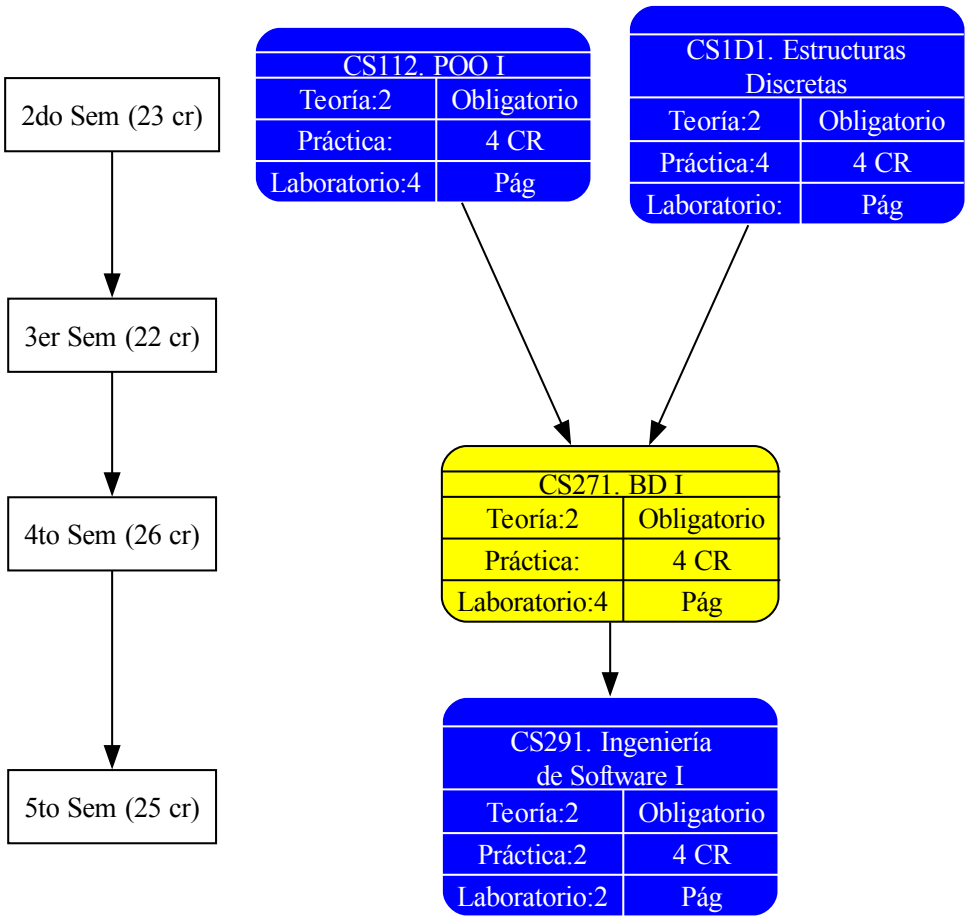


Figura 3.23: Mapa de Conexión. CS271 Bases de Datos I

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Conceptos Fundamentales de Sistemas de Bases de Datos, 8 hr(s)

Competencias: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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: AG-C09,AG-C12

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS401 Metodología de la Investigación (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - CS100 Introducción a la Ciencia de la Computación (2do Sem-Pág 69)

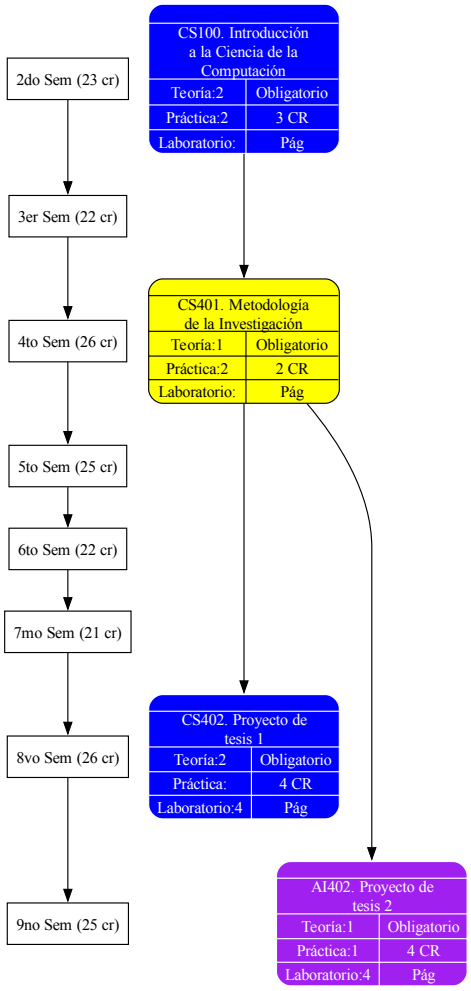


Figura 3.24: Mapa de Conexión. CS401 Metodología de la Investigación

2. Fundamentación

Este curso tiene por objetivo que el alumno aprenda a realizar una investigación de carácter científico en el área de computación. Los docentes del curso determinarán un área de estudio para cada alumno y se les hará entrega de bibliografía para analizar. A partir de la misma y de fuentes adicionales investigadas por el alumno, el alumno deberá construir un artículo del tipo survey sobre el tema asignado.

3. Objetivos del curso.

- Aprender a iniciar una investigación científica en el área de computación con rigor metodológico.
- Conocer las principales fuentes bibliográficas académicas: IEEE Xplore, ACM Digital Library, Google Scholar y arXiv.
- Analizar críticamente propuestas existentes sobre un tópico y relacionarlas en una revisión bibliográfica.
- Redactar documentos técnicos en computación usando LaTeXcon plantillas de conferencias IEEE y ACM.
- Reproducir resultados existentes en un tópico mediante implementación y experimentación.
- Construir un artículo de tipo survey con taxonomía, síntesis y comparación de enfoques existentes.
- Los entregables de este curso son:

  Avance parcial: Dominio del tema del artículo y bibliografía preliminar en formato de artículo LaTeX.

  Final: Artículo del tipo survey concluido, conteniendo opcionalmente los resultados experimentales de la(s) técnica(s) estudiada(s).

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Fundamentos de Investigación Científica en Computación, 8 hr(s)

Competencias: AG-C05,AG-C08

Tópicos: **I:** Tipos de investigación en computación: empírica, teórica, constructiva y evaluativa. **II:** El método científico aplicado a la investigación en computación. **III:** Principales bases de datos académicas: IEEE Xplore, ACM Digital Library, Google Scholar, Semantic Scholar y arXiv. **IV:** Estrategias de búsqueda bibliográfica: operadores booleanos, seguimiento de citas y búsqueda sistemática. **V:** Gestores de referencias: Zotero, Mendeley e integración con BibTeX.

Objetivos de Aprendizaje **I:** Identificar los tipos de investigación en computación y sus diferencias metodológicas. [Familiarizarse] **II:** Utilizar las principales bases de datos académicas para encontrar literatura relevante en un tópico dado. [Usar] **III:** Organizar referencias bibliográficas usando gestores de referencias y formato BibTeX. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.2 Lectura Crítica y Análisis de Literatura Científica, 8 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Estructura de un artículo científico: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. **II:** Técnicas para la lectura eficiente de artículos científicos (lectura en tres pasadas). **III:** Evaluación crítica: identificación de contribuciones, limitaciones y problemas abiertos. **IV:** Construcción de una matriz de lectura y tablas comparativas de trabajos relacionados. **V:** Clasificación y diseño de taxonomías de enfoques existentes sobre un tópico.

Objetivos de Aprendizaje **I:** Leer y evaluar críticamente artículos científicos identificando sus contribuciones y limitaciones. [Usar] **II:** Construir una matriz de lectura organizada que sintetice múltiples trabajos sobre un tópico dado. [Usar] **III:** Diseñar una taxonomía preliminar de enfoques para un área de investigación asignada. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Ética en Investigación e Integridad Académica, 6 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** Integridad académica: plagio, autoplagio, fabricación y falsificación de datos. **II:** Ética en investigación en computación: privacidad, sesgo algorítmico y estudios con sujetos humanos. **III:** Acceso abierto, licencias (Creative Commons, GPL) y propiedad intelectual en software. **IV:** Buenas prácticas de citación y atribución de autoría en trabajos colaborativos. **V:** El proceso de revisión por pares: rol del revisor y responsabilidades del autor.

Objetivos de Aprendizaje **I:** Distinguir las formas de mala conducta académica y aplicar prácticas de citación correctas. [Usar] **II:** Analizar dilemas éticos en investigación computacional aplicando principios de integridad. [Evaluar] **III:** Explicar los derechos de propiedad intelectual y las licencias relevantes en computación. [Familiarizarse]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Redacción Técnica y Herramientas de Escritura Científica, 10 hr(s)

Competencias: AG-C04,AG-C09

Tópicos: **I:** Estructura de LaTeX: preámbulo, secciones, ecuaciones, figuras, tablas y referencias cruzadas. **II:** Plantillas de conferencias y revistas: IEEE y ACM. **III:** Estilo de escritura académica en computación: claridad, precisión y voz activa. **IV:** Redacción de cada sección del artículo: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. **V:** Integración Continua de documentos: control de versiones de documentos LaTeXcon Git. **VI:** Gestión de bibliografía con BibTeX y JabRef.

Objetivos de Aprendizaje **I:** Redactar un artículo técnico completo en LaTeXusando plantillas IEEE o ACM. [Usar] **II:** Aplicar las convenciones de escritura académica propias de la comunidad de computación. [Usar] **III:** Evaluar la calidad de redacción de un artículo propio o ajeno según criterios de claridad y estructura. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Revisión Sistemática de Literatura y Artículos de Survey, 10 hr(s)

Competencias: AG-C04,AG-C08,AG-C09

Tópicos: **I:** Tipos de revisiones: narrativa, sistemática, estudio de mapeo y meta-análisis. **II:** Protocolo de revisión sistemática de literatura (SLR): preguntas de investigación, estrategia de búsqueda y criterios de selección. **III:** Pautas PRISMA: diagrama de flujo de selección de estudios primarios. **IV:** Técnicas de síntesis: narrativa, votación y análisis temático. **V:** Diseño de taxonomías y esquemas de clasificación para un survey. **VI:** Estructura y redacción de un artículo del tipo survey.

Objetivos de Aprendizaje I: Diseñar y ejecutar un protocolo de revisión sistemática de literatura para un tópico de computación. [Usar] **II:** Aplicar las pautas PRISMA para documentar y reportar el proceso de selección de estudios. [Usar] **III:** Redactar un artículo de tipo survey con taxonomía, síntesis crítica y comparación de enfoques. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.6 Presentación y Difusión de Resultados de Investigación, 6 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: I: Técnicas de presentación oral para audiencias científicas: estructura, manejo del tiempo y sesión de preguntas. **II:** Diseño de pósters para conferencias y simposios de investigación. **III:** Respuesta a revisores: cómo abordar comentarios de forma constructiva y argumentada. **IV:** Planificación de un proyecto de investigación: cronograma, hitos y entregables. **V:** Prácticas de colaboración y coautoría en investigación científica.

Objetivos de Aprendizaje I: Preparar y exponer una presentación oral de resultados de investigación ante una audiencia académica. [Usar] **II:** Elaborar un plan de trabajo para un proyecto de investigación con cronograma e hitos. [Usar] **III:** Responder constructivamente a comentarios de revisores en el proceso de publicación. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library*. Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library*. College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library*. IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial
AI261 Probabilidad y Estadistica para IA (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - BMA102 Cálculo Diferencial (1er Sem-Pág 57)
  - BMA101 Álgebra Lineal (1er Sem-Pág 54)
  - STA251 Estadística y Probabilidades (3er Sem-Pág 113)

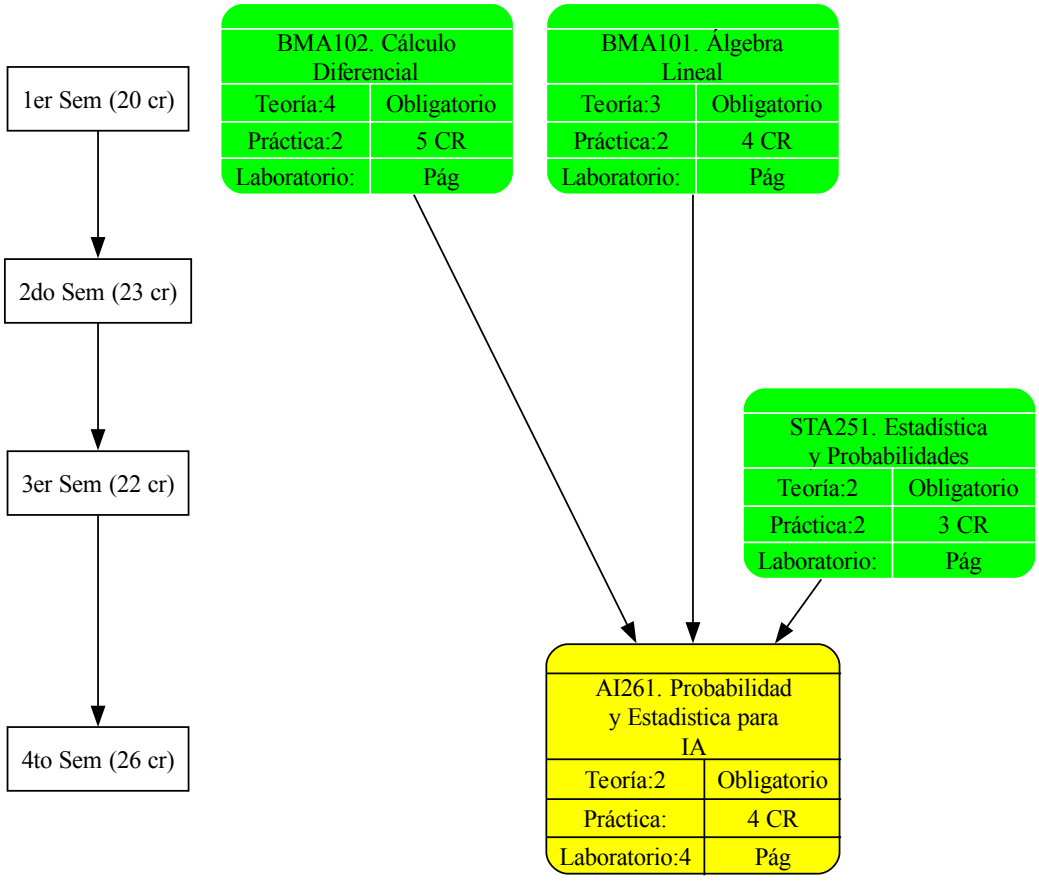


Figura 3.25: Mapa de Conexión. AI261 Probabilidad y Estadistica para IA

2. Fundamentación

Este curso proporciona las bases probabilísticas y estadísticas esenciales para el desarrollo de sistemas de Inteligencia Artificial. Cubre distribuciones clave, inferencia bayesiana y métodos estadísticos aplicados directamente a problemas de aprendizaje automático, visión por computadora y procesamiento de lenguaje natural.

3. Objetivos del curso.

- Comprender las distribuciones de probabilidad fundamentales en modelos de IA.

- Aplicar métodos de inferencia estadística en el entrenamiento de modelos.
- Implementar soluciones probabilísticas usando bibliotecas modernas (NumPy, SciPy).

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Usar)

5. Contenido

5.1 Fundamentos Probabilísticos, 15 hr(s)

Competencias: AG-C07

Tópicos: **I:** Espacios de probabilidad y variables aleatorias **II:** Distribuciones clave: Normal, Poisson, Binomial, Multinomial **III:** Teorema de Bayes y aplicaciones en IA

Objetivos de Aprendizaje **I:** Calcular probabilidades en modelos discretos/continuos [Usar] **II:** Explicar el rol de distribuciones en algoritmos de ML [Familiarizarse]

Bibliografía: [Bishop, 2006, Murphy, 2012]

5.2 Estadística para IA, 15 hr(s)

Competencias: AG-C07

Tópicos: **I:** Estimación de parámetros (MLE, MAP) **II:** Intervalos de confianza y pruebas de hipótesis **III:** Análisis de correlación y regresión lineal

Objetivos de Aprendizaje **I:** Implementar MLE/MAP en Python [Usar] **II:** Evaluar significancia estadística en resultados de modelos [Evaluar]

Bibliografía: [DeGroot and Schervish, 2014, Wasserman, 2004]

5.3 Cadenas de Markov y Procesos Estocásticos, 15 hr(s)

Competencias: AG-C07

Tópicos: **I:** Procesos de Markov ocultos (HMMs) **II:** Cadenas de Markov Monte Carlo (MCMC) **III:** Aplicaciones en NLP y visión computacional

Objetivos de Aprendizaje **I:** Simular procesos estocásticos básicos [Usar] **II:** Relacionar teoría con arquitecturas de redes neuronales [Familiarizarse]

Bibliografía: [Ross, 2019, Bishop, 2006]

5.4 Herramientas Computacionales, 15 hr(s)

Competencias: AG-C07

Tópicos: **I:** NumPy para operaciones probabilísticas **II:** SciPy para tests estadísticos **III:** Visualización con Matplotlib/Seaborn

Objetivos de Aprendizaje **I:** Programar simulaciones probabilísticas [Usar] **II:** Generar visualizaciones estadísticas [Evaluar]

Bibliografía: [VanderPlas, 2016, McKinney, 2022]

6. Bibliografía

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[DeGroot and Schervish, 2014] DeGroot, M. H. and Schervish, M. J. (2014). *Probability and Statistics*. Pearson, 4th edition.

[McKinney, 2022] McKinney, W. (2022). *Python for Data Analysis*. O'Reilly, 3rd edition.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[Ross, 2019] Ross, S. M. (2019). *Introduction to Probability Models*. Academic Press, 12th edition.

[VanderPlas, 2016] VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly.

[Wasserman, 2004] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer. Aunque es de 2004, sigue siendo referencia estándar.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

FG350 Liderazgo y Desempeño (Obligatorio)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 2 horas;
- **Prerrequisitos**:
  - FG106 Teatro (3er Sem-Pág 118)

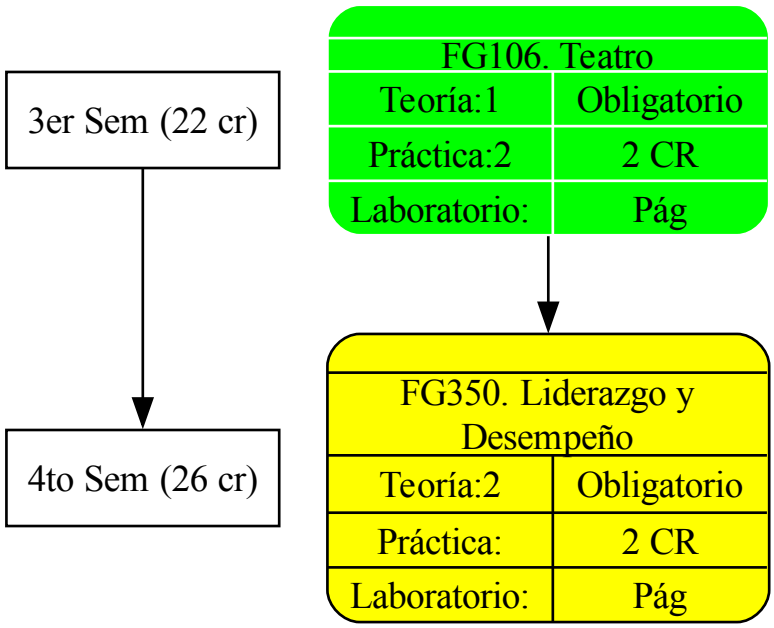


Figura 3.26: Mapa de Conexión. FG350 Liderazgo y Desempeño

2. Fundamentación

Las organizaciones contemporáneas exigen de sus integrantes el ejercicio efectivo del liderazgo: asumir con eficacia y espíritu de servicio los retos que se les encomiendan, en la búsqueda de una sociedad más justa y reconciliada. Este desafío pasa por formar personas con un recto conocimiento de sí mismas, con capacidad de juzgar objetivamente la realidad y de proponer orientaciones que modifiquen positivamente el entorno. El curso combina fundamentos teóricos con el desarrollo vivencial de competencias, capacitando al estudiante para liderar equipos con integridad, coherencia personal y genuino servicio a los demás.

3. Objetivos del curso.

- Comprender los fundamentos teóricos del liderazgo y su dimensión ética, integrándolos a un proceso de autoconocimiento.
- Identificar y desarrollar las competencias y virtudes personales necesarias para el ejercicio eficaz del liderazgo.

- Adquirir habilidades prácticas para la gestión, motivación y dirección de equipos de alto desempeño.
- Contribuir a la construcción de una sociedad más justa a través del liderazgo con sentido de servicio y responsabilidad social.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Fundamentos del Liderazgo, 15 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Definición y concepto de liderazgo. **II:** Principales teorías del liderazgo: transaccional, transformacional y situacional. **III:** Visión integral del ser humano: dimensiones corporal, afectiva y espiritual. **IV:** Motivos de la acción humana: extrínsecos, intrínsecos y trascendentes. **V:** La práctica de la virtud en el ejercicio del liderazgo. **VI:** El líder como modelo: ética y coherencia personal. **VII:** Estilos de liderazgo y su aplicación en distintos contextos.

Objetivos de Aprendizaje I: Analizar y comprender las bases teóricas del ejercicio del liderazgo [Familiarizarse]. **II:** Asumir la actitud correcta para llevar el liderazgo a la práctica [Familiarizarse]. **III:** Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo [Familiarizarse]. **IV:** Reconocer la dimensión ética del liderazgo y su impacto en el entorno personal y social [Familiarizarse].

Bibliografía: [Cardona and Lombardi, 2002, Ferreiro and Alcázar, 2009, Dianine-Havard, 2007, D'Souza, 1994]

5.2 Formación de Virtudes y Competencias, 15 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Teoría de las competencias y su aplicación al liderazgo. **II:** Reconocimiento e inventario de competencias personales. **III:** Plan de desarrollo personal: diagnóstico, metas y seguimiento. **IV:** Modelos mentales y su influencia en el comportamiento. **V:** Necesidades y perfiles emocionales. **VI:** Vicios motivacionales: identificación y superación. **VII:** Inteligencia emocional aplicada al liderazgo. **VIII:** La toma de decisiones bajo presión y ambigüedad.

Objetivos de Aprendizaje I: Conocer y desarrollar competencias de liderazgo centradas en la eficacia y el servicio [Familiarizarse]. **II:** Reconocer las tendencias personales y grupales necesarias para el ejercicio del liderazgo [Familiarizarse]. **III:** Elaborar un plan de desarrollo personal de competencias de liderazgo [Usar].

Bibliografía: [Cardona and Wilkinson, 2009, Huete, 2008, Cardona and Lombardi, 2002, Chinchilla and Moragas, 2007, Goleman, 2006]

5.3 Gestión y Dirección de Equipos, 18 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: La relación personal con el equipo: confianza y comunicación. **II:** Liderazgo integral: misión, visión y valores compartidos. **III:** Delegación y acompañamiento: el arte de motivar y delegar. **IV:** Coaching y mentoring como herramientas de desarrollo. **V:** Resolución de conflictos

y dinámica grupal. **VI:** Las cinco disfunciones de un equipo y cómo superarlas. **VII:** Fundamentos de unidad: cohesión y cultura organizacional. **VIII:** El liderazgo de servicio y su impacto en la organización.

Objetivos de Aprendizaje I: Desarrollar habilidades para el trabajo en equipo eficaz y cohesionado [Familiarizarse]. **II:** Aplicar técnicas de liderazgo situacional para dirigir equipos en distintos contextos [Usar]. **III:** Diseñar estrategias para la resolución de conflictos y la mejora del desempeño grupal [Usar].

Bibliografía: [Goleman, 2006, Cardona and na, 2008, Hersey et al., 2007, Hunsaker, 2001, Hawkins, 2011, Lencioni, 2002]

6. Bibliografía

[Cardona and Lombardi, 2002] Cardona, P. and Lombardi, P. G. (2002). *Cómo desarrollar las Competencias de Liderazgo*. PAD, Lima, 3rd edition.

[Cardona and na, 2008] Cardona, P. and na, C. R. P. (2008). *Dirección por misiones: Cómo generar empresas de alto rendimiento*. Deusto.

[Cardona and Wilkinson, 2009] Cardona, P. and Wilkinson, H. (2009). *Creciendo como Líder*. Ediciones Universidad de Navarra EUNSA.

[Chinchilla and Moragas, 2007] Chinchilla, N. and Moragas, M. (2007). *Dueños de Nuestro Destino*. Editorial Ariel.

[Dianine-Havard, 2007] Dianine-Havard, A. (2007). *Virtuous Leadership: An Agenda for Personal Excellence*. Scepter Publishers.

[D'Souza, 1994] D'Souza, A. (1994). *Discover Your Leadership Style*. St. Pauls.

[Ferreiro and Alcázar, 2009] Ferreiro, P. and Alcázar, M. (2009). *Gobierno de Personas en la Empresa*. Ediciones Universidad de Navarra EUNSA.

[Goleman, 2006] Goleman, D. (2006). *Emotional Intelligence: Why It Can Matter More Than IQ*. Bantam Books.

[Hawkins, 2011] Hawkins, P. (2011). *Leadership Team Coaching: Developing Collective Transformational Leadership*. Kogan Page.

[Hersey et al., 2007] Hersey, P., Blanchard, K. H., and Johnson, D. E. (2007). *Management of Organizational Behavior: Leading Human Resources*. Prentice Hall, 9th edition.

[Huete, 2008] Huete, L. M. (2008). *Construye tu Sueño*. LID Editorial Empresarial.

[Hunsaker, 2001] Hunsaker, P. (2001). *Training in Management Skills*. Pearson Prentice Hall.

[Lencioni, 2002] Lencioni, P. (2002). *The Five Dysfunctions of a Team: A Leadership Fable*. Jossey-Bass.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BEI202 Inglés IV (Electivo)

1. Información general

- **Semestre**: 4to Sem. **Créditos**: 2
- **Horas del curso**: **Práctica**: 4 horas;
- **Prerrequisitos**:
  - BEI201 Inglés III (3er Sem-Pág 122)

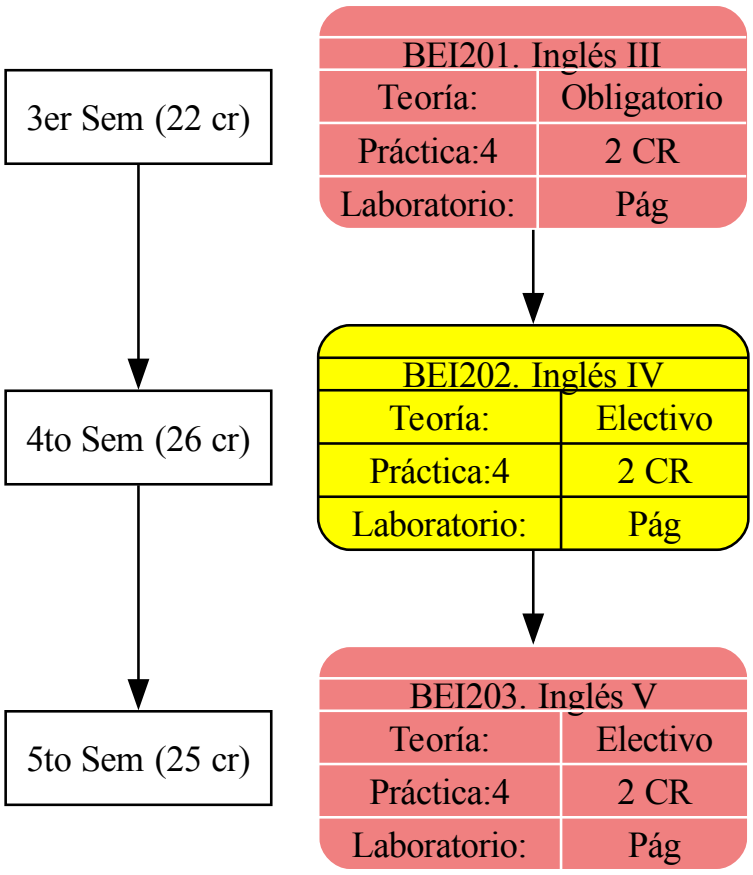


Figura 3.27: Mapa de Conexión. BEI202 Inglés IV

2. Fundamentación

En el nivel intermedio, los estudiantes aplican el inglés al rango completo de comunicación técnica requerida en la práctica de la computación y la ingeniería: lectura de artículos de investigación, redacción de documentos técnicos estructurados, análisis de estudios de caso y participación en diálogos profesionales. Las estructuras gramaticales se enseñan a través de contextos auténticos de computación e ingeniería, reforzando simultáneamente la competencia lingüística y el conocimiento disciplinar.

3. Objetivos del curso.

- Desarrollar la capacidad de escribir y leer documentos técnicos estructurados en inglés.
- Llevar al estudiante a una expresión más fluida y natural en el inglés técnico y científico.
- Dominar el vocabulario de computación, ingeniería de sistemas y ciencias aplicadas.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Familiarizarse)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Buenas Prácticas y Ética, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Modales de obligación y consejo: Should, Must, Have to en códigos de ingeniería. **II:** Prohibición y ausencia de obligación en licencias de software y privacidad de datos. **III:** Vocabulario: Ética profesional, propiedad intelectual, IA responsable. **IV:** Redacción: Documentos formales de recomendación técnica y política.

Objetivos de Aprendizaje I: Al finalizar la unidad, el estudiante utiliza correctamente los auxiliares modales para expresar estándares de ingeniería, obligaciones éticas y recomendaciones profesionales en entornos técnicos formales.

Bibliografía: [? line:47, ? line:47, ? line:47]

5.2 Fallos y Causas Raíz, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Pasado Perfecto: Estructura y uso en análisis de causa raíz y post-mortems. **II:** Contraste entre Pasado Simple, Continuo y Perfecto en cronologías de fallos. **III:** Vocabulario: Interrupciones de sistema, brechas de seguridad, fallos en cascada. **IV:** Expresiones para reportar información: estilo indirecto en informes de incidentes.

Objetivos de Aprendizaje I: El estudiante reconstruye cronologías complejas de fallos y análisis de causas raíz estableciendo secuencias cronológicas claras mediante el contraste de tiempos de pasado.

Bibliografía: [? line:60, ? line:60, ? line:60]

5.3 Lógica Si-Entonces, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Condicionales Tipo 0, 1 y 2 aplicados a la lógica computacional y decisiones de ingeniería. **II:** Oraciones con Ïfz Ünless"para restricciones y requisitos de sistemas. **III:** Vocabulario: Escenarios hipotéticos, análisis de riesgo y trade-offs de optimización. **IV:** Pensamiento de diseño hipotético: Ïf we used X, the system would..."

Objetivos de Aprendizaje I: El estudiante plantea escenarios técnicos reales e hipotéticos y sus consecuencias, utilizando estructuras condicionales para razonar sobre decisiones de diseño e ingeniería.

Bibliografía: [? line:73, ? line:73, ? line:73]

5.4 Investigación en Curso, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Presente Perfecto Continuo para describir investigación en progreso. **II:** Contraste entre Presente Perfecto Simple y Continuo en escritura científica. **III:** Vocabulario: Integración continua, desarrollo ágil, investigación iterativa. **IV:** Formación de palabras: Prefijos y sufijos en terminología científica y técnica.

Objetivos de Aprendizaje **I:** El estudiante describe actividades de investigación y procesos de desarrollo que comenzaron en el pasado y continúan en el presente, enfatizando la duración y el progreso iterativo.

Bibliografía: [**?** line:86 , **?** line:86 , **?** line:86]

5.5 Fabricación y Despliegue, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Voz Pasiva en todos los tiempos básicos para descripciones de procesos técnicos. **II:** Verbos con dos objetos en pasiva: data is sent to the server by the client. **III:** Vocabulario: Fabricación de hardware, despliegue en la nube, pipelines CI/CD. **IV:** Redacción: Descripción técnica de un proceso de despliegue o producción.

Objetivos de Aprendizaje **I:** El estudiante utiliza la voz pasiva en múltiples tiempos verbales para describir procesos técnicos donde la acción, el sistema o el resultado es el foco en lugar del agente.

Bibliografía: [**?** line:99 , **?** line:99 , **?** line:99]

5.6 Hojas de Ruta Tecnológicas, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Futuro Continuo y Futuro Perfecto para cronogramas e hitos de proyectos. **II:** Adverbios de probabilidad: perhaps, probably, definitely — en prospectiva tecnológica. **III:** Vocabulario: Hojas de ruta de producto, planificación de versiones, áreas de investigación emergente. **IV:** Presentaciones orales: .[El] estado de la IA / computación cuántica en 5 años".

Objetivos de Aprendizaje **I:** El estudiante proyecta estados técnicos, hitos futuros completados y resultados de investigación, discutiendo hojas de ruta de computación e ingeniería con vocabulario avanzado.

Bibliografía: [**?** line:112 , **?** line:112 , **?** line:112]

5.7 Equipos Técnicos, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Preguntas indirectas y etiquetas (Tag questions) en entrevistas técnicas y revisiones de pares. **II:** Phrasal Verbs comunes en entornos de ingeniería e investigación. **III:** Vocabulario: Herramientas de colaboración, prácticas de revisión de código, roles de equipo. **IV:** Redacción: Publicaciones cortas de blog técnico y actualizaciones de equipo de ingeniería.

Objetivos de Aprendizaje **I:** El estudiante utiliza formas de consulta cortés y estrategias de confirmación en entrevistas técnicas y discusiones de equipo, mejorando su fluidez y naturalidad en contextos profesionales de ingeniería.

Bibliografía: [**?** line:125 , **?** line:125 , **?** line:125]

5.8 Inglés Técnico IV: Redes y Seguridad, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Terminología de redes: protocolos, topologías, modelo OSI, enrutamiento. **II:** Vocabulario de ciberseguridad: vulnerabilidades, amenazas, cifrado, autenticación. **III:** Lectura y redacción de informes de incidentes de red y auditorías de seguridad. **IV:** Proyecto: Evaluación de seguridad para una infraestructura computacional.

Objetivos de Aprendizaje I: Al finalizar el curso, el estudiante domina la terminología técnica de redes y ciberseguridad para leer documentación especializada y comunicarse eficazmente con profesionales de TI y seguridad en inglés.

Bibliografía: [**?** line:138, **?** line:138, **?** line:138]

6. Bibliografía

line:143 line:143 line:143

Quinto Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS212 Análisis y Diseño de Algoritmos (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 132)
  - CS211 Teoría de la Computación (4to Sem-Pág 137)

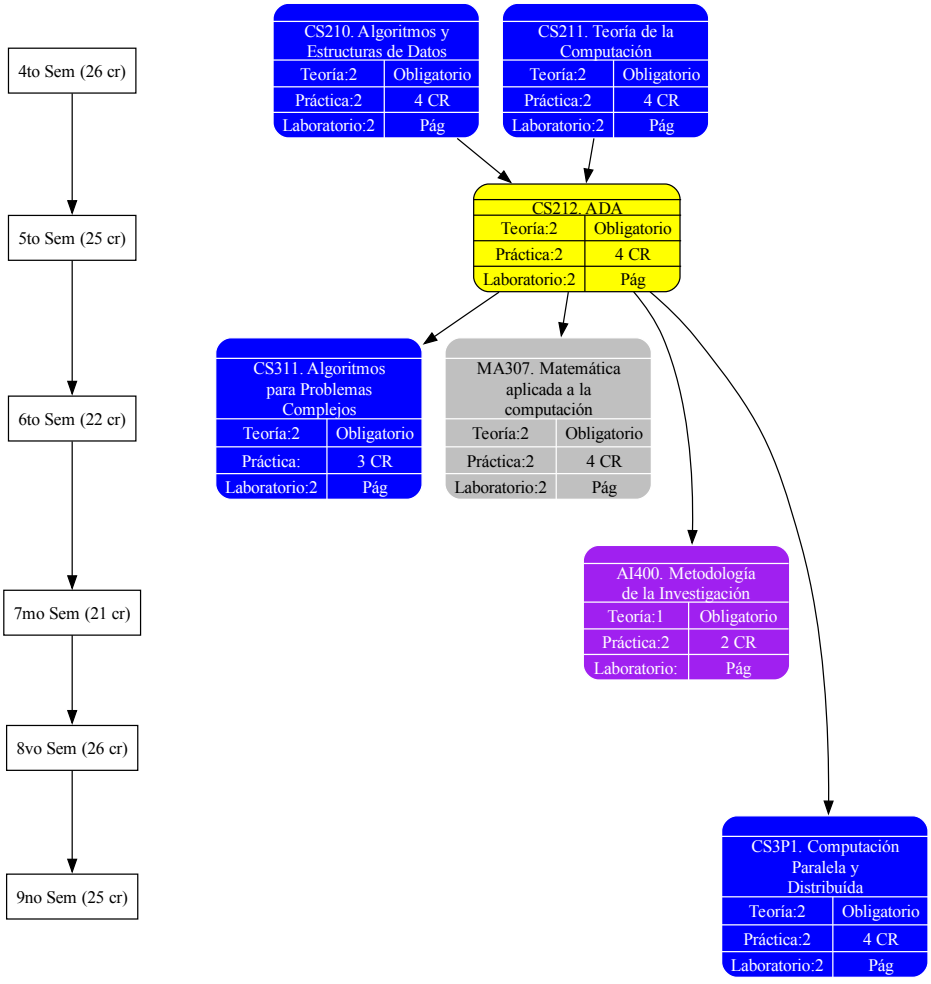


Figura 3.28: Mapa de Conexión. CS212 Análisis y Diseño de Algoritmos

2. Fundamentación

Los algoritmos son el corazón de la ciencia de la computación. Este curso proporciona las herramientas matemáticas y analíticas necesarias para evaluar la eficiencia de los algoritmos en términos de

tiempo y espacio. Los estudiantes explorarán estructuras de datos fundamentales, análisis de complejidad y varios paradigmas de diseño, como fuerza bruta, algoritmos voraces y programación dinámica. Comprender estas técnicas es esencial para construir software que no solo sea funcional, sino también escalable y eficiente en el uso de recursos.

3. Objetivos del curso.

- Dominar el uso de la notación asintótica para analizar el rendimiento de los algoritmos.
- Aplicar estructuras de datos fundamentales para resolver problemas computacionales.
- Usar estrategias algorítmicas como Voraz, Divide y Vencerás y Programación Dinámica.
- Analizar la complejidad de algoritmos en el mejor, promedio y peor de los casos.
- Desarrollar la capacidad de seleccionar el algoritmo más apropiado para un problema dado.

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Estructuras de Datos Fundamentales, 6 hr(s)

Competencias: AG-C08,AG-C12

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. [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: [Cormen et al., 2022, Kleinberg and Tardos, 2005]

5.2 Algoritmos Fundamentales, 6 hr(s)

Competencias: AG-C08,AG-C12

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 **VII:** Emparejamiento: *a*) Emparejamiento eficiente de cadenas (por ejemplo, Boyer-Moore, Knuth-Morris-Pratt) *b*) Emparejamiento de subsecuencia común más larga *c*) Emparejamiento de expresiones regulares

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] **V:** 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]

Bibliografía: [Cormen et al., 2022, Kleinberg and Tardos, 2005]

5.3 Algoritmos Avanzados, 4 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Algoritmos de criptografía (por ejemplo, SHA-256) **II:** Algoritmos paralelos **III:** Algoritmos de consenso (por ejemplo, Blockchain): *a*) Prueba de trabajo vs prueba de participación **IV:** Algoritmos de computación cuántica: *a*) Basados en oráculo (por ejemplo, Deutsch-Jozsa, Bernstein-Vazirani, Simon) *b*) Aceleración superpolinomial mediante QFT (por ejemplo, Shor) *c*) Aceleración polinomial mediante amplificación de amplitud (por ejemplo, Grover) **V:** Algoritmo de Transformada Rápida de Fourier (FFT) **VI:** Algoritmo de evolución diferencial

Objetivos de Aprendizaje I: Una apreciación de la computación cuántica y su aplicación a ciertos problemas [Explicar]

Bibliografía: [Cormen et al., 2022, Kleinberg and Tardos, 2005]

5.4 Estrategias Algorítmicas, 24 hr(s)

Competencias: AG-C08,AG-C12

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:** Manejo del crecimiento exponencial (por ejemplo, heurística A*, ramificación y poda, retroceso) **III:** Iteración vs recursión (por ejemplo, factorial, búsqueda en árbol) **IV:** Paradigmas: *a*) Algoritmos de aproximación *b*) Mejora iterativa (por ejemplo, Ford-Fulkerson, simplex) *c*) Algoritmos aleatorizados/estocásticos (por ejemplo, corte máximo, bolas y cubos) **V:** Computación cuántica

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. [Explicar] **II:** 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] **III:** Dado un algoritmo, explicar el paradigma utilizado por el algoritmo

y cómo ejemplifica este paradigma [Explicar] **IV:** 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] **V:** Dar ejemplos de algoritmos iterativos y recursivos que resuelvan el mismo problema, explicar los beneficios y desventajas de cada enfoque [Explicar] **VI:** Evaluar si un enfoque voraz conduce a una solución óptima [Evaluar] **VII:** Explicar varios enfoques para abordar problemas computacionales cuyas soluciones algorítmicas son exponenciales [Explicar]

Bibliografía: [Cormen et al., 2022, Dasgupta et al., 2006]

5.5 Marco de Análisis de Complejidad, 6 hr(s)

Competencias: AG-C08,AG-C12

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: 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] **II:** 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] **III:** Explicar ejemplos que ilustren los compromisos tiempo-espacio de algoritmos [Explicar] **IV:** Explicar cómo el balance del árbol afecta la eficiencia de las operaciones de árbol de búsqueda binaria [Explicar]

Bibliografía: [Cormen et al., 2022, Sipser, 2012]

5.6 Notación Asintótica y Clases de Complejidad, 6 hr(s)

Competencias: AG-C08,AG-C12

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: Para cada clase de complejidad fundamental en esta unidad, explicar un algoritmo que demuestre la complejidad de tiempo de ejecución asociada [Explicar] **II:** Aplicar la notación Big-O para dar cotas superiores en la complejidad tiempo/espacio de algoritmos [Aplicar]

Bibliografía: [Cormen et al., 2022, Sipser, 2012]

5.7 Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Notaciones Little-o, Little-Omega y Little Theta **II:** Análisis recursivo formal **III:** Análisis amortizado

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] **II:** Usar relaciones de recurrencia para evaluar la complejidad de tiempo de algoritmos definidos recursivamente [Aplicar] **III:** Aplicar relaciones de recurrencia elementales usando una forma del Teorema Maestro [Aplicar]

Bibliografía: [Cormen et al., 2022, Sipser, 2012]

5.8 Teoría de la Complejidad Computacional, 6 hr(s)

Competencias:

Tópicos: I: Tractabilidad e intractabilidad: *a*) Clases de Complejidad P, NP y NP-Completo *b*) Problemas NP-Completos (por ejemplo, SAT, Mochila, TSP) *c*) Reducciones

Objetivos de Aprendizaje I: Explicar la importancia de la NP-Completitud [Explicar] **II:** Explicar ejemplos de problemas NP-completos [Explicar] **III:** Explicar las clases P y NP [Explicar]

Bibliografía: [Sipser, 2012, Cormen et al., 2022]

6. Bibliografía

[Cormen et al., 2022] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to Algorithms*. MIT Press, 4th edition.

[Dasgupta et al., 2006] Dasgupta, S., Papadimitriou, C., and Vazirani, U. (2006). *Algorithms*. McGraw-Hill Education.

[Kleinberg and Tardos, 2005] Kleinberg, J. and Tardos, É. (2005). *Algorithm Design*. Pearson.

[Sipser, 2012] Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS261 Inteligencia Artificial (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - STA251 Estadística y Probabilidades (3er Sem-Pág 113)

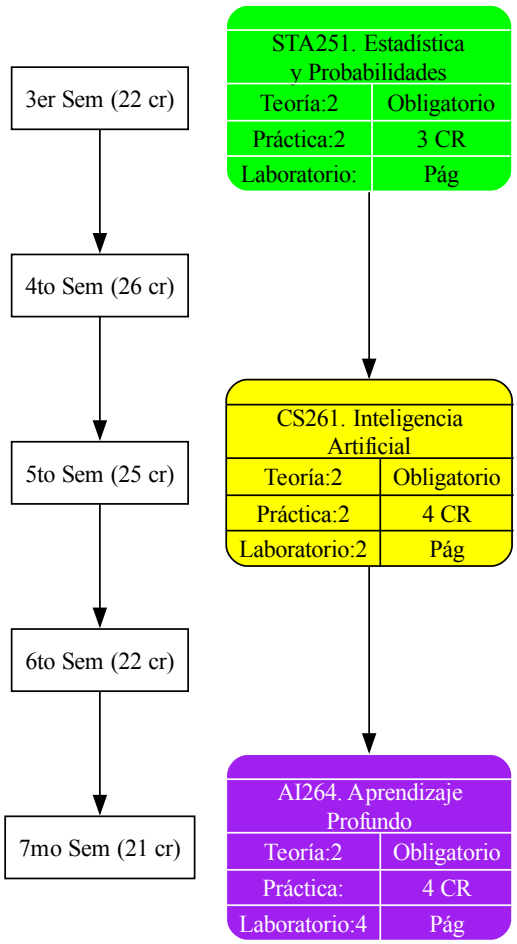


Figura 3.29: Mapa de Conexión. CS261 Inteligencia Artificial

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*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Introducción a la IA: Conceptos y Comportamiento Inteligente, 4 hr(s)

Competencias: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C09

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: AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS291 Ingeniería de Software I (Obligatorio)

1. Información general

- **Semestre**: 5to 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 99)
  - CS271 Bases de Datos I (4to Sem-Pág 147)

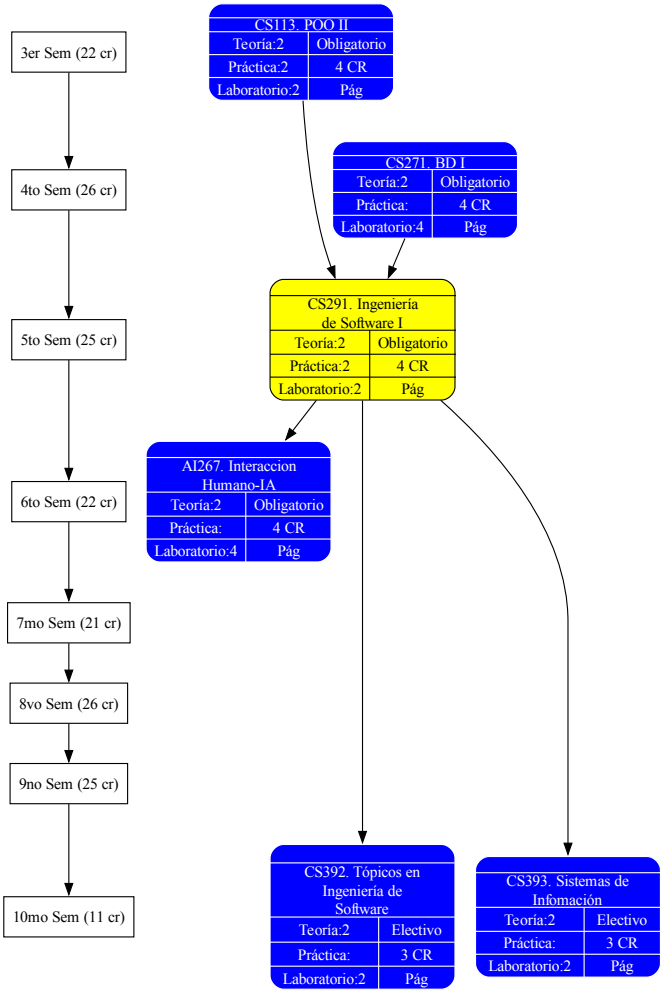


Figura 3.30: Mapa de Conexión. CS291 Ingeniería de Software I

2. Fundamentación

La ingeniería de software es una disciplina colaborativa que aplica procesos estructurados para construir sistemas de alta calidad. Este primer curso se enfoca en los aspectos fundamentales del ciclo de vida del software, desde los requisitos iniciales y el diseño centrado en el usuario hasta las pruebas sistemáticas. Al integrar principios de HCI y fundamentos de desarrollo rigurosos, los estudiantes aprenderán a crear software que sea técnicamente sólido y orientado al usuario.

3. Objetivos del curso.

- Comprender y aplicar varios modelos de proceso de software y gestión de configuración.
- Dominar la elicitación y modelado de requisitos usando notaciones estandarizadas.
- Diseñar interfaces de usuario intuitivas basadas en principios de evaluación y diseño de HCI.
- Implementar estrategias sistemáticas de prueba y comprender la calidad del software.
- Desarrollar habilidades profesionales para el desarrollo colaborativo de software.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Control de Versiones y CI/CD, 12 hr(s)

Competencias: AG-C03,AG-C09

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:** 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 **IV:** 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. **V:** Flujo de revisión colaborativa como puerta de calidad *a*) Revisión obligatoria de *pull requests* antes de fusionar *b*) Propiedad del código mediante reglas de tipo CODEOWNERS *c*) Convenciones de mensajes de *commit* (*conventional commits*) *d*) Reglas de protección de ramas que exigen verificaciones antes de fusionar **VI:** Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI

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:** Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] **V:** 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] **VI:** 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] **VII:** Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]

Bibliografía: [Sommerville, 2015, Pressman and Maxim, 2019]

5.2 Ingeniería de Requisitos, 14 hr(s)

Competencias: AG-C03,AG-C09

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:** Propiedades de los requisitos, incluyendo consistencia, validez, completitud y viabilidad. **III:** 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. **IV:** 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 **V:** 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 **VI:** Comunicar y/o formalizar especificaciones de requisitos.

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]

Bibliografía: [Sommerville, 2015, Wiegers and Beatty, 2013]

5.3 Colaboración y Dinámica de Equipo, 14 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** 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 **II:** Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos. **III:** Programación cooperativa: *a*) Programación por parejas (pair programming) o Enjambre (swarming) *b*) Revisión de código *c*) Colaboración mediante control de versiones **IV:** 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 *a*) Ventajas del trabajo en equipo *b*) Riesgos y complejidad de dicha colaboración **V:** Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo

Objetivos de Aprendizaje **I:** Seguir prácticas efectivas de comunicación en equipo [Analizar] **II:** 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] **III:** Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar] **IV:** Colaborar efectivamente en desarrollo/programación cooperativa [Analizar] **V:** Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer] **VI:** Redactar y seguir una agenda para una reunión de equipo [Componer] **VII:** 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]

Bibliografía: [Wigdor and Wixon, 2011, Johnson, 2013]

5.4 Principios y Arquitectura de Software, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Principios de diseño de sistemas. Confiabilidad del Sistema *a*) Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) *b*) Separación de preocupaciones (separation of concerns) *c*) Ocultación de información (information hiding) *d*) Acoplamiento y cohesión (coupling and cohesion) **II:** Arquitectura de software. Confiabilidad del Sistema *a*) Paradigmas de diseño 1) Descomposición funcional de arriba hacia abajo/diseño por capas 2) Arquitectura orientada a datos 3) Análisis y diseño orientado a objetos 4) Diseño dirigido por eventos *b*) Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) *c*) Identificar límites y dependencias de componentes **III:** Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema **IV:** 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

Objetivos de Aprendizaje **I:** Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] **II:** 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] **III:** Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] **IV:** Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]

Bibliografía: [Gamma et al., 1994, Pressman and Maxim, 2019]

5.5 Conceptos de verificación y validación, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Conceptos de verificación y validación *a*) Verificación: ¿Estamos construyendo la cosa correctamente? *b*) Validación: ¿Construimos la cosa correcta? **II:** Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? **III:** Objetivos de las pruebas *a*) Usabilidad *b*) Confiabilidad *c*) Conformidad con la especificación *d*) Rendimiento *e*) Seguridad

Objetivos de Aprendizaje **I:** Explicar por qué las pruebas son importantes [Explicar] **II:** Distinguir entre validación y verificación de programas [Distinguir] **III:** Describir diferentes objetivos de las pruebas [Describir]

Bibliografía: [Sommerville, 2015, Pressman and Maxim, 2019]

5.6 Herramientas y Análisis de Pruebas, 4 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Herramientas y automatización de verificación y validación *a*) Análisis estático *b*) Cobertura de código *c*) Fuzzing *d*) Análisis dinámico y contención de fallos (sanitizadores, etc.) *e*) Registro y seguimiento de fallos **II:** Planificación y generación de pruebas *a*) Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos *b*) Uso de números aleatorios y pseudoaleatorios en las pruebas **III:** Pruebas de sistemas asíncronos, paralelos y concurrentes **IV:** Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) **V:** Desarrollo guiado por comportamiento (BDD) *a*) Expresar criterios de aceptación como escenarios Gherkin Given/When/Then *b*) Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo **VI:** Pruebas de mutación y análisis estático de calidad de

código *a*) Pruebas de mutación para evaluar la efectividad de una suite de pruebas *b*) Detección estática de *code smells* y complejidad ciclomática excesiva

Objetivos de Aprendizaje I: Describir y comparar diferentes herramientas para verificación y validación [Describir] **II:** Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] **III:** Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] **IV:** Describir enfoques para la estimación de fallos [Describir] **V:** Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] **VI:** Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] **VII:** 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] **VIII:** Describir técnicas para la verificación y validación de artefactos que no son código [Describir] **IX:** Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] **X:** Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]

Bibliografía: [Sommerville, 2015, Pressman and Maxim, 2019]

5.7 Pruebas de Rendimiento y Evaluación Comparativa, 4 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Pruebas de rendimiento y evaluación comparativa *a*) Rendimiento (*throughput*) y latencia *b*) Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) *c*) Aceleración (*speedup*) y escalado (*scaling*) 1) Ley de Amdahl 2) Ley de Gustafson 3) Escalado suave (*soft*) y débil (*weak*) *d*) Identificar y medir figuras de mérito (*figures of merit*) *e*) Cuellos de botella de rendimiento comunes 1) Limitado por cómputo (*compute-bound*) 2) Limitado por ancho de banda de memoria 3) Limitado por latencia *f*) Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) 1) Estimación de la incertidumbre 2) Intervalos de confianza *g*) Análisis y presentación (gráficos, etc.) *h*) Técnicas de medición de tiempo (*timing*)

Objetivos de Aprendizaje I: Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] **II:** Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] **III:** Describir cuellos de botella de rendimiento comunes [Describir] **IV:** Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] **V:** Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] **VI:** 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]

Bibliografía: [Bondi, 2015, Gregg, 2020]

6. Bibliografía

[Bondi, 2015] Bondi, A. B. (2015). *Foundations of Software and System Performance Engineering: Process, Performance Modeling, Requirements, Testing, Scalability, and Practice*. Addison-Wesley, Upper Saddle River, NJ.

[Gamma et al., 1994] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

[Gregg, 2020] Gregg, B. (2020). *Systems Performance: Enterprise and the Cloud*. Addison-Wesley Professional, Boston, MA, 2nd edition.

[Johnson, 2013] Johnson, J. (2013). *Designing with the Mind in Mind*. Morgan Kaufmann, 2nd edition.

[Pressman and Maxim, 2019] Pressman, R. S. and Maxim, B. (2019). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 9th edition.

[Sommerville, 2015] Sommerville, I. (2015). *Software Engineering*. Pearson, 10th edition.

[Wiegers and Beatty, 2013] Wiegers, K. and Beatty, J. (2013). *Software Requirements*. Microsoft Press, 3rd edition.

[Wigdor and Wixon, 2011] Wigdor, D. and Wixon, D. (2011). *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS2S1 Sistemas Operativos (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS221 Arquitectura de Computadores (4to Sem-Pág 142)

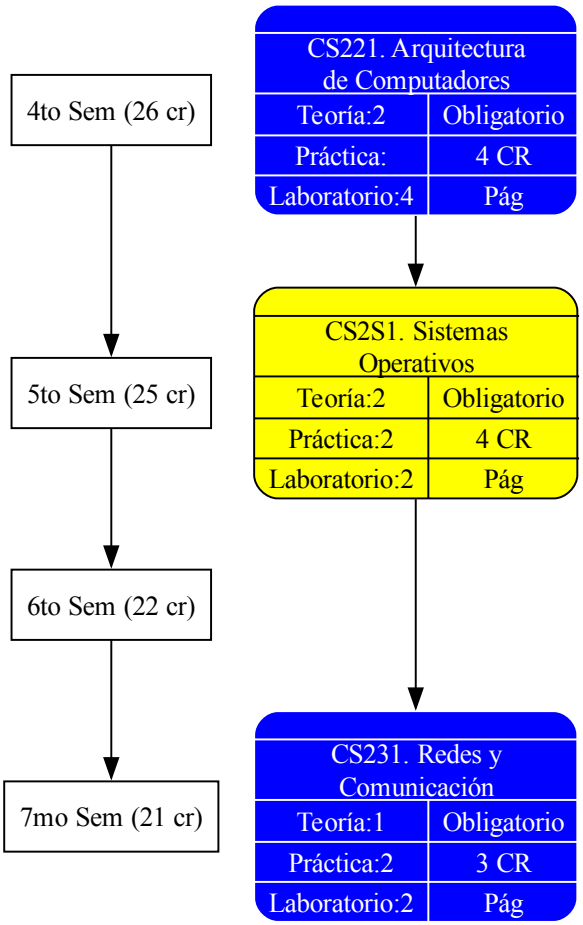


Figura 3.31: Mapa de Conexión. CS2S1 Sistemas Operativos

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*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Rol y Propósito de los Sistemas Operativos, 8 hr(s)

Competencias: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI162 Política y gobernanza de la IA (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**: Ninguno

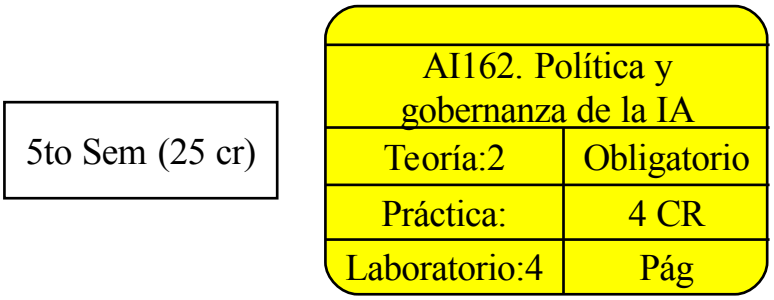


Figura 3.32: Mapa de Conexión. AI162 Política y gobernanza de la IA

2. Fundamentación

Este curso examina el impacto de la Inteligencia Artificial en la sociedad, las políticas públicas y los marcos regulatorios. Explora los desafíos éticos, legales y sociales asociados al despliegue de IA, preparando a los estudiantes para participar en la creación de políticas responsables.

3. Objetivos del curso.

- Analizar el impacto socioeconómico de la IA en distintos contextos globales
- Evaluar críticamente los marcos regulatorios existentes y propuestos para IA
- Desarrollar propuestas de políticas públicas para el uso ético de IA

4. Competencias Generales (*Outcomes*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Usar)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

5. Contenido

5.1 Fundamentos de Políticas de IA, 12 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: **I:** Historia de la regulación tecnológica **II:** Conceptos clave: transparencia, rendición de cuentas, equidad **III:** Casos de estudio: GDPR (UE), Ley de IA (España), enfoques en EE.UU. y Asia

Objetivos de Aprendizaje I: Explicar los principios básicos de gobernanza de IA [Familiarizarse] **II:** Comparar marcos regulatorios internacionales [Usar]

Bibliografía: [Cath et al., 2020, Fjeld et al., 2020]

5.2 Impacto Social y Económico, 12 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Sesgos algorítmicos y discriminación sistémica **II:** Automatización y futuro del trabajo **III:** Brechas digitales y acceso a tecnologías de IA

Objetivos de Aprendizaje I: Evaluar cómo la IA amplifica desigualdades sociales [Evaluar] **II:** Proponer soluciones para mitigar impactos negativos [Usar]

Bibliografía: [Zuboff, 2019, Eubanks, 2018]

5.3 Diseño de Políticas Públicas, 18 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Metodologías para evaluación de riesgos en sistemas de IA **II:** Participación multiactor: gobiernos, empresas, sociedad civil **III:** Herramientas para auditoría algorítmica

Objetivos de Aprendizaje I: Diseñar un marco de políticas para un caso real [Evaluar] **II:** Argumentar propuestas con evidencia técnica y ética [Usar]

Bibliografía: [Crawford, 2021, Jobin et al., 2019]

5.4 Ética y Derechos Humanos, 18 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Derechos digitales y protección de datos **II:** IA militar y dilemas éticos **III:** Recomendaciones de la UNESCO sobre ética en IA

Objetivos de Aprendizaje I: Analizar conflictos entre innovación y derechos humanos [Familiarizarse] **II:** Redactar un código de ética para proyectos de IA [Evaluar]

Bibliografía: [Binns, 2018, UNESCO, 2021]

6. Bibliografía

[Binns, 2018] Binns, R. (2018). *Algorithmic Accountability and Public Reason*. Springer.

[Cath et al., 2020] Cath, C. et al. (2020). *Governing Artificial Intelligence: Ethical, Legal, and Technical Opportunities and Challenges*. Oxford University Press.

[Crawford, 2021] Crawford, K. (2021). *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. Yale University Press.

[Eubanks, 2018] Eubanks, V. (2018). *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin's Press.

[Fjeld et al., 2020] Fjeld, J. et al. (2020). Principled artificial intelligence: Mapping consensus in ethical and rights-based approaches. Technical report, Harvard Berkman Klein Center.

[Jobin et al., 2019] Jobin, A. et al. (2019). The global landscape of AI Ethics guidelines. *Nature Machine Intelligence*, 1:389–399.

[UNESCO, 2021] UNESCO (2021). Recommendation on the ethics of artificial intelligence. Technical report.

[Zuboff, 2019] Zuboff, S. (2019). *The Age of Surveillance Capitalism*. PublicAffairs.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

MA202 Métodos Numéricos (Obligatorio)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - BMA103 Cálculo Integral (2do Sem-Pág 88)

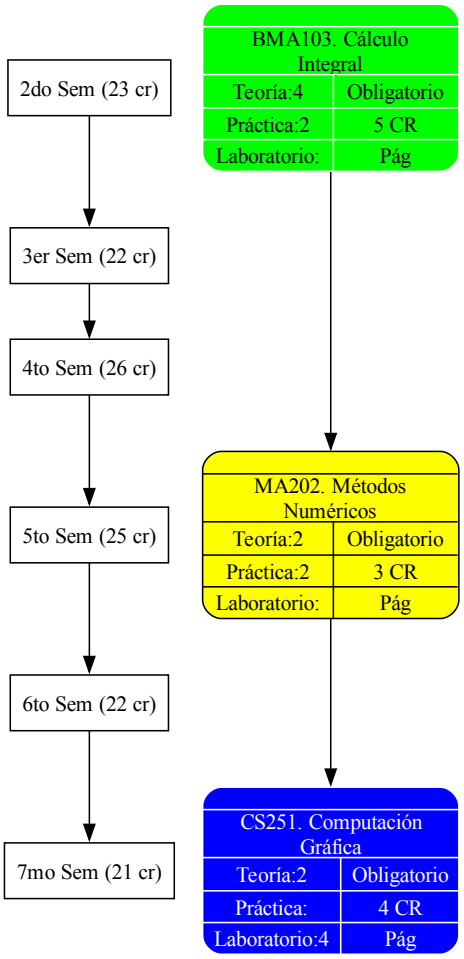


Figura 3.33: Mapa de Conexión. MA202 Métodos Numéricos

2. Fundamentación

Los métodos numéricos son una herramienta fundamental compartida por la informática y las disciplinas de ingeniería para aproximar soluciones a problemas matemáticos que no pueden resolverse analíticamente. Este curso proporciona una introducción a los métodos numéricos más comunes, incluyendo el análisis de errores, la solución de ecuaciones no lineales, la interpolación, la integración numérica y la solución de ecuaciones diferenciales, en contextos científicos y de ingeniería.

3. Objetivos del curso.

- Comprender la importancia de los métodos numéricos en la resolución de problemas científicos y de ingeniería.
- Aplicar diferentes métodos numéricos para aproximar soluciones a problemas matemáticos.
- Analizar la precisión y eficiencia de los métodos numéricos utilizados.

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Análisis de Errores y Aritmética de Punto Flotante, 6 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Representación en punto flotante IEEE 754: épsilon de máquina, desbordamiento, subdesbordamiento y modos de redondeo **II:** Errores de redondeo, cancelación catastrófica y pérdida de significado **III:** Análisis de error directo e inverso; número de condición de un problema **IV:** Estabilidad numérica: algoritmos estables vs. inestables; análisis de error inverso de Wilkinson **V:** Propagación de errores en operaciones aritméticas y evaluación de funciones

Objetivos de Aprendizaje **I:** Explicar el estándar IEEE 754 de punto flotante e identificar las fuentes de errores de redondeo y cancelación [Familiarizarse] **II:** Estimar el número de condición de un problema y predecir la precisión de un resultado numérico [Usar] **III:** Analizar un algoritmo para determinar su estabilidad numérica usando cotas de error directo e inverso [Evaluar]

Bibliografía: [Chapra and Canale, 2015, Burden and Faires, 2010]

5.2 Búsqueda de Raíces de Ecuaciones No Lineales, 10 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Método de bisección: acotamiento del intervalo, criterio de parada y cota del error **II:** Método de Newton-Raphson: derivación, convergencia cuadrática y casos de falla **III:** Método de la secante como alternativa al método de Newton sin derivada explícita **IV:** Orden de convergencia y comparación de la eficiencia entre métodos iterativos **V:** Extensión del método de Newton a sistemas de ecuaciones no lineales mediante la matriz Jacobiana, base de solucionadores en aprendizaje automático

Objetivos de Aprendizaje **I:** Comparar las garantías de convergencia de los métodos de bisección, Newton-Raphson y secante [Familiarizarse] **II:** Implementar el método de Newton-Raphson para aproximar raíces de una función no lineal [Usar] **III:** Analizar el orden de convergencia de un método iterativo y justificar su elección según el costo computacional [Evaluar]

Bibliografía: [Chapra and Canale, 2015, Burden and Faires, 2010]

5.3 Teoría de Aproximación e Interpolación, 10 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Interpolación polinomial: Lagrange, diferencias divididas de Newton y fenómeno de Runge **II:** Interpolación por splines: splines cúbicos, B-splines y métodos de funciones polinómicas a trozos **III:** Mejor aproximación en espacios normados: Chebyshev (minimax) y mínimos cuadrados **IV:** Polinomios de Chebyshev: propiedades, ortogonalidad y convergencia espectral **V:** Aproximación trigonométrica y transformada rápida de Fourier (FFT)

Objetivos de Aprendizaje **I:** Explicar el fenómeno de Runge y justificar la elección de nodos de Chebyshev para mitigarlo [Familiarizarse] **II:** Construir interpolantes por splines cúbicos y de Chebyshev para datos dados y estimar el error de interpolación [Usar] **III:** Aplicar la FFT para calcular eficientemente aproximaciones trigonométricas de una función muestreada [Evaluar]

Bibliografía: [Chapra and Canale, 2015, Burden and Faires, 2010]

5.4 Integración Numérica y Cuadratura, 10 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Reglas de Newton-Cotes: trapezoidal, Simpson y reglas compuestas; análisis de error **II:** Cuadratura adaptativa y control automático del error **III:** Cuadratura gaussiana: nodos y pesos óptimos, conexión con polinomios ortogonales **IV:** Tratamiento numérico de integrales impropias y singulares **V:** Integración de Monte Carlo y métodos cuasi-Monte Carlo para integrales de alta dimensión

Objetivos de Aprendizaje **I:** Derivar la fórmula de error para la regla compuesta de Simpson e identificar su orden de precisión [Familiarizarse] **II:** Seleccionar y aplicar una regla de cuadratura apropiada (Gaussiana, adaptativa) según la regularidad del integrando [Usar] **III:** Aplicar integración de Monte Carlo para estimar integrales de alta dimensión y cuantificar el error estadístico [Evaluar]

Bibliografía: [Chapra and Canale, 2015, Burden and Faires, 2010]

5.5 Métodos Numéricos para Ecuaciones Diferenciales, 12 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Métodos de Runge-Kutta: Euler, RK4 y métodos embebidos para control de errores **II:** Métodos de pasos múltiples: Adams-Bashforth, Adams-Moulton y métodos BDF **III:** Análisis de estabilidad: cero-estabilidad, regiones de estabilidad absoluta y EDOs rígidas **IV:** Métodos de diferencias finitas para EDPs parabólicas y elípticas: estabilidad y convergencia

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 adaptativo del paso y aplicarlo a un sistema de EDOs [Usar] **III:** Analizar la estabilidad y convergencia de un esquema de diferencias finitas para una EDP parabólica [Evaluar]

Bibliografía: [Chapra and Canale, 2015, Burden and Faires, 2010]

5.6 Álgebra Lineal Numérica, 12 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Eliminación gaussiana con pivoteo parcial, factorización LU y análisis de complejidad **II:** Factorización QR vía Householder y Gram-Schmidt; problemas de mínimos cuadrados **III:** Algoritmos de valores propios: iteración de la potencia, algoritmo QR y el método de Lanczos **IV:** Descomposición en valores singulares: cálculo, truncamiento y aplicaciones (ACP,

pseudoinversa) **V:** Métodos iterativos para sistemas dispersos grandes: GC, GMRES y precondicionamiento

Objetivos de Aprendizaje I: Comparar solucionadores directos e iterativos y seleccionar el método adecuado para una estructura matricial dada [Familiarizarse] **II:** Aplicar la factorización QR y la SVD para resolver problemas de mínimos cuadrados y calcular aproximaciones de bajo rango [Usar] **III:** Analizar la estabilidad numérica y el costo computacional de los algoritmos de valores propios [Evaluar]

Bibliografía: [Burden and Faires, 2010, Trefethen and III, 1997]

5.7 Método de Elementos Finitos, 8 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: I: Formulación débil (variacional) de problemas de valor en la frontera y espacios de Sobolev **II:** Método de Galerkin: espacios de aproximación de dimensión finita y matriz de rigidez **III:** Espacios de elementos finitos triangulares y cuadriláteros: elementos de Lagrange y conformidad **IV:** Estimaciones de error a priori: lema de Céa y cotas de error de interpolación **V:** Estimaciones de error a posteriori y refinamiento adaptativo de malla

Objetivos de Aprendizaje I: Derivar la formulación débil de un PVE elíptico y mostrar su equivalencia con la forma fuerte [Familiarizarse] **II:** Ensamblar la matriz de rigidez global y el vector de carga para una discretización lineal por elementos finitos [Usar] **III:** Estimar el error en H^1 de una solución por elementos finitos usando el lema de Céa y la teoría de interpolación [Evaluar]

Bibliografía: [Reddy, 2019, Burden and Faires, 2010]

5.8 Algoritmos de Optimización, 4 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Descenso de gradiente y métodos de búsqueda lineal: condiciones de Armijo-Wolfe y tasas de convergencia **II:** Método de Newton y métodos cuasi-Newton (BFGS, L-BFGS) **III:** Optimización con restricciones: condiciones KKT, métodos de penalización y programación cuadrática secuencial **IV:** Algoritmos de optimización convexa: métodos de punto interior y ADMM **V:** Descenso de gradiente estocástico (SGD), reducción de varianza y el optimizador Adam

Objetivos de Aprendizaje I: Explicar las garantías de convergencia del descenso de gradiente para funciones convexas suaves [Familiarizarse] **II:** Aplicar los métodos de Newton y cuasi-Newton a problemas de optimización sin restricciones [Usar] **III:** Formular un problema de optimización con restricciones como un sistema KKT y aplicar un método de punto interior [Evaluar]

Bibliografía: [Nocedal and Wright, 2006]

5.9 Computación Paralela y de Alto Rendimiento, 4 hr(s)

Competencias: AG-C11

Tópicos: I: Arquitecturas paralelas: memoria compartida (OpenMP), memoria distribuida (MPI) y GPU (CUDA) **II:** Modelos de rendimiento: ley de Amdahl, modelo de línea techada y complejidad de comunicación **III:** Algoritmos paralelos para álgebra lineal densa y dispersa (ScaLAPACK, PETSc) **IV:** Descomposición de dominio y solucionadores paralelos de EDPs **V:** Diferenciación automática (modo directo y reverso) y su rol en ML científico

Objetivos de Aprendizaje I: Explicar la ley de Amdahl e identificar los cuellos de botella que limitan la aceleración paralela en un algoritmo dado [Familiarizarse] **II:** Implementar una rutina paralela de álgebra lineal usando MPI o OpenMP y medir la eficiencia paralela [Usar] **III:** Aplicar diferenciación automática para calcular gradientes de funciones científicas complejas con fines de optimización [Evaluar]

Bibliografía: [Pacheco and Malensek, 2022]

5.10 Teoría de Grafos, 4 hr(s)

Competencias: AG-C08

Tópicos: **I:** Grafos: definiciones, isomorfismo, sucesiones de grados, árboles y árboles de expansión **II:** Conectividad, Teorema de Menger y flujos en redes (flujo máximo-corte mínimo) **III:** Emparejamientos (Teorema de Hall), coloraciones de grafos y el polinomio cromático **IV:** Grafos planos, fórmula de Euler, Teorema de Kuratowski y el teorema de los cuatro colores **V:** Matrices de adyacencia y laplaciana, valores propios y grafos expansores

Objetivos de Aprendizaje **I:** Identificar propiedades estructurales de grafos (conectividad, planaridad, bipartición) y aplicar la fórmula de Euler [Familiarizarse] **II:** Aplicar el Teorema de Hall y algoritmos de flujo en redes a problemas de emparejamiento y enrutamiento [Usar] **III:** Analizar el espectro de un grafo para acotar su número cromático y sus propiedades de conectividad [Evaluar]

Bibliografía: [West, 2001]

5.11 Análisis de Series Temporales, 2 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: **I:** Estacionariedad, función de autocovarianza y función de autocorrelación (ACF) **II:** Modelos ARMA: identificación, estimación (Yule-Walker, MLE) y diagnóstico **III:** Densidad espectral, el periodograma y el teorema de Wiener-Khinchin **IV:** Modelos de espacio de estados y el filtro de Kalman **V:** Modelos ARIMA, ajuste estacional y predicción a múltiples pasos

Objetivos de Aprendizaje **I:** Identificar la estacionariedad y determinar el orden de un modelo ARMA a partir de gráficos ACF y PACF [Familiarizarse] **II:** Ajustar modelos ARIMA a datos de series temporales, validar residuos y producir predicciones [Usar] **III:** Aplicar el filtro de Kalman para estimar estados ocultos en un modelo lineal gaussiano de espacio de estados [Evaluar]

Bibliografía: [Shumway and Stoffer, 2017]

6. Bibliografía

[Burden and Faires, 2010] Burden, R. L. and Faires, J. D. (2010). *Numerical Analysis*. Cengage Learning.

[Chapra and Canale, 2015] Chapra, S. C. and Canale, R. P. (2015). *Numerical Methods for Engineers*. McGraw-Hill Education.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition.

[Pacheco and Malensek, 2022] Pacheco, P. S. and Malensek, M. (2022). *An Introduction to Parallel Programming*. Morgan Kaufmann, 2nd edition.

[Reddy, 2019] Reddy, J. N. (2019). *An Introduction to the Finite Element Method*. McGraw-Hill Education, 4th edition.

[Shumway and Stoffer, 2017] Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, 4th edition.

[Trefethen and III, 1997] Trefethen, L. N. and III, D. B. (1997). *Numerical Linear Algebra*. SIAM.

[West, 2001] West, D. B. (2001). *Introduction to Graph Theory*. Prentice Hall, 2nd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial
BEI203 Inglés V (Electivo)

1. Información general

- **Semestre**: 5to Sem. **Créditos**: 2
- **Horas del curso**: **Práctica**: 4 horas;
- **Prerrequisitos**:
  - BEI202 Inglés IV (4to Sem-Pág 162)

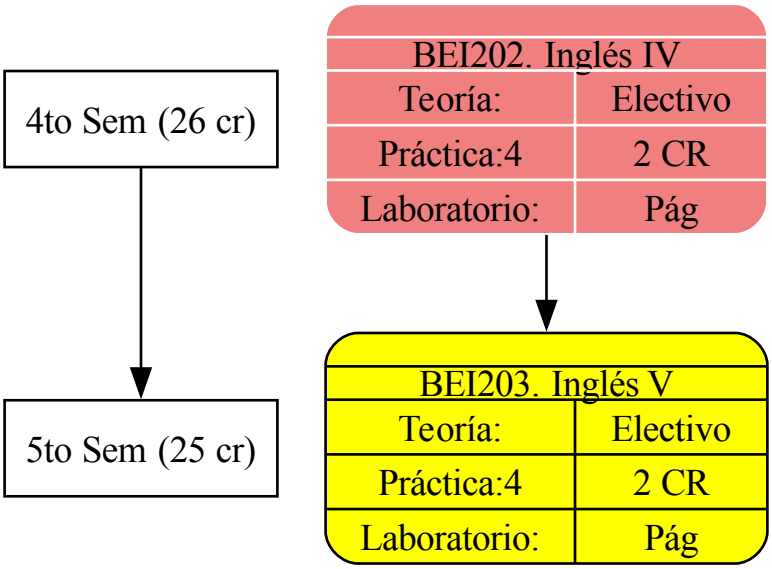


Figura 3.34: Mapa de Conexión. BEI203 Inglés V

2. Fundamentación

Una parte fundamental de la formación integral de un profesional es la capacidad de comunicarse en un idioma extranjero además del propio idioma nativo. En la carrera de ingeniería y computación, el inglés técnico es la herramienta más crítica para acceder a la vanguardia del conocimiento y la investigación, redactar artículos científicos, participar en conferencias internacionales y colaborar en equipos globales. Este curso lleva al estudiante al nivel de comunicador técnico autónomo, capaz de producir y evaluar críticamente contenido académico y profesional.

3. Objetivos del curso.

- Incrementar la capacidad y fluidez para comunicar ideas técnicas complejas en inglés.
- Lograr que los estudiantes interactúen con énfasis en la argumentación científica y la redacción técnica avanzada.
- Dominar el vocabulario de ingeniería de software, ciencia de datos e investigación de sistemas.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Familiarizarse)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Argumentación Científica, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Repaso avanzado de verbos auxiliares y tiempos verbales en escritura académica. **II:** Preguntas negativas y respuestas enfáticas en el debate científico. **III:** Formación de palabras: Prefijos y sufijos comunes en computación y ciencias. **IV:** Corrección de afirmaciones erróneas: Refutación de argumentos en discusiones técnicas.

Objetivos de Aprendizaje **I:** Al finalizar la unidad, el estudiante utiliza auxiliares y tiempos verbales con precisión para construir argumentos científicos, cuestionar afirmaciones técnicas erróneas y participar en debate académico estructurado.

Bibliografía: [? line:48, ? line:48, ? line:48, ? line:48, ? line:48]

5.2 Metodología de Investigación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Presente Simple vs. Presente Continuo: Distinción entre hechos generales y experimentos en curso. **II:** Verbos de estado (State verbs) y verbos de acción en descripciones experimentales. **III:** Vocabulario: Diseño de investigación, variables experimentales, métodos de recolección de datos. **IV:** Redacción: Descripción de un método de investigación o procedimiento experimental.

Objetivos de Aprendizaje **I:** El estudiante diferencia entre usos estáticos y progresivos para describir con precisión leyes científicas, experimentos en curso y metodologías de investigación en computación e ingeniería.

Bibliografía: [? line:61, ? line:61, ? line:61, ? line:61, ? line:61]

5.3 Estudios de Caso, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Tiempos narrativos: Pasado Simple, Pasado Continuo y Pasado Perfecto en análisis de casos. **II:** Estructura de Üsed toz "Would"para describir sistemas heredados y prácticas pasadas. **III:** Vocabulario: Casos históricos de computación, cambios de paradigma, lecciones aprendidas. **IV:** Redacción: Informe estructurado de estudio de caso sobre un incidente o innovación computacional.

Objetivos de Aprendizaje **I:** El estudiante analiza y narra secuencias de eventos técnicos y casos históricos de computación con fluidez, aplicando correctamente los tiempos narrativos en informes profesionales de estudios de caso.

Bibliografía: [? line:74, ? line:74, ? line:74, ? line:74, ? line:74]

5.4 Propuestas de Investigación, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Predicciones y planes: Repaso del futuro para la redacción de propuestas académicas. **II:** Futuro Continuo y Futuro Perfecto en el alcance del proyecto y los resultados esperados. **III:** Vocabulario: Objetivos de investigación, contribuciones esperadas, alcance y limitaciones. **IV:** Redacción: Propuesta de investigación corta o abstract de proyecto técnico.

Objetivos de Aprendizaje I: El estudiante proyecta metas de investigación y resultados anticipados, y realiza predicciones basadas en evidencia sobre la evolución de la tecnología utilizando formas de futuro apropiadas en escritura académica formal.

Bibliografía: [? line:87, ? line:87, ? line:87, ? line:87, ? line:87]

5.5 Documentación Técnica, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Patrones verbales (Verbos seguidos de -ing o infinitivo) en escritura instruccional. **II:** Expresiones con "Like"para analogías y comparaciones técnicas. **III:** Vocabulario: Documentación de APIs, manuales de usuario, especificaciones técnicas. **IV:** Debates: Estándares de documentación de código abierto y reproducibilidad.

Objetivos de Aprendizaje I: El estudiante utiliza patrones verbales complejos para redactar documentación técnica clara y precisa, y emplea el razonamiento analógico para explicar conceptos técnicos a audiencias diversas.

Bibliografía: [? line:100, ? line:100, ? line:100, ? line:100, ? line:100]

5.6 Proyectos a Largo Plazo, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Presente Perfecto Continuo vs. Presente Perfecto Simple en informes de avance de investigación. **II:** Verbos modales de deducción (Presente y Pasado) para interpretar resultados experimentales. **III:** Vocabulario: Sistemas de largo plazo, programas de investigación sostenidos, desarrollo de startups. **IV:** Presentaciones orales: Ïnforme de avance de un proyecto de investigación en computación".

Objetivos de Aprendizaje I: El estudiante discute actividades de investigación y desarrollo a largo plazo, realiza deducciones sobre resultados técnicos y presenta informes de avance de proyectos computacionales con el registro académico apropiado.

Bibliografía: [? line:113, ? line:113, ? line:113, ? line:113, ? line:113]

5.7 Redacción Técnica Avanzada, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: I: Oraciones de relativo (Defining vs Non-defining) en la redacción de artículos técnicos. **II:** Voz pasiva avanzada en prosa científica y de ingeniería formal. **III:** Vocabulario: Revisión de pares, factor de impacto, prácticas de citación, acceso abierto. **IV:** Redacción: Abstract e introducción de un artículo de investigación en computación.

Objetivos de Aprendizaje I: El estudiante redacta textos técnicos y científicos precisos y bien estructurados utilizando cláusulas de relativo y construcciones pasivas avanzadas para añadir profundidad y claridad a la escritura de investigación.

Bibliografía: [? line:126, ? line:126, ? line:126, ? line:126, ? line:126]

5.8 Inglés para Ingeniería de Software, 12 hr(s)

Competencias: AG-C03,AG-C04

Tópicos: **I:** Metodologías Ágiles: Scrum, Kanban y XP — terminología en inglés. **II:** QA (Quality Assurance) y Testing: Planes de prueba, reportes de bugs, criterios de aceptación. **III:** Documentación de requerimientos de software: historias de usuario, casos de uso, especificaciones. **IV:** Simulación: Reunión técnica de equipo (Sprint planning y retrospectiva).

Objetivos de Aprendizaje **I:** Al finalizar el curso, el estudiante domina la terminología de ingeniería de software y gestión de proyectos, puede producir documentos técnicos profesionales y participar activamente en reuniones y revisiones técnicas internacionales en inglés.

Bibliografía: [**?** line:139 , **?** line:139 , **?** line:139 , **?** line:139 , **?** line:139]

6. Bibliografía

line:144 line:144 line:144 line:144 line:144

Sexto Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS311 Algoritmos para Problemas Complejos (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)

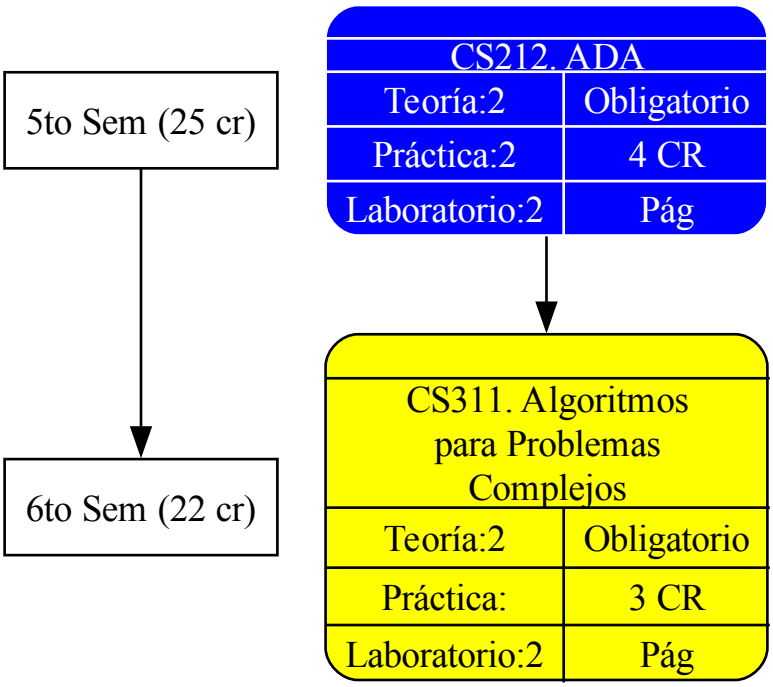


Figura 3.35: Mapa de Conexión. CS311 Algoritmos para Problemas Complejos

2. Fundamentación

La Programación Competitiva entrena a los estudiantes para resolver problemas algorítmicos complejos bajo restricciones de tiempo y recursos. Este curso cierra la brecha entre el diseño algorítmico teórico y la implementación de alto rendimiento. Cubre técnicas avanzadas en programación dinámica, teoría de grafos y matemáticas, fomentando el pensamiento analítico requerido para entrevistas de ingeniería de software de primer nivel y concursos de programación internacionales como el ICPC.

3. Objetivos del curso.

- Dominar técnicas y paradigmas avanzados de programación competitiva.
- Desarrollar la capacidad de analizar rápidamente la complejidad de soluciones potenciales.
- Implementar estructuras de datos eficientes para consultas de rango y actualizaciones dinámicas.

- Aplicar algoritmos matemáticos y geométricos especializados para resolver problemas de concursos.
- Optimizar código para velocidad de ejecución y uso de memoria en un entorno competitivo.

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Paradigmas de resolución de problemas, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Búsqueda Completa: Retroceso Recursivo y Máscaras de bits. **II:** Estrategias Voraces: Propiedad de elección y subestructura óptima. **III:** Divide y Vencerás: Búsqueda Binaria sobre la respuesta y Búsqueda Ternaria. **IV:** Programación Dinámica: Reducción de estado, DP de dígitos y DP en árboles.

Objetivos de Aprendizaje I: Aplicar técnicas de poda de búsqueda para resolver problemas NP-difíciles con restricciones pequeñas [Usar] **II:** Identificar el paradigma más eficiente para un enunciado de problema dado [Evaluar]

Bibliografía: [Halim and Halim, 2020, Laaksonen, 2020]

5.2 Búsquedas por rango avanzadas, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Árboles de Fenwick (Árboles Binarios Indexados). **II:** Árboles de Segmentos: Propagación Diferida y Árboles de Segmentos Persistentes. **III:** Descomposición de Raíz Cuadrada y Algoritmo de Mo. **IV:** Tablas Dispersas para RMQ.

Objetivos de Aprendizaje I: Implementar estructuras de consulta de rango para actualizaciones de datos en tiempo real [Evaluar] **II:** Analizar las compensaciones entre diferentes estructuras de datos de consulta de rango [Usar]

Bibliografía: [Halim and Halim, 2020, Skiena, 2020]

5.3 Algoritmos avanzados de grafos, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Flujo en Redes: Flujo Máximo Corte Mínimo (Dinic y Edmonds-Karp). **II:** Componentes Fuertemente Conexas (Tarjan y Kosaraju). **III:** Emparejamiento Bipartito y Descomposición Pesado-Ligero. **IV:** Descomposición por Centroide en árboles.

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] **V:** 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]

Bibliografía: [Laaksonen, 2020, Halim and Halim, 2020]

5.4 Matemática y Geometría, 12 hr(s)

Competencias: AG-C08,AG-C12

Tópicos: I: Teoría de Números: Criba de Eratóstenes, Euclides Extendido e Inverso Modular. **II:** Combinatoria: Inclusión-Exclusión y Lema de Burnside. **III:** Geometría Computacional: Producto Cruz, Envolvente Convexa (Cadena Monótona) e Intersección de líneas. **IV:** Transformada Rápida de Fourier (FFT) para multiplicación de polinomios.

Objetivos de Aprendizaje I: Resolver problemas combinatorios usando aritmética modular [Usar] **II:** Implementar primitivas geométricas para análisis de relaciones espaciales [Evaluar]

Bibliografía: [Skiena, 2020, Laaksonen, 2020]

6. Bibliografía

[Halim and Halim, 2020] Halim, S. and Halim, F. (2020). *Competitive Programming 4: The Lower Bound of Programming Contests*. Lulu.

[Laaksonen, 2020] Laaksonen, A. (2020). *Competitive Programmer's Handbook*. Draft.

[Skiena, 2020] Skiena, S. S. (2020). *The Algorithm Design Manual*. Springer, 3rd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI262 Procesamiento de Lenguaje Natural (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - STA251 Estadística y Probabilidades (3er Sem-Pág 113)

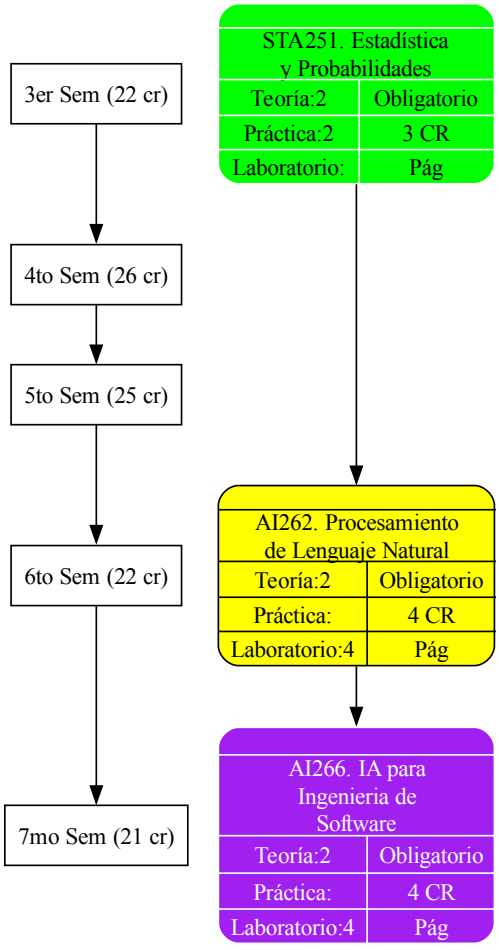


Figura 3.36: Mapa de Conexión. AI262 Procesamiento de Lenguaje Natural

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Fundamentos de NLP, 15 hr(s)

Competencias: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI263 Introducción a Machine Learning (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - STA251 Estadística y Probabilidades (3er Sem-Pág 113)
  - AI161 IA Aplicada (3er Sem-Pág 109)
  - CS210 Algoritmos y Estructuras de Datos (4to Sem-Pág 132)

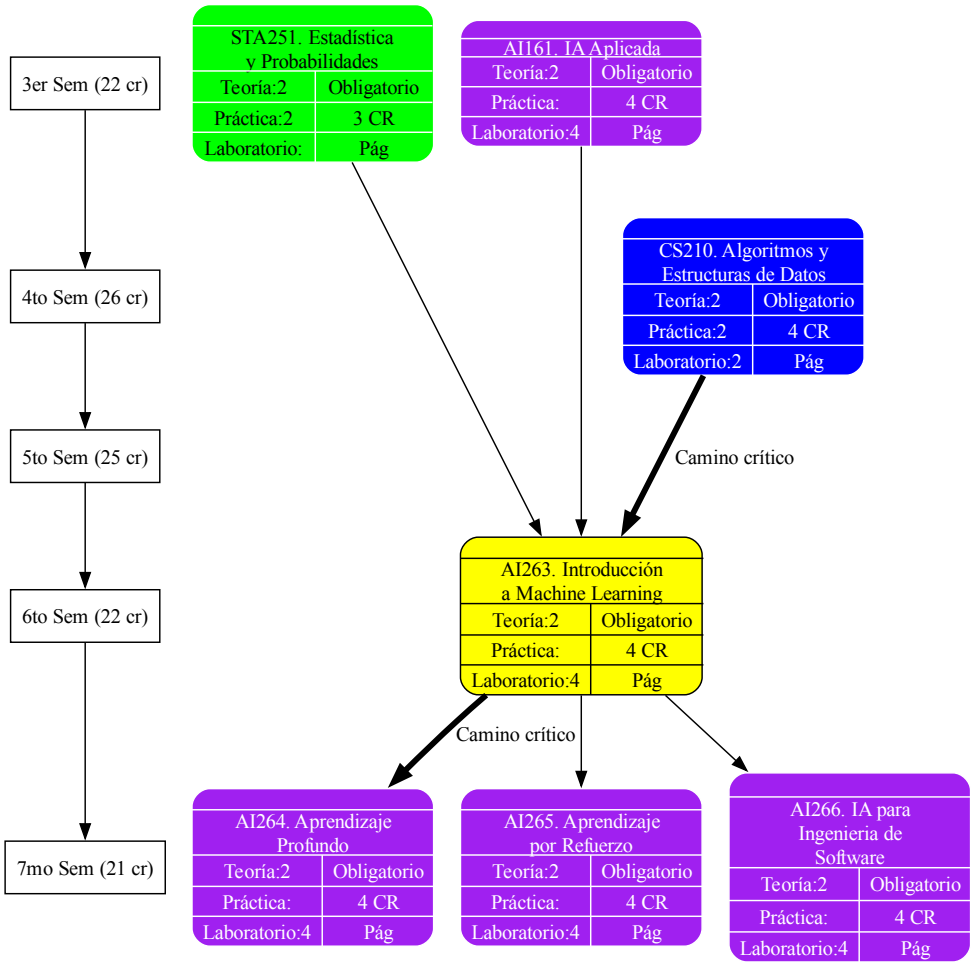


Figura 3.37: Mapa de Conexión. AI263 Introducción a Machine Learning

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*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Fundamentos de ML, 15 hr(s)

Competencias: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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: AG-C08,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI267 Interaccion Humano-IA (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS291 Ingeniería de Software I (5to Sem-Pág 178)

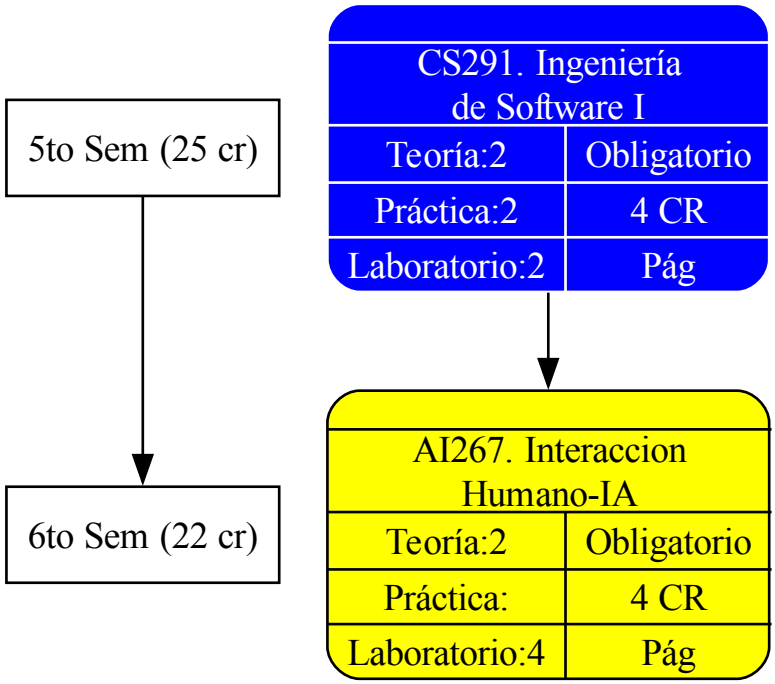


Figura 3.38: Mapa de Conexión. AI267 Interaccion Humano-IA

2. Fundamentación

Este curso se enfoca en el diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano. Más allá de los principios clásicos de HCI, aborda los desafíos emergentes de la interacción con sistemas de inteligencia artificial, agentes inteligentes y software de procesamiento de lenguaje natural. Los estudiantes desarrollarán competencias en diseño centrado en el usuario, accesibilidad, responsabilidad ética y evaluación de interfaces tradicionales y conversacionales basadas en IA, adquiriendo habilidades esenciales para diseñar tecnología responsable y centrada en las personas.

3. Objetivos del curso.

- Comprender las características cognitivas, físicas y emocionales del ser humano relevantes para el diseño de sistemas interactivos.

- Aplicar principios de responsabilidad ética y diseño centrado en valores, incluyendo equidad algorítmica y sesgo.
- Aplicar estándares de accesibilidad y diseño inclusivo para poblaciones y capacidades diversas.
- Ejecutar métodos de evaluación de usabilidad para iterar y mejorar interfaces de usuario.
- Diseñar sistemas interactivos mediante procesos de diseño centrado en el usuario e iterativo, para múltiples plataformas y dominios.
- Analizar las dimensiones sociales, éticas y profesionales de la tecnología interactiva.
- Comprender arquitecturas de agentes y aplicar principios de interacción humano-agente, incluyendo asistentes de IA y sistemas autónomos.
- Aplicar técnicas de procesamiento de lenguaje natural para diseñar y evaluar interfaces humano-software basadas en modelos de lenguaje.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Comprender al Usuario: Objetivos individuales e interacciones con otros, 8 hr(s)

Competencias: AG-C03,AG-C09

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

Objetivos de Aprendizaje **I:** Describir las capacidades y limitaciones humanas que afectan el diseño de interacción [Describir] **II:** Analizar cómo las características del usuario influyen en la interacción con sistemas informáticos [Analizar] **III:** Identificar objetivos y tareas del usuario en contextos de uso específicos [Analizar] **IV:** Explicar factores sociales y organizacionales que impactan la adopción y uso de la tecnología [Explicar] **V:** Aplicar modelos cognitivos para predecir y explicar el comportamiento del usuario [Aplicar] **VI:** Considerar factores emocionales y afectivos en el diseño de interacción [Analizar] **VII:** Realizar investigación de usuarios usando métodos apropiados como entrevistas u observaciones [Evaluar]

Bibliografía: [Norman, 2013, Johnson, 2020]

5.2 Responsabilidad y Rendición de Cuentas en el Diseño, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Consideraciones éticas en HCI: equidad, transparencia, rendición de cuentas **II:** Diseño para valores y diseño sensible a valores **III:** Innovación responsable y ética del diseño **IV:** Rendición de cuentas algorítmica y sesgo en sistemas interactivos **V:** Responsabilidad profesional en la práctica de HCI **VI:** Consideraciones legales y regulatorias para sistemas interactivos

Objetivos de Aprendizaje **I:** Identificar problemas éticos en el diseño e implementación de HCI [Analizar] **II:** Aplicar enfoques de diseño sensible a valores al desarrollo tecnológico [Aplicar] **III:** Discutir prácticas de innovación responsable en HCI [Debatir] **IV:** Analizar la rendición de cuentas algorítmica y el sesgo en sistemas interactivos [Analizar] **V:** Describir responsabilidades profesionales en la práctica de HCI [Describir] **VI:** Explicar consideraciones legales y regulatorias relevantes para sistemas interactivos [Explicar]

Bibliografía: [Shneiderman, 2022, Norman, 2013]

5.3 Accesibilidad y Diseño Inclusivo, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Principios y pautas de accesibilidad **II:** Discapacidades y tecnologías de asistencia **III:** Diseño inclusivo y principios de diseño universal **IV:** Métodos de evaluación para accesibilidad **V:** Diseñar para diversas capacidades y contextos **VI:** Estándares de accesibilidad y cumplimiento

Objetivos de Aprendizaje **I:** Aplicar principios de accesibilidad al diseño de sistemas interactivos [Aplicar] **II:** Describir cómo diferentes discapacidades afectan la interacción con la tecnología [Describir] **III:** Diseñar sistemas inclusivos usando principios de diseño universal [Diseñar] **IV:** Realizar evaluaciones de accesibilidad de sistemas interactivos [Evaluar] **V:** Crear diseños que acomoden diversas capacidades y contextos de uso [Crear] **VI:** Garantizar el cumplimiento con estándares y regulaciones de accesibilidad [Evaluar]

Bibliografía: [Johnson, 2020]

5.4 Evaluación del Diseño, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Principios y heurísticas de usabilidad **II:** Métodos de evaluación: evaluación heurística, recorrido cognitivo (cognitive walkthrough) **III:** Pruebas con usuarios: planificación, ejecución, análisis **IV:** Métodos avanzados de evaluación: experimentos controlados, estudios de campo **V:** Métricas y medición de UX **VI:** Métodos de evaluación analítica: modelado y simulación

Objetivos de Aprendizaje **I:** Aplicar principios y heurísticas de usabilidad para evaluar diseños [Aplicar] **II:** Realizar evaluaciones heurísticas y recorridos cognitivos [Evaluar] **III:** Planificar, conducir y analizar sesiones de pruebas con usuarios [Planear] **IV:** Diseñar y conducir experimentos controlados para investigación en HCI [Diseñar] **V:** Medir la experiencia de usuario usando métricas apropiadas [Evaluar] **VI:** Usar métodos analíticos para modelar y predecir el desempeño del usuario [Usar]

Bibliografía: [Norman, 2013, Johnson, 2020]

5.5 Diseño de Sistemas, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Proceso de diseño: diseño centrado en el usuario, diseño iterativo **II:** Diseño de interacción: conceptos, principios, patrones **III:** Técnicas de prototipado: baja fidelidad, alta fidelidad, interactivo **IV:** Arquitectura de información y diseño de navegación **V:** Técnicas avanzadas de interacción: gestos, voz, multimodal **VI:** Diseño para dominios específicos: móvil, web, computación ubicua **VII:** Sistemas de computación colaborativa y social

Objetivos de Aprendizaje **I:** Aplicar procesos de diseño centrado en el usuario para desarrollar sistemas interactivos [Aplicar] **II:** Diseñar interacciones usando principios y patrones establecidos [Diseñar] **III:** Crear prototipos en diferentes niveles de fidelidad para exploración de diseño [Crear] **IV:** Estructurar información y navegación para una interacción efectiva del usuario [Estructurar] **V:** Implementar técnicas avanzadas de interacción como interfaces de gestos o voz [Implementar] **VI:** Diseñar sistemas interactivos para dominios y plataformas específicos [Diseñar] **VII:** Diseñar sistemas colaborativos que apoyen la interacción grupal [Diseñar]

Bibliografía: [Johnson, 2020, Norman, 2013]

5.6 Sociedad, ética y Profesión, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Impacto social de las tecnologías interactivas **II:** Problemas éticos en la investigación y práctica de HCI **III:** Práctica profesional en HCI **IV:** Consideraciones culturales e interculturales en HCI **V:** Sostenibilidad y HCI **VI:** Tendencias futuras y desafíos en HCI

Objetivos de Aprendizaje **I:** Analizar el impacto social de las tecnologías interactivas en individuos y sociedad [Analizar] **II:** Abordar problemas éticos en proyectos de investigación y diseño de HCI [Evaluar] **III:** Describir prácticas profesionales y trayectorias profesionales en HCI [Describir] **IV:** Diseñar sistemas que tomen en cuenta diferencias y contextos culturales [Diseñar] **V:** Incorporar consideraciones de sostenibilidad en el diseño de HCI [Integrar] **VI:** Discutir tendencias emergentes y desafíos futuros en HCI [Debatir]

Bibliografía: [Shneiderman, 2022, Norman, 2013]

5.7 Agentes y Sistemas Cognitivos, 8 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Arquitecturas de agentes (por ejemplo, reactivas, en capas, cognitivas) **II:** Teoría de agentes (incluyendo formalismos matemáticos) **III:** Racionalidad, Teoría de juegos: *a*) Agentes decisorios-teóricos *b*) Procesos de decisión de Markov (MDP) *c*) Algoritmos de bandidos **IV:** Agentes de software, asistentes personales y acceso a información: *a*) Agentes colaborativos *b*) Agentes de recolección de información *c*) Agentes creíbles (personajes sintéticos, modelado de emociones en agentes) **V:** Agentes de aprendizaje **VI:** Sistemas cognitivos: *a*) Arquitecturas cognitivas (por ejemplo, ACT-R, SOAR, ICARUS, FORR) *b*) Capacidades (por ejemplo, percepción, toma de decisiones, predicción, mantenimiento del conocimiento) *c*) Representación, organización, utilización, adquisición y refinamiento del conocimiento *d*) Aplicaciones y evaluación de sistemas cognitivos **VII:** Sistemas multiagente: *a*) Agentes colaboradores *b*) Equipos de agentes *c*) Agentes competitivos (por ejemplo, subastas, votación) *d*) Sistemas de enjambre y modelos biológicamente inspirados *e*) Aprendizaje multiagente **VIII:** Interacción humano-agente: *a*) Metodologías de comunicación (verbal y no verbal) *b*) Problemas prácticos *c*) Aplicaciones: 1) Agentes comerciales, gestión de cadena de suministro 2) Problemas éticos de las interacciones de IA con humanos 3) Regulación y requisitos legales de los sistemas de IA para interactuar con humanos

Objetivos de Aprendizaje **I:** Caracterizar y contrastar las arquitecturas estándar de agentes [Evaluar] **II:** 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] **III:** Describir los paradigmas principales utilizados por los agentes de aprendizaje [Describir] **IV:** Demostrar usando ejemplos apropiados cómo los sistemas multiagente soportan la interacción de agentes [Demostrar] **V:** Construir un agente inteligente usando una arquitectura cognitiva establecida (ACT-R, SOAR) para resolver un problema específico [Aplicar]

Bibliografía: [Russell and Norvig, 2021, Shneiderman, 2022]

5.8 Procesamiento del Lenguaje Natural, 8 hr(s)

Competencias: AG-C03,AG-C09

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: [Jurafsky and Martin, 2024, Russell and Norvig, 2021]

6. Bibliografía

[Johnson, 2020] Johnson, J. (2020). *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Morgan Kaufmann, 3rd edition.

[Jurafsky and Martin, 2024] Jurafsky, D. and Martin, J. H. (2024). *Speech and Language Processing*. Prentice Hall, 3rd edition.

[Norman, 2013] Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2nd edition.

[Russell and Norvig, 2021] Russell, S. and Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition.

[Shneiderman, 2022] Shneiderman, B. (2022). *Human-Centered AI*. Oxford University Press, 1st edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

MA307 Matemática aplicada a la computación (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)

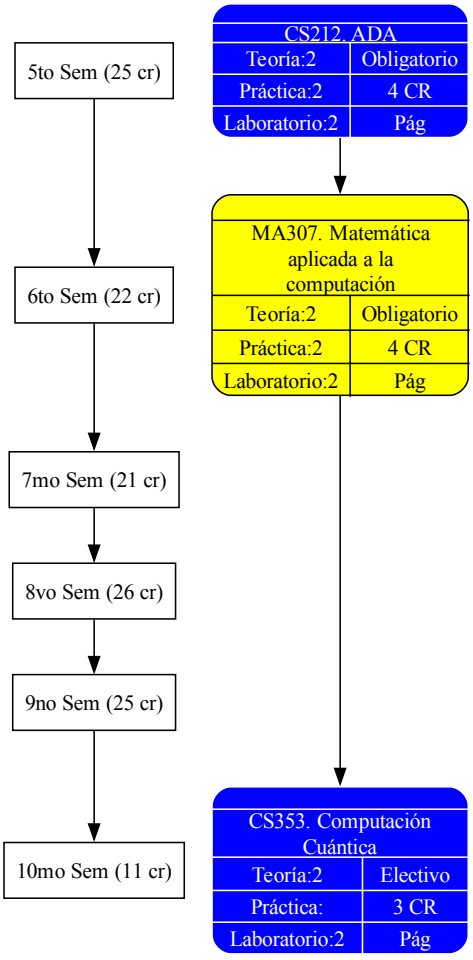


Figura 3.39: Mapa de Conexión. MA307 Matemática aplicada a la computación

2. Fundamentación

Este curso es importante porque desarrolla tópicos del Álgebra Lineal y de Ecuaciones Diferenciales Ordinarias útiles en todas aquellas áreas de la ciencia de la computación donde se trabaja con sistemas lineales y sistemas dinámicos.

3. Objetivos del curso.

- Que el alumno tenga la base matemática para el modelamiento de sistemas lineales y sistemas dinámicos necesarios en el Área de Computación Gráfica e Inteligencia Artificial.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Evaluar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Evaluar)

5. Contenido

5.1 Espacios Lineales, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Espacios vectoriales. **II:** Independencia, base y dimensión. **III:** Dimensiones y ortogonalidad de los cuatro subespacios. **IV:** Aproximaciones por mínimos cuadrados. **V:** Proyecciones **VI:** Bases ortogonales y Gram-Schmidt

Objetivos de Aprendizaje **I:** Identificar espacios generados por vectores linealmente independientes[Usar] **II:** Construir conjuntos de vectores ortogonales[Usar] **III:** Aproximar funciones por polinomios trigonométricos[Usar]

Bibliografía: [Gilbert, 2003, Apóstol, 1973]

5.2 Transformaciones lineales, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Concepto de transformación lineal. **II:** Matriz de una transformación lineal. **III:** Cambio de base. **IV:** Diagonalización y pseudoinversa

Objetivos de Aprendizaje **I:** Determinar el núcleo y la imagen de una transformación[Usar] **II:** Construir la matriz de una transformación[Usar] **III:** Determinar la matriz de cambio de base[Usar]

Bibliografía: [Gilbert, 2003, Apóstol, 1973]

5.3 Autovalores y autovectores, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Diagonalización de una matriz **II:** Matrices simétricas **III:** Matrices definidas positivas **IV:** Matrices similares **V:** La descomposición de valor singular

Objetivos de Aprendizaje **I:** Encontrar la representación diagonal de una matriz[Usar] **II:** Determinar la similaridad entre matrices[Usar] **III:** Reducir una forma cuadrática real a diagonal[Usar]

Bibliografía: [Gilbert, 2003, Apóstol, 1973]

5.4 Sistemas de ecuaciones diferenciales, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Exponencial de una matriz **II:** Teoremas de existencia y unicidad para sistemas lineales homogéneos con coeficientes constantes **III:** Sistemas lineales no homogéneas con coeficientes constantes.

Objetivos de Aprendizaje **I:** Hallar la solución general de un sistema lineal no homogéneo[Usar] **II:** Resolver problemas donde intervengan sistemas de ecuaciones diferenciales[Usar]

Bibliografía: [Zill, 2002, Apóstol, 1973]

5.5 Teoría fundamental, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Sistemas dinámicos **II:** El teorema fundamental **III:** Existencia y unicidad **IV:** El flujo de una ecuación diferencial

Objetivos de Aprendizaje **I:** Discutir la existencia y la unicidad de una ecuación diferencial[Usar] **II:** Analizar la continuidad de las soluciones[Usar] **III:** Estudiar la prolongación de las soluciones [Usar]

Bibliografía: [Hirsh and Smale, 1974]

5.6 Estabilidad de equilibrio, 0 hr(s)

Competencias: AG-C11

Tópicos: **I:** Estabilidad **II:** Funciones de Liapunov **III:** Sistemas gradientes

Objetivos de Aprendizaje **I:** Analizar la estabilidad de una solución[Usar] **II:** Hallar la función de Liapunov para puntos de equilibrio[Usar] **III:** Trazar el retrato de fase un flujo gradiente[Usar]

Bibliografía: [Zill, 2002, Hirsh and Smale, 1974]

6. Bibliografía

[Apóstol, 1973] Apóstol, T. M. (1973). *Calculus Vol II.* Editorial Reverté.

[Gilbert, 2003] Gilbert, S. (2003). *Introduction to Linear Algebra, 3rd edición.* Wellesley-Cambridge Press.

[Hirsh and Smale, 1974] Hirsh, M. W. and Smale, S. (1974). *Differential Equatons, Dynamical Systems, and Linear Álgebra.* Academia Press.

[Zill, 2002] Zill, D. G. (2002). *Ecuaciones Diferenciales con Problemas de Valores en la Frontera.* Thomson Learning.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

FG211 Ética Profesional (Obligatorio)

1. Información general

- **Semestre**: 6to Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

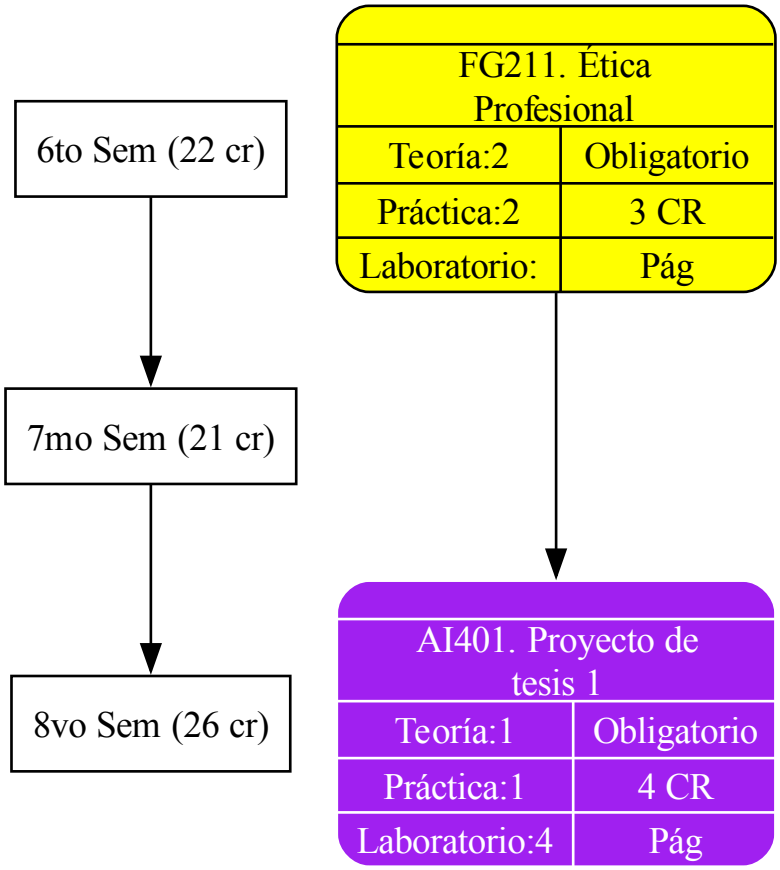


Figura 3.40: Mapa de Conexión. FG211 Ética Profesional

2. Fundamentación

Este curso adapta los principios éticos profesionales al campo de la Inteligencia Artificial, integrando los códigos de ética de ACM e IEEE con desafíos específicos de IA. Combina fundamentos regulatorios (GDPR) con casos actuales de impacto social.

3. Objetivos del curso.

- Aplicar los códigos de ética de ACM/IEEE a problemas de IA.
- Analizar el cumplimiento de regulaciones (GDPR) en sistemas autónomos.
- Diseñar protocolos éticos para proyectos de IA.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Usar)

5. Contenido

5.1 Fundamentos Éticos en IA, 10 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Comparación: Códigos de ética ACM vs IEEE para IA. **II:** Principios de transparencia y justicia algorítmica.

Objetivos de Aprendizaje I: Explicar diferencias entre marcos éticos [Familiarizarse]. **II:** Evaluar compliance en casos reales [Evaluar].

Bibliografía: [ACM, 2018, IEEE, 2020]

5.2 Privacidad y Regulaciones, 14 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: GDPR aplicado a ML: derecho a explicación. **II:** Auditoría ética de datasets (ej: ImageNet).

Objetivos de Aprendizaje I: Implementar checklist GDPR para modelos [Usar]. **II:** Identificar violaciones en estudios de caso [Evaluar].

Bibliografía: [Voigt and von dem Bussche, 2018, Müller, 2021]

5.3 Sesgos y Equidad, 18 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Métricas de equidad (paridad demográfica, igualdad de oportunidades). **II:** Herramientas: IBM Fairness 360, Google What-If.

Objetivos de Aprendizaje I: Medir sesgos en modelos con Python [Usar]. **II:** Proponer estrategias de mitigación [Evaluar].

Bibliografía: [Force, 2020, Mehrabi et al., 2021]

5.4 Responsabilidad en Sistemas Autónomos, 18 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: I: Atribución legal en fallos de IA (ej: vehículos autónomos). **II:** Normas ISO para IA confiable.

Objetivos de Aprendizaje I: Redactar cláusulas de responsabilidad [Evaluar]. **II:** Analizar conflictos legales [Familiarizarse].

Bibliografía: [Bostrom, 2014, Committee, 2019]

6. Bibliografía

[ACM, 2018] ACM (2018). Acm code of ethics and professional conduct. Technical report, ACM.

[Bostrom, 2014] Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

[Committee, 2019] Committee, I. E. (2019). Ethical dilemmas in engineering. Technical report, IEEE.

[Force, 2020] Force, A. E. T. (2020). Case studies in computing and society. Technical report, ACM.

[IEEE, 2020] IEEE (2020). Ieee code of ethics. Technical report, IEEE.

[Mehrabi et al., 2021] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6).

[Müller, 2021] Müller, V. C. (2021). *Ethics of Artificial Intelligence and Robotics*. Cambridge University Press.

[Voigt and von dem Bussche, 2018] Voigt, P. and von dem Bussche, A. (2018). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer.

Séptimo Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS231 Redes y Comunicación (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS2S1 Sistemas Operativos (5to Sem-Pág 184)

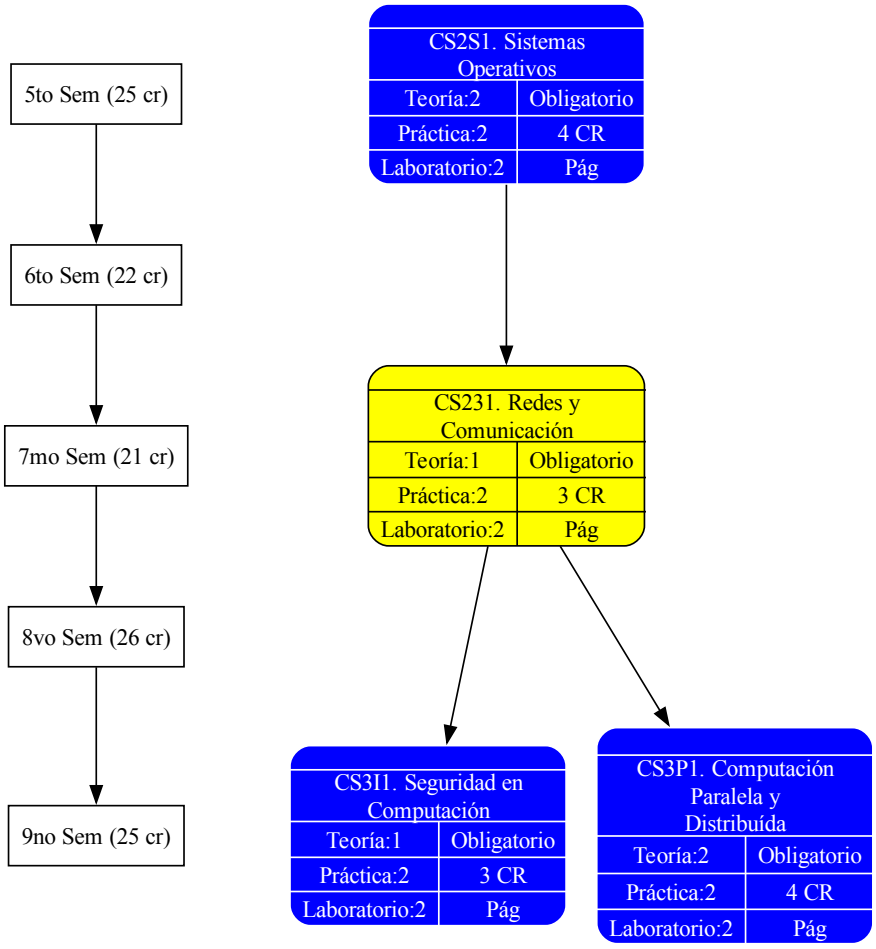


Figura 3.41: Mapa de Conexión. CS231 Redes y Comunicación

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*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Fundamentos de Redes y Comunicaciones, 6 hr(s)

Competencias: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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: AG-C03,AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS251 Computación Gráfica (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - MA202 Métodos Numéricos (5to Sem-Pág 191)

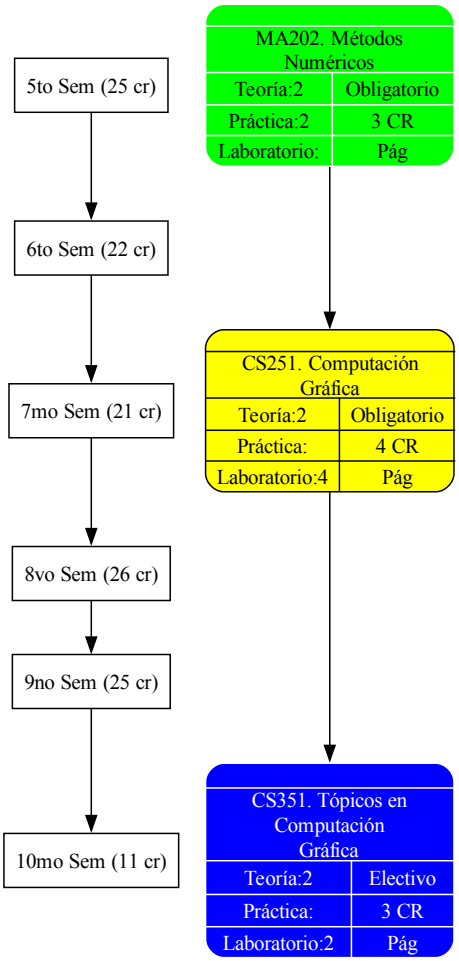


Figura 3.42: Mapa de Conexión. CS251 Computación Gráfica

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*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Usar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Conceptos Fundamentales de Gráficos y Técnicas Interactivas, 16 hr(s)

Competencias: AG-C07,AG-C08

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: AG-C09

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: AG-C08,AG-C09

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: AG-C09

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: AG-C09

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI264 Aprendizaje Profundo (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI263 Introducción a Machine Learning (6to Sem-Pág 206)
  - CS261 Inteligencia Artificial (5to Sem-Pág 171)

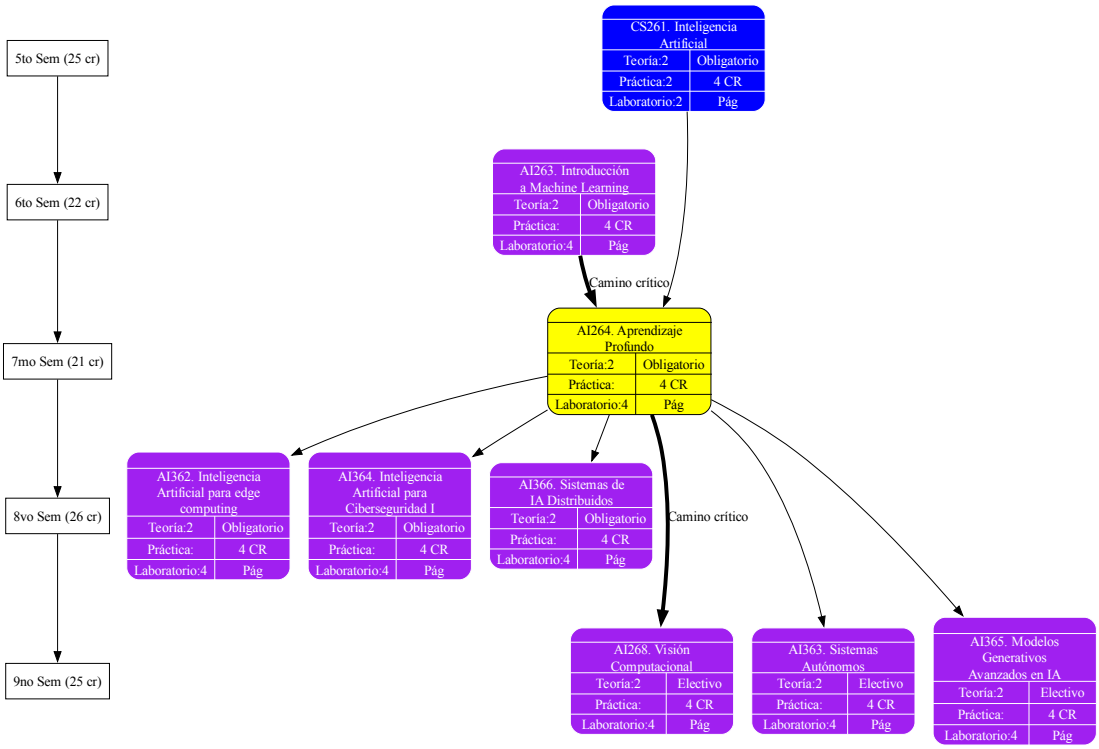


Figura 3.43: Mapa de Conexión. AI264 Aprendizaje Profundo

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Evaluar)

5. Contenido

5.1 Entrenamiento y Optimización de Redes Neuronales, 10 hr(s)

Competencias: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI265 Aprendizaje por Refuerzo (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI263 Introducción a Machine Learning (6to Sem-Pág 206)

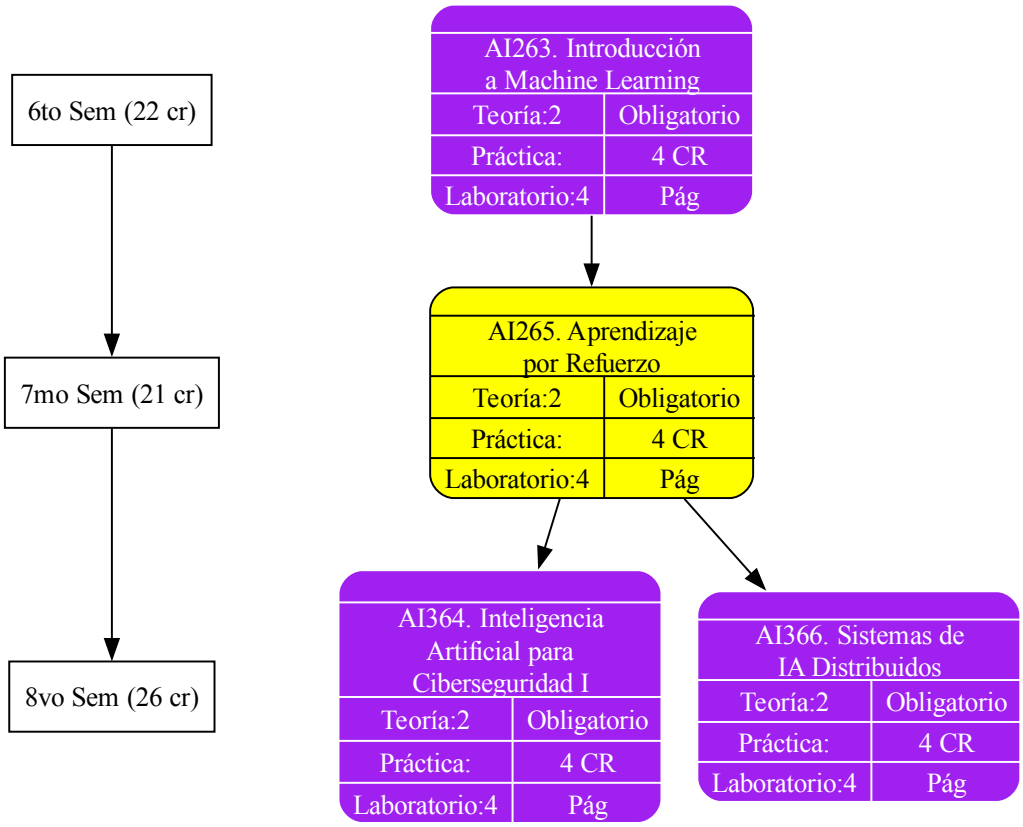


Figura 3.44: Mapa de Conexión. AI265 Aprendizaje por Refuerzo

2. Fundamentación

El aprendizaje por refuerzo (RL) es fundamental para sistemas autónomos. Este curso cubre algoritmos clave como Q-Learning, DQN y Policy Gradients, con aplicaciones en robótica y juegos.

3. Objetivos del curso.

- Diseñar agentes RL para entornos estocásticos.
- Implementar algoritmos RL usando frameworks modernos (OpenAI Gym, TensorFlow).
- Analizar problemas éticos en sistemas autónomos basados en RL.

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Fundamentos de RL, 18 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: I: Introducción al paradigma del Aprendizaje por Refuerzo: agente, entorno, acciones, estados, recompensas. **II:** Formalización de problemas secuenciales: Procesos de Decisión Markovianos (MDPs) y Procesos de Decisión Markovianos Parcialmente Observables (POMDPs). **III:** Concepto de horizonte y factor de descuento (gamma). **IV:** La Ecuación de Bellman: fundamento de optimalidad para funciones valor estado (V) y acción-estado (Q). **V:** Métodos de Programación Dinámica: Evaluación y Mejora Iterativa de Políticas. **VI:** Métodos de Monte Carlo: aprendizaje a partir de episodios completos. **VII:** Algoritmos de diferencia temporal (TD): TD(0), SARSA, y el balance entre bias y varianza. **VIII:** Q-Learning (off-policy): derivación, propiedades de convergencia y parámetros (learning rate, epsilon-greedy).

Objetivos de Aprendizaje I: Modelar problemas como MDPs [Evaluar] **II:** Implementar Q-Learning [Usar]

Bibliografía: [Sutton and Barto, 2018]

5.2 Métodos Avanzados de RL Clásico, 14 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: I: Limitaciones de los métodos tabulares y la maldición de la dimensionalidad. **II:** Aproximación de funciones valor: sustitución de tablas por modelos paramétricos (lineales, no lineales). **III:** Algoritmos de Gradiente Temporal: minimización del error de Bellman. **IV:** Eligibilidad Traces: unificación de Monte Carlo y TD (TD(δ)). **V:** Planificación y Aprendizaje Integrados: Modelos del entorno y algoritmos Dyna-Q. **VI:** Métodos basados en Políticas (Policy Search): introducción al gradiente de política. **VII:** Balance entre exploración y explotación: estrategias avanzadas (UCB, exploración basada en entropía).

Objetivos de Aprendizaje I: Modelar problemas como MDPs [Evaluar] **II:** Implementar Q-Learning [Usar]

Bibliografía: [Sutton and Barto, 2018]

5.3 RL Profundo (Deep RL), 20 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: I: Revolución del RL con Deep Learning: estabilización del entrenamiento. **II:** Deep Q-Networks (DQN): arquitectura, función de pérdida, y mecanismos de estabilización (Target Network, Experience Replay). **III:** Extensiones de DQN: Double DQN, Dueling DQN, Prioritized Experience Replay. **IV:** Política Estocástica en espacios continuos: introducción a los métodos Actor-Critic. **V:** Actor-Critic Methods: descomposición del gradiente, función ventaja (A2C). **VI:** Métodos de Actor-Critic Avanzados: A3C (Asynchronous Advantage Actor-Critic) y su arquitectura paralela. **VII:** Policy Gradient directos: REINFORCE con y sin baseline. **VIII:** Introducción a métodos de máxima entropía (Soft Actor-Critic - SAC) y aprendizaje offline.

Objetivos de Aprendizaje I: Entrenar agentes DQN en entornos complejos [Evaluar] **II:** Optimizar políticas con A3C [Usar]

Bibliografía: [Mnih et al., 2015, Silver et al., 2017]

5.4 Ética, Seguridad y Aplicaciones en RL, 12 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: I: Sesgos en el diseño de funciones de recompensa: especificación incorrecta, recompensas no alineadas y optimización de recompensas por hackeo (reward hacking). **II:** Desafíos de la generalización y robustez en agentes entrenados con RL. **III:** Seguridad en sistemas autónomos: evitación de fallos catastróficos, aprendizaje seguro (Safe RL) y alineación (AI Alignment). **IV:** Casos de estudio: análisis de fallos en simulaciones y robots reales. **V:** Ataques adversarios en RL: ataques a los sensores, a la función de política y a la función de recompensa del agente. **VI:** Implicaciones sociales: automatización, toma de decisiones y transparencia (agentes interpretables). **VII:** Marcos éticos para el desarrollo y despliegue de agentes autónomos.

Objetivos de Aprendizaje I: Identificar riesgos éticos en aplicaciones de RL [Familiarizarse] **II:** Proponer estrategias de mitigación [Usar]

Bibliografía: [Sutton and Barto, 2018]

6. Bibliografía

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

[Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550:354–359.

[Sutton and Barto, 2018] Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI266 IA para Ingenieria de Software (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI262 Procesamiento de Lenguaje Natural (6to Sem-Pág 203)
  - AI263 Introducción a Machine Learning (6to Sem-Pág 206)

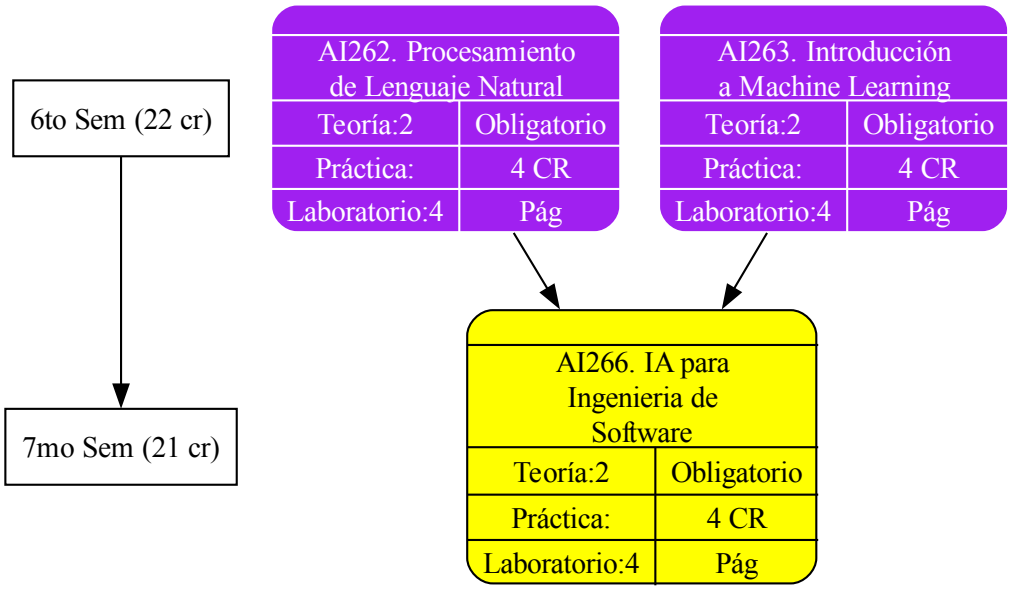


Figura 3.45: Mapa de Conexión. AI266 IA para Ingenieria de Software

2. Fundamentación

Este curso integra técnicas de IA en el ciclo de vida del desarrollo de software, cubriendo aplicaciones prácticas como generación de código, depuración automática y optimización de sistemas software mediante aprendizaje automático.

3. Objetivos del curso.

- Implementar pipelines de MLOps en entornos de desarrollo tradicionales
- Diseñar sistemas que integren modelos de IA en aplicaciones software
- Evaluar implicaciones éticas en la automatización de procesos de desarrollo

4. Competencias Generales (*Outcomes*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Usar)

5. Contenido

5.1 Fundamentos de IA para SE, 12 hr(s)

Competencias: AG-C09,AG-C12

Tópicos: I: Patrones de diseño para sistemas AI-SE **II:** Arquitecturas de sistemas híbridos (IA + software tradicional) **III:** Métricas de calidad para componentes inteligentes

Objetivos de Aprendizaje I: Diseñar arquitecturas software que incorporen componentes de IA [Evaluar] **II:** Evaluar trade-offs en sistemas híbridos [Usar]

Bibliografía: [Zhang and Chen, 2022]

5.2 Aplicaciones Avanzadas, 24 hr(s)

Competencias: AG-C09,AG-C12

Tópicos: I: Generación automática de código (Codex, CodeT5) **II:** Depuración inteligente **III:** Optimización automática de rendimiento

Objetivos de Aprendizaje I: Implementar herramientas de generación de código asistido por IA [Usar] **II:** Integrar sistemas de recomendación en IDEs [Evaluar]

Bibliografía: [Allamanis et al., 2018, Chen et al., 2021]

5.3 Ética en Automatización, 8 hr(s)

Competencias: AG-C09,AG-C12

Tópicos: I: Riesgos de la generación automática de código **II:** Sesgos en sistemas de recomendación para desarrolladores **III:** Responsabilidad legal en sistemas autónomos

Objetivos de Aprendizaje I: Identificar puntos críticos éticos en AI-SE [Familiarizarse] **II:** Proponer protocolos de mitigación [Usar]

Bibliografía: [Zhang and Chen, 2022]

6. Bibliografía

[Allamanis et al., 2018] Allamanis, M., Barr, E., Devanbu, P., and Sutton, C. (2018). A survey of machine learning for big code and naturalness. *ACM Computing Surveys*, 51(4).

[Chen et al., 2021] Chen, Y., Liu, X., and Shi, Z. (2021). Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *EMNLP*.

[Zhang and Chen, 2022] Zhang, M. and Chen, S. (2022). *AI Engineering: Applying AI in Software Development*. O'Reilly.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI400 Metodología de la Investigación (Obligatorio)

1. Información general

- **Semestre**: 7mo Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)
  - BEI201 Inglés III (3er Sem-Pág 122)

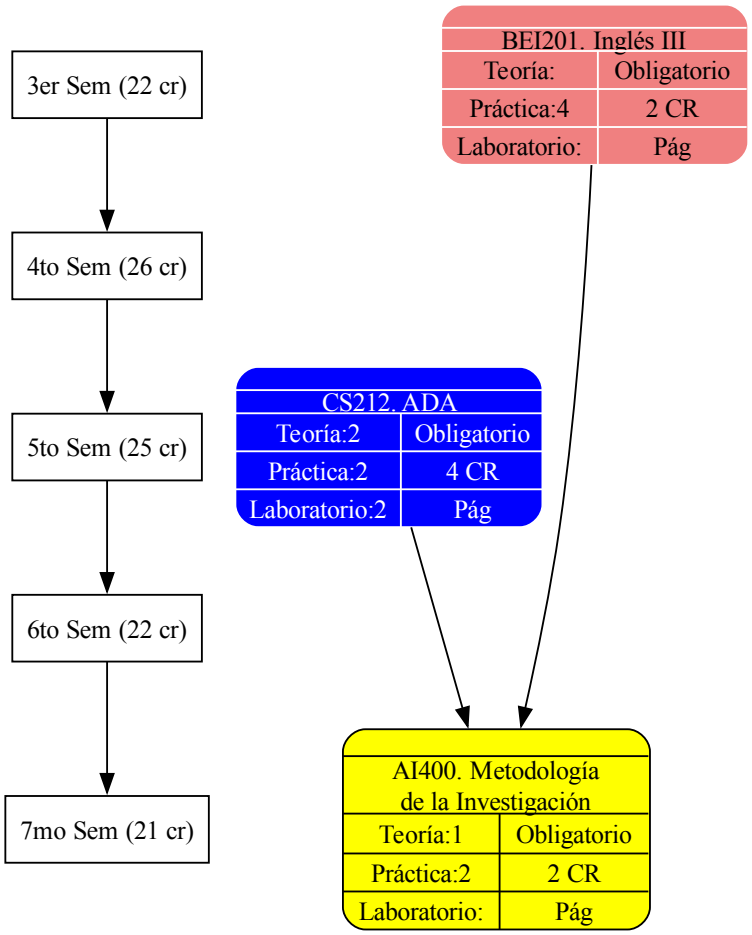


Figura 3.46: Mapa de Conexión. AI400 Metodología de la Investigación

2. Fundamentación

Este curso proporciona las herramientas metodológicas para diseñar, ejecutar y comunicar proyectos de investigación en computación, integrando aspectos éticos, técnicas de comunicación académica y trabajo colaborativo.

3. Objetivos del curso.

- Formular problemas de investigación en computación con rigor metodológico.

- Aplicar técnicas de búsqueda y análisis de literatura científica.
- Comunicar resultados de investigación de manera efectiva y ética.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Fundamentos de Investigación, 12 hr(s)

Competencias: AG-C02,AG-C04,AG-C05,AG-C08,AG-C09

Tópicos: I: El método científico en computación **II:** Tipos de estudios de investigación **III:** Formulación de preguntas de investigación

Objetivos de Aprendizaje I: Analizar problemas de investigación complejos [Usar] **II:** Diferenciar metodologías cuantitativas y cualitativas [Familiarizarse]

Bibliografía: [Booth et al., 2016, Creswell, 2014]

5.2 Ética en la Investigación, 12 hr(s)

Competencias: AG-C02,AG-C04,AG-C05,AG-C08,AG-C09

Tópicos: I: Responsabilidad profesional en investigación **II:** Plagio y propiedad intelectual **III:** Comités de ética en investigación

Objetivos de Aprendizaje I: Evaluar dilemas éticos en proyectos de investigación [Evaluar] **II:** Aplicar normas de citación y derechos de autor [Usar]

Bibliografía: [Resnik, 2015, IEEE, 2020]

5.3 Comunicación Científica, 24 hr(s)

Competencias: AG-C02,AG-C04,AG-C05,AG-C08,AG-C09

Tópicos: I: Redacción de artículos científicos **II:** Presentaciones académicas efectivas **III:** Revisión por pares

Objetivos de Aprendizaje I: Producir documentos técnicos siguiendo estándares [Usar] **II:** Defender resultados de investigación oralmente [Evaluar]

Bibliografía: [Sword, 2012, Zobel, 2014]

6. Bibliografía

[Booth et al., 2016] Booth, W. C., Colomb, G. G., and Williams, J. M. (2016). *The Craft of Research*. University of Chicago Press, 4th edition.

[Creswell, 2014] Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 4th edition.

[IEEE, 2020] IEEE (2020). Ieee code of ethics. Technical report, Institute of Electrical and Electronics Engineers.

[Resnik, 2015] Resnik, D. B. (2015). *The Ethics of Science: An Introduction*. Routledge.

[Sword, 2012] Sword, H. (2012). *Stylish Academic Writing*. Harvard University Press.

[Zobel, 2014] Zobel, J. (2014). *Writing for Computer Science*. Springer, 3rd edition.

Octavo Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS402 Proyecto de tesis 1 (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS401 Metodología de la Investigación (4to Sem-Pág 152)

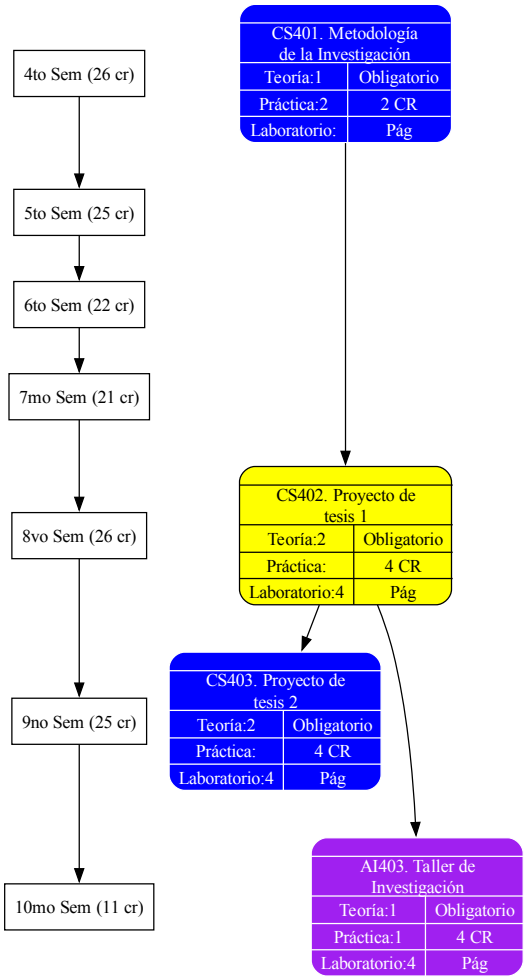


Figura 3.47: Mapa de Conexión. CS402 Proyecto de tesis 1

2. Fundamentación

Este curso tiene por objetivo que el alumno conduzca un estudio profundo del estado del arte de un tópico específico elegido para su proyecto de tesis. El alumno deberá producir un reporte técnico con experimentos preliminares comparativos y formular una propuesta de investigación, culminando junto a un asesor especializado en el área elegida.

3. Objetivos del curso.

- Realizar un estudio profundo del estado del arte en un tópico específico del área de computación.
- Demostrar dominio de la literatura relevante mediante análisis crítico y comparativo de enfoques existentes.
- Diseñar y ejecutar experimentos preliminares que reproduzcan y comparen técnicas del estado del arte.
- Redactar un reporte técnico de calidad con marco teórico, metodología y resultados experimentales.
- Formular una propuesta de investigación con pregunta, objetivos y cronograma.
- Seleccionar un asesor de tesis que domine el área de investigación elegida.
- Los entregables de este curso son:

  Avance parcial: Bibliografía sólida y borrador del marco teórico en formato LaTeX.

  Final: Reporte Técnico con experimentos preliminares comparativos y propuesta de investigación aprobada por el asesor.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Profundización en el Estado del Arte Específico, 10 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Selección y refinamiento del tópico de investigación en coordinación con el asesor. **II:** Búsqueda exhaustiva multi-fuente: IEEE Xplore, ACM Digital Library, Google Scholar, arXiv y bases de datos especializadas. **III:** Identificación de trabajos seminales y artículos más citados del área. **IV:** Construcción del grafo de citaciones y análisis de la evolución temporal del tópico. **V:** Identificación de subáreas, variantes y enfoques principales del campo.

Objetivos de Aprendizaje **I:** Construir un mapa completo de la literatura de un tópico específico identificando los trabajos fundamentales. [Evaluar] **II:** Analizar la evolución temporal de un área de investigación identificando tendencias y brechas. [Evaluar] **III:** Organizar la literatura en subcategorías coherentes que reflejen la estructura del campo. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.2 Análisis Crítico y Comparativo de Enfoques, 10 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Análisis crítico de propuestas: metodología, datasets, métricas y limitaciones de cada enfoque. **II:** Construcción de tablas comparativas de propuestas existentes con criterios objetivos.

III: Identificación de brechas de investigación (research gaps) no resueltas en el área. **IV:** Formulación de la pregunta de investigación propia a partir del análisis del estado del arte. **V:** Marco teórico: síntesis estructurada y coherente del conocimiento relevante del área.

Objetivos de Aprendizaje I: Comparar sistemáticamente múltiples enfoques usando criterios objetivos y cuantificables. [Evaluar] **II:** Identificar brechas de investigación no resueltas en un área específica de computación. [Evaluar] **III:** Formular una pregunta de investigación propia sustentada en el análisis del estado del arte. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Diseño y Ejecución de Experimentos Preliminares, 12 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Diseño experimental: variables, métricas y posibles benchmarks del área. **II:** Reproducción de resultados de trabajos relevantes del estado del arte. **III:** Implementación de baselines para comparación objetiva de enfoques. **IV:** Análisis estadístico básico: pruebas de significancia e intervalos de confianza. **V:** Gestión de datos experimentales: datasets, configuraciones, versiones y registro de resultados.

Objetivos de Aprendizaje I: Reproducir los resultados de al menos dos trabajos relevantes del estado del arte del área. [Usar] **II:** Diseñar y ejecutar un experimento comparativo preliminar con baselines definidos. [Usar] **III:** Interpretar y reportar resultados experimentales con rigor estadístico apropiado. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Redacción del Reporte Técnico, 12 hr(s)

Competencias: AG-C04,AG-C09

Tópicos: I: Estructura del reporte técnico: introducción, marco teórico, metodología, experimentos, resultados y conclusiones. **II:** Redacción del marco teórico: organización temática y síntesis crítica de la literatura. **III:** Documentación de la metodología y configuración experimental en LaTeX. **IV:** Presentación de resultados: tablas, gráficas y análisis comparativos de alta calidad. **V:** Ciclos de revisión: escritura, revisión con asesor e incorporación de retroalimentación.

Objetivos de Aprendizaje I: Redactar un reporte técnico completo con marco teórico y experimentos preliminares. [Usar] **II:** Estructurar y presentar resultados experimentales con visualizaciones claras y precisas. [Usar] **III:** Aplicar ciclos de revisión para mejorar la calidad, coherencia y rigor del documento. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Propuesta de Investigación y Selección de Asesor, 8 hr(s)

Competencias: AG-C02,AG-C05,AG-C09

Tópicos: I: Estructura de una propuesta de investigación: problema, motivación y objetivos. **II:** Definición del alcance del proyecto de tesis: metodología, contribución esperada y cronograma. **III:** Análisis de viabilidad: recursos, tiempo, riesgos y contingencias del proyecto. **IV:** Criterios para la selección del asesor de tesis: experiencia, disponibilidad y alineación temática. **V:** Proceso de formalización del vínculo investigador-asesor en el programa académico.

Objetivos de Aprendizaje I: Elaborar una propuesta de investigación que defina el problema, objetivos y posible solución. [Evaluar] **II:** Seleccionar un asesor con experiencia demostrada en el área de investigación elegida. [Usar] **III:** Definir un plan de tesis viable con objetivos, alcance, cronograma y análisis de riesgos. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.6 Presentación y Defensa del Estado del Arte, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: **I:** Preparación de la presentación oral del reporte técnico ante el comité evaluador. **II:** Técnicas de defensa ante preguntas y cuestionamientos de un comité académico. **III:** Elaboración de póster técnico del estado del arte para presentación en simposio. **IV:** Incorporación de retroalimentación del comité evaluador en la revisión final del reporte. **V:** Planificación de la transición a CS403: objetivos, contribución esperada y siguientes pasos.

Objetivos de Aprendizaje **I:** Presentar y defender oralmente el reporte técnico ante un comité académico con rigor. [Evaluar] **II:** Incorporar constructivamente la retroalimentación del comité en una revisión del reporte. [Usar] **III:** Planificar la transición hacia el proyecto de tesis (CS403) con objetivos y pasos claros. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library.* Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library.* College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library.* IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI362 Inteligencia Artificial para edge computing (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)

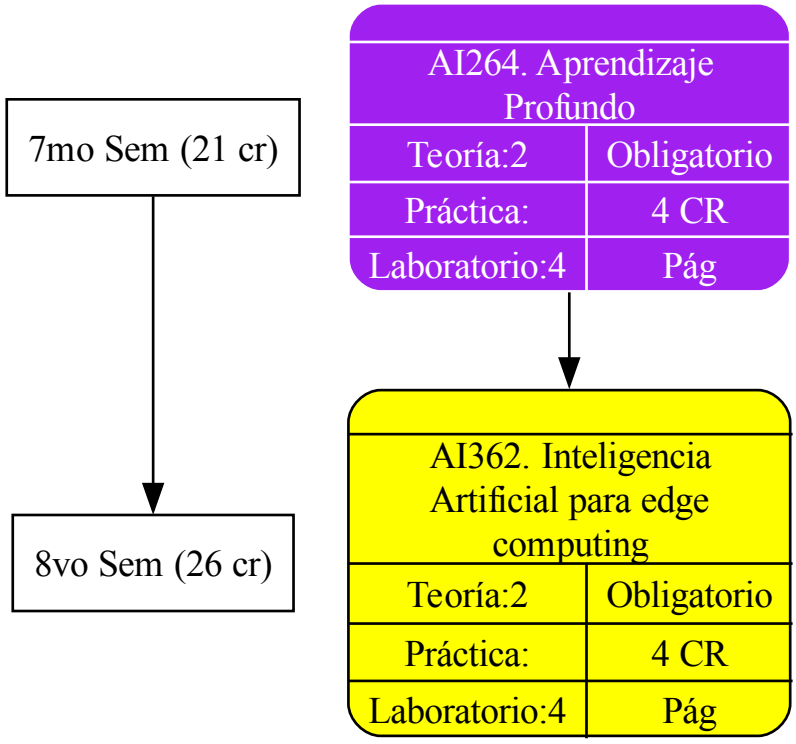


Figura 3.48: Mapa de Conexión. AI362 Inteligencia Artificial para edge computing

2. Fundamentación

Este curso aborda el despliegue de modelos de IA en dispositivos edge (IoT, móviles, embedded), cubriendo técnicas de optimización, frameworks especializados y consideraciones de privacidad/seguridad en sistemas distribuidos.

3. Objetivos del curso.

- Implementar modelos de IA optimizados para hardware edge
- Diseñar pipelines de inferencia distribuida
- Evaluar compensaciones entre precisión, latencia y consumo energético

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Evaluar)

5. Contenido

5.1 Fundamentos de Edge AI, 12 hr(s)

Competencias: AG-C09

Tópicos: I: Arquitecturas hardware para edge computing **II:** Taxonomía de modelos edge-friendly (MobileNets, TinyML) **III:** Frameworks de despliegue (TensorFlow Lite, ONNX Runtime)

Objetivos de Aprendizaje I: Seleccionar modelos adecuados para restricciones hardware [Evaluar] **II:** Perfilar consumo energético en dispositivos reales [Usar]

Bibliografía: [Han and Lin, 2022]

5.2 Técnicas de Optimización, 24 hr(s)

Competencias: AG-C09

Tópicos: I: Cuantización y pruning **II:** Knowledge distillation **III:** Compilación especializada (TVM, Apache MXNet)

Objetivos de Aprendizaje I: Reducir modelos para microcontroladores [Evaluar] **II:** Implementar pipelines híbridos (edge+cloud) [Usar]

Bibliografía: [Howard et al., 2017, Saraph and Warden, 2021]

5.3 Seguridad y Privacidad, 8 hr(s)

Competencias: AG-C09

Tópicos: I: Federated learning en edge **II:** Homomorphic encryption **III:** Ataques a modelos edge (adversarial examples)

Objetivos de Aprendizaje I: Analizar vulnerabilidades en sistemas edge [Familiarizarse] **II:** Implementar defensas básicas [Usar]

Bibliografía: [Han and Lin, 2022]

6. Bibliografía

[Han and Lin, 2022] Han, S. and Lin, Y. (2022). *Edge AI: Systems Design for Embedded Machine Learning*. Springer.

[Howard et al., 2017] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*.

[Saraph and Warden, 2021] Saraph, V. and Warden, P. (2021). Tinyml: Enabling always-on ml on the edge. In *ACM HotNets*.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI364 Inteligencia Artificial para Ciberseguridad I (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)
  - AI265 Aprendizaje por Refuerzo (7mo Sem-Pág 231)

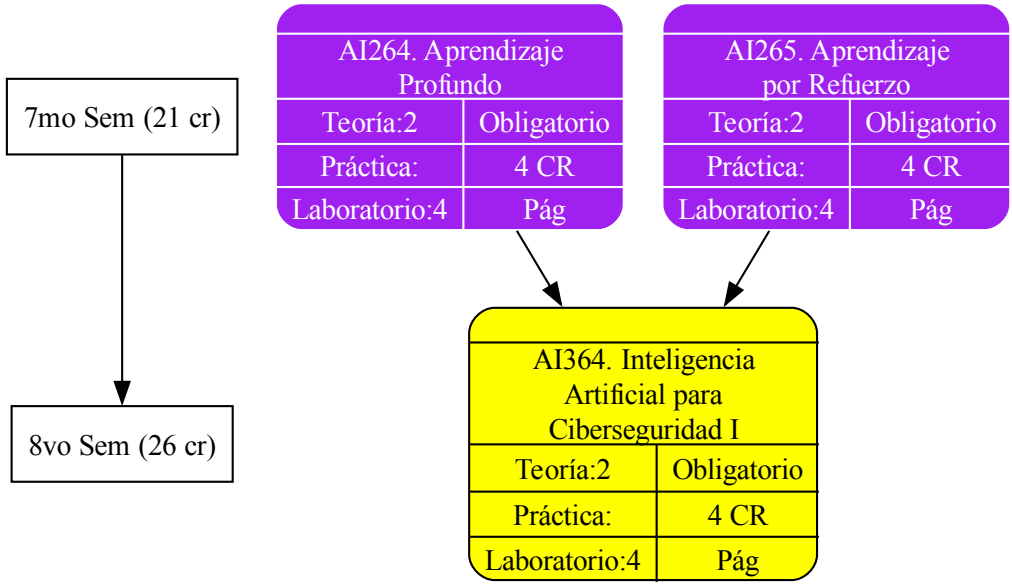


Figura 3.49: Mapa de Conexión. AI364 Inteligencia Artificial para Ciberseguridad I

2. Fundamentación

Este curso cubre técnicas avanzadas de IA aplicadas a ciberseguridad, incluyendo detección de amenazas mediante aprendizaje automático, generación de adversarios y defensas autónomas contra ataques modernos.

3. Objetivos del curso.

- Implementar sistemas de detección de intrusos basados en IA
- Diseñar pruebas de estrés para modelos de seguridad
- Analizar implicaciones éticas del uso ofensivo de IA

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Evaluar)

5. Contenido

5.1 Fundamentos de IA en Cybersecurity, 12 hr(s)

Competencias: AG-C02

Tópicos: I: Taxonomía de amenazas asistidas por IA **II:** Arquitecturas de modelos para análisis de malware **III:** Datasets de seguridad (CIC-IDS, NSL-KDD)

Objetivos de Aprendizaje I: Clasificar amenazas usando redes neuronales [Evaluar] **II:** Preprocesar logs de seguridad para modelos ML [Usar]

Bibliografía: [Apruzzese and Laskov, 2023]

5.2 Ataques y Defensas, 24 hr(s)

Competencias: AG-C02

Tópicos: I: Adversarial examples en modelos de seguridad **II:** Generación automática de exploits (Fuzzing con RL) **III:** Sistemas de detección autoadaptativos

Objetivos de Aprendizaje I: Implementar ataques adversarios a modelos ML [Usar] **II:** Diseñar defensas metamórficas [Evaluar]

Bibliografía: [Goodfellow et al., 2014, Anderson and Roth, 2020]

5.3 Ética en Cybersecurity, 8 hr(s)

Competencias: AG-C02

Tópicos: I: Armas cibernéticas autónomas **II:** Responsabilidad en vulnerabilidades IA-generadas **III:** Regulaciones internacionales (AI Cyber Initiative)

Objetivos de Aprendizaje I: Evaluar riesgos de sistemas autónomos [Familiarizarse] **II:** Diseñar protocolos de rendición de cuentas [Usar]

Bibliografía: [Apruzzese and Laskov, 2023]

6. Bibliografía

[Anderson and Roth, 2020] Anderson, H. and Roth, P. (2020). Metamorphic testing for adversarial defenses. *ACM CCS*.

[Apruzzese and Laskov, 2023] Apruzzese, G. and Laskov, P. (2023). *AI for Cybersecurity*. Springer.

[Goodfellow et al., 2014] Goodfellow, I., Shlens, J., and Szegedy, C. (2014). *Explaining and Harnessing Adversarial Examples*. ICLR.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI366 Sistemas de IA Distribuidos (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)
  - AI265 Aprendizaje por Refuerzo (7mo Sem-Pág 231)

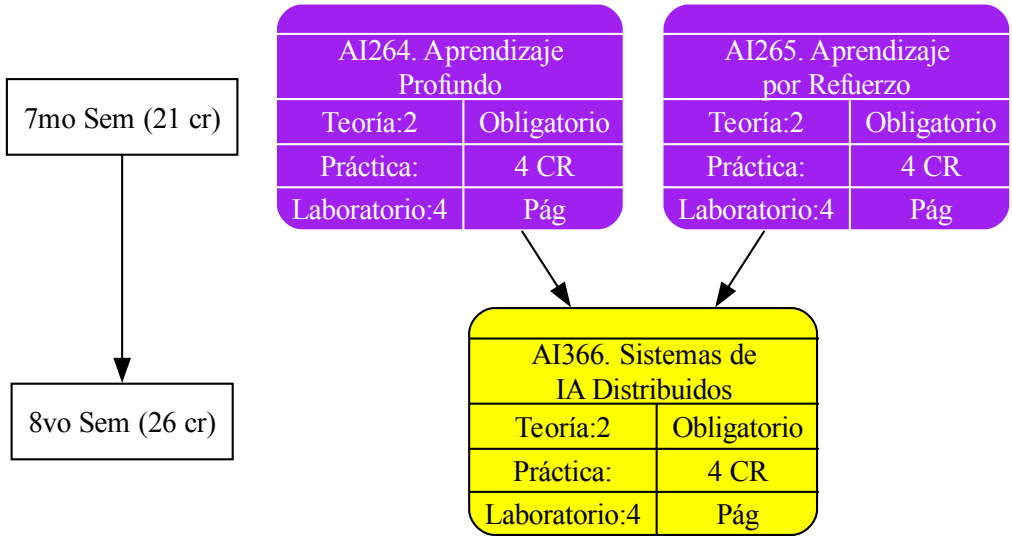


Figura 3.50: Mapa de Conexión. AI366 Sistemas de IA Distribuidos

2. Fundamentación

Este curso cubre el diseño e implementación de sistemas de IA distribuidos, incluyendo aprendizaje federado, inteligencia de enjambre y arquitecturas descentralizadas para entrenamiento colaborativo de modelos.

3. Objetivos del curso.

- Implementar algoritmos de aprendizaje federado
- Diseñar arquitecturas descentralizadas para entrenamiento distribuido
- Evaluar compensaciones en privacidad-comunicación-eficiencia

4. Competencias Generales (*Outcomes*)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Fundamentos de IA Distribuida, 12 hr(s)

Competencias: AG-C05,AG-C09

Tópicos: I: Arquitecturas centralizadas vs descentralizadas **II:** Protocolos de comunicación (gRPC, MPI) **III:** Sincronización en sistemas heterogéneos

Objetivos de Aprendizaje I: Comparar enfoques de distribución [Evaluar] **II:** Configurar entornos de prueba distribuidos [Usar]

Bibliografía: [Bonawitz et al., 2019]

5.2 Técnicas Avanzadas, 24 hr(s)

Competencias: AG-C05,AG-C09

Tópicos: I: Aprendizaje federado (horizontal/vertical) **II:** Algoritmos de enjambre para optimización **III:** Blockchain para IA descentralizada

Objetivos de Aprendizaje I: Implementar FL con PySyft [Usar] **II:** Diseñar sistemas tolerantes a fallos [Evaluar]

Bibliografía: [Nguyen et al., 2022, Li et al., 2021]

5.3 Seguridad y Gobernanza, 8 hr(s)

Competencias: AG-C05,AG-C09

Tópicos: I: Ataques a sistemas distribuidos (poisoning, sybil) **II:** Mecanismos de consenso para modelos **III:** Gobernanza en sistemas autónomos colectivos

Objetivos de Aprendizaje I: Analizar vectores de ataque [Familiarizarse] **II:** Implementar defensas básicas [Usar]

Bibliografía: [Bonawitz et al., 2019]

6. Bibliografía

[Bonawitz et al., 2019] Bonawitz, K., Kairouz, P., and McMahan, B. (2019). *Federated Learning: Fundamentals and Advances*. MIT Press.

[Li et al., 2021] Li, Q., Lin, Y., and Zhou, Z. (2021). Decentralized federated learning: A survey and perspective. In *NeurIPS*.

[Nguyen et al., 2022] Nguyen, T., Nguyen, D., and Khoa, N. L. D. (2022). Swarm intelligence for distributed machine learning. *IEEE Transactions on Neural Networks*, 33(7).

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI401 Proyecto de tesis 1 (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS400 Prácticas Pre-profesionales (9no Sem-Pág 274)
  - FG211 Ética Profesional (6to Sem-Pág 217)

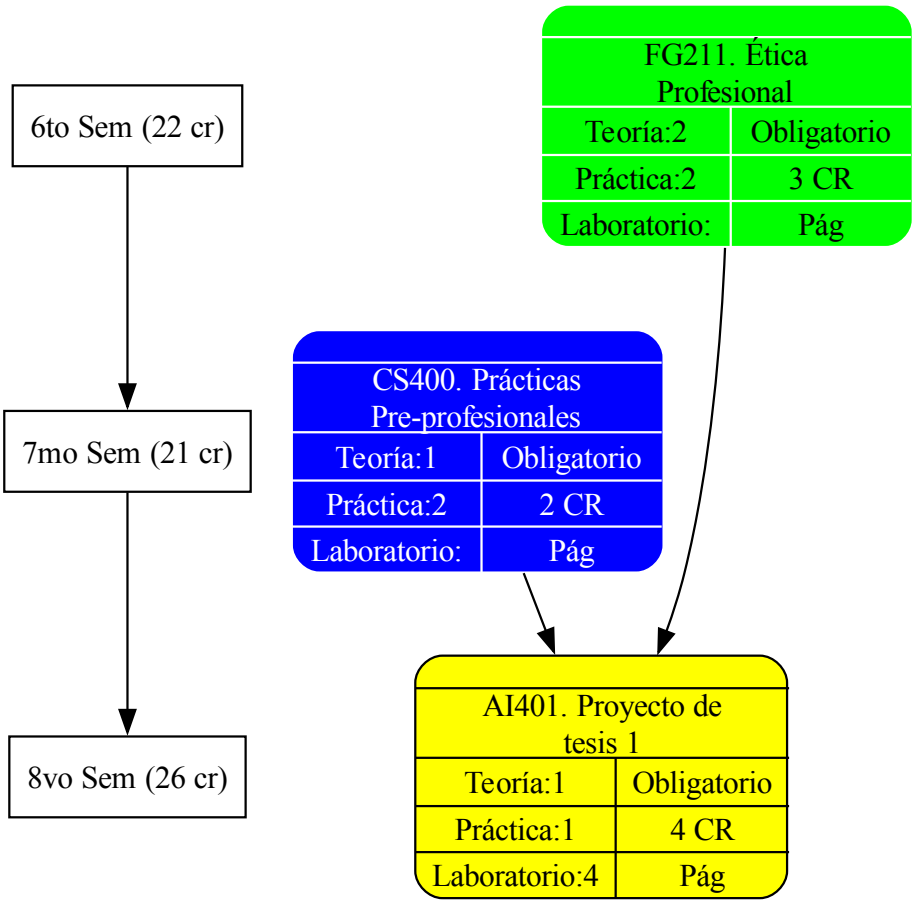


Figura 3.51: Mapa de Conexión. AI401 Proyecto de tesis 1

2. Fundamentación

Este curso tiene por objetivo que el alumno aprenda a realizar una investigación de carácter científico en el área de computación. Los docentes del curso determinarán un área de estudio para cada alumno y se les hará entrega de bibliografía para analizar. A partir de la misma y de fuentes adicionales investigadas por el alumno, el alumno deberá construir un artículo del tipo survey sobre el tema asignado.

3. Objetivos del curso.

- Aprender a iniciar una investigación científica en el área de computación con rigor metodológico.
- Conocer las principales fuentes bibliográficas académicas: IEEE Xplore, ACM Digital Library, Google Scholar y arXiv.
- Analizar críticamente propuestas existentes sobre un tópico y relacionarlas en una revisión bibliográfica.
- Redactar documentos técnicos en computación usando LaTeXcon plantillas de conferencias IEEE y ACM.
- Reproducir resultados existentes en un tópico mediante implementación y experimentación.
- Construir un artículo de tipo survey con taxonomía, síntesis y comparación de enfoques existentes.
- Los entregables de este curso son:

  Avance parcial: Dominio del tema del artículo y bibliografía preliminar en formato de artículo LaTeX.

  Final: Artículo del tipo survey concluido, conteniendo opcionalmente los resultados experimentales de la(s) técnica(s) estudiada(s).

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Fundamentos de Investigación Científica en Computación, 8 hr(s)

Competencias: AG-C05,AG-C08

Tópicos: I: Tipos de investigación en computación: empírica, teórica, constructiva y evaluativa. **II:** El método científico aplicado a la investigación en computación. **III:** Principales bases de datos académicas: IEEE Xplore, ACM Digital Library, Google Scholar, Semantic Scholar y arXiv. **IV:** Estrategias de búsqueda bibliográfica: operadores booleanos, seguimiento de citas y búsqueda sistemática. **V:** Gestores de referencias: Zotero, Mendeley e integración con BibTeX.

Objetivos de Aprendizaje I: Identificar los tipos de investigación en computación y sus diferencias metodológicas. [Familiarizarse] **II:** Utilizar las principales bases de datos académicas para encontrar literatura relevante en un tópico dado. [Usar] **III:** Organizar referencias bibliográficas usando gestores de referencias y formato BibTeX. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.2 Lectura Crítica y Análisis de Literatura Científica, 8 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Estructura de un artículo científico: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. **II:** Técnicas para la lectura eficiente de artículos científicos (lectura en tres pasadas). **III:** Evaluación crítica: identificación de contribuciones, limitaciones y problemas abiertos. **IV:** Construcción de una matriz de lectura y tablas comparativas de trabajos relacionados. **V:** Clasificación y diseño de taxonomías de enfoques existentes sobre un tópico.

Objetivos de Aprendizaje **I:** Leer y evaluar críticamente artículos científicos identificando sus contribuciones y limitaciones. [Usar] **II:** Construir una matriz de lectura organizada que sintetice múltiples trabajos sobre un tópico dado. [Usar] **III:** Diseñar una taxonomía preliminar de enfoques para un área de investigación asignada. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Ética en Investigación e Integridad Académica, 6 hr(s)

Competencias: AG-C02,AG-C04

Tópicos: **I:** Integridad académica: plagio, autoplagio, fabricación y falsificación de datos. **II:** Ética en investigación en computación: privacidad, sesgo algorítmico y estudios con sujetos humanos. **III:** Acceso abierto, licencias (Creative Commons, GPL) y propiedad intelectual en software. **IV:** Buenas prácticas de citación y atribución de autoría en trabajos colaborativos. **V:** El proceso de revisión por pares: rol del revisor y responsabilidades del autor.

Objetivos de Aprendizaje **I:** Distinguir las formas de mala conducta académica y aplicar prácticas de citación correctas. [Usar] **II:** Analizar dilemas éticos en investigación computacional aplicando principios de integridad. [Evaluar] **III:** Explicar los derechos de propiedad intelectual y las licencias relevantes en computación. [Familiarizarse]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Redacción Técnica y Herramientas de Escritura Científica, 10 hr(s)

Competencias: AG-C04,AG-C09

Tópicos: **I:** Estructura de LaTeX: preámbulo, secciones, ecuaciones, figuras, tablas y referencias cruzadas. **II:** Plantillas de conferencias y revistas: IEEE y ACM. **III:** Estilo de escritura académica en computación: claridad, precisión y voz activa. **IV:** Redacción de cada sección del artículo: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. **V:** Integración Continua de documentos: control de versiones de documentos LaTeXcon Git. **VI:** Gestión de bibliografía con BibTeX y JabRef.

Objetivos de Aprendizaje **I:** Redactar un artículo técnico completo en LaTeXusando plantillas IEEE o ACM. [Usar] **II:** Aplicar las convenciones de escritura académica propias de la comunidad de computación. [Usar] **III:** Evaluar la calidad de redacción de un artículo propio o ajeno según criterios de claridad y estructura. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Revisión Sistemática de Literatura y Artículos de Survey, 10 hr(s)

Competencias: AG-C04,AG-C08,AG-C09

Tópicos: **I:** Tipos de revisiones: narrativa, sistemática, estudio de mapeo y meta-análisis. **II:** Protocolo de revisión sistemática de literatura (SLR): preguntas de investigación, estrategia de búsqueda y criterios de selección. **III:** Pautas PRISMA: diagrama de flujo de selección de estudios primarios. **IV:** Técnicas de síntesis: narrativa, votación y análisis temático. **V:** Diseño de taxonomías y esquemas de clasificación para un survey. **VI:** Estructura y redacción de un artículo del tipo survey.

Objetivos de Aprendizaje I: Diseñar y ejecutar un protocolo de revisión sistemática de literatura para un tópico de computación. [Usar] **II:** Aplicar las pautas PRISMA para documentar y reportar el proceso de selección de estudios. [Usar] **III:** Redactar un artículo de tipo survey con taxonomía, síntesis crítica y comparación de enfoques. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.6 Presentación y Difusión de Resultados de Investigación, 6 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: I: Técnicas de presentación oral para audiencias científicas: estructura, manejo del tiempo y sesión de preguntas. **II:** Diseño de pósters para conferencias y simposios de investigación. **III:** Respuesta a revisores: cómo abordar comentarios de forma constructiva y argumentada. **IV:** Planificación de un proyecto de investigación: cronograma, hitos y entregables. **V:** Prácticas de colaboración y coautoría en investigación científica.

Objetivos de Aprendizaje I: Preparar y exponer una presentación oral de resultados de investigación ante una audiencia académica. [Usar] **II:** Elaborar un plan de trabajo para un proyecto de investigación con cronograma e hitos. [Usar] **III:** Responder constructivamente a comentarios de revisores en el proceso de publicación. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library.* Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library.* College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library.* IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

BI101 Biología (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**: Ninguno

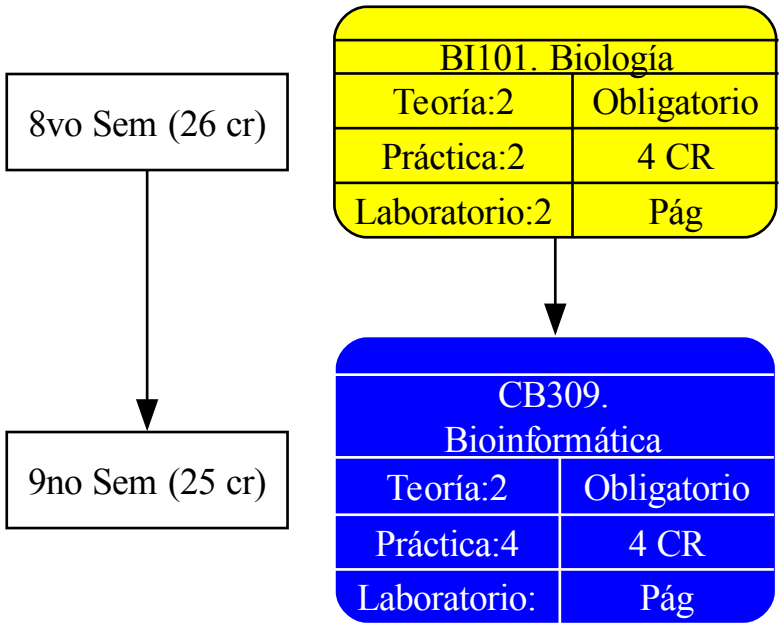


Figura 3.52: Mapa de Conexión. BI101 Biología

2. Fundamentación

La biología es la ciencia de la vida y proporciona una base fundamental para campos interdisciplinarios como la bioinformática, la biología computacional y la inteligencia artificial inspirada en la biología. Este curso introduce los conceptos fundamentales de la biología, desde la célula hasta la evolución, con un enfoque especial en aplicaciones computacionales.

3. Objetivos del curso.

- Comprender los principios básicos de la biología celular y molecular.
- Conocer los procesos fundamentales de la vida, como la replicación del ADN, la transcripción y la traducción.
- Apreciar la interrelación entre la biología y la computación en áreas como la bioinformática.
- Desarrollar habilidades para analizar datos biológicos de manera cualitativa y cuantitativa.
- Fomentar el pensamiento interdisciplinario entre biología y ciencias computacionales.

4. Competencias Generales (*Outcomes*)

AG-C07) Conocimientos de Computación: Aplica conocimientos de matemáticas, ciencias y computación. (Familiarizarse)

AG-C10) Indagación: Estudia problemas complejos de computación usando métodos de ciencias de la información. (Familiarizarse)

5. Contenido

5.1 Introducción a la Biología y la Vida, 6 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Características de los seres vivos y definición de vida. **II:** Niveles de organización biológica (de moléculas a ecosistemas). **III:** El método científico en biología. **IV:** Modelos computacionales de sistemas vivos.

Objetivos de Aprendizaje **I:** Describir las características de los seres vivos. [Familiarizarse] **II:** Identificar los diferentes niveles de organización biológica. [Usar] **III:** Aplicar el método científico en el contexto de la biología. [Evaluar] **IV:** Explicar modelos computacionales de ecosistemas. [Usar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.2 Química de la Vida, 6 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Átomos, moléculas y enlaces químicos. **II:** Biomoléculas: proteínas, carbohidratos, lípidos, ácidos nucleicos. **III:** Relación con estructuras de datos en computación. **IV:** Cadenas de ADN como secuencias computacionales.

Objetivos de Aprendizaje **I:** Identificar las principales biomoléculas y sus funciones. [Familiarizarse] **II:** Explicar la relación entre estructuras biológicas y de datos. [Usar] **III:** Analizar secuencias biológicas desde una perspectiva computacional. [Evaluar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.3 Estructura y Función Celular, 8 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Tipos de células: procariotas y eucariotas. **II:** Orgánulos celulares y sus funciones. **III:** Membranas celulares y transporte. **IV:** Redes biológicas y modelado computacional.

Objetivos de Aprendizaje **I:** Comparar y contrastar células procariotas y eucariotas. [Familiarizarse] **II:** Describir la función de los principales orgánulos celulares. [Usar] **III:** Explicar los mecanismos de transporte a través de membranas. [Evaluar] **IV:** Introducir conceptos de redes biológicas y su modelado. [Usar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.4 Metabolismo y Energía, 6 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Fotosíntesis y respiración celular. **II:** Enzimas y catálisis biológica. **III:** Analogías con algoritmos de optimización. **IV:** Eficiencia energética en sistemas biológicos y computacionales.

Objetivos de Aprendizaje **I:** Explicar los procesos de fotosíntesis y respiración celular. [Familiarizarse] **II:** Describir el papel de las enzimas en el metabolismo. [Usar] **III:** Establecer analogías entre eficiencia energética biológica y computacional. [Evaluar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.5 Genética Básica, 8 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** ADN: estructura y replicación. **II:** ARN: transcripción y traducción. **III:** Conceptos de herencia mendeliana. **IV:** Aplicación en algoritmos genéticos.

Objetivos de Aprendizaje **I:** Describir la estructura y replicación del ADN. [Familiarizarse] **II:** Explicar los procesos de transcripción y traducción. [Usar] **III:** Comprender los principios de la herencia mendeliana. [Evaluar] **IV:** Aplicar conceptos genéticos en algoritmos computacionales. [Usar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.6 Evolución y Diversidad, 6 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Teoría de la evolución por selección natural. **II:** Evidencias de la evolución. **III:** Biodiversidad y clasificación biológica. **IV:** Simulación evolutiva en computación.

Objetivos de Aprendizaje **I:** Explicar la teoría de la evolución por selección natural. [Familiarizarse] **II:** Describir las evidencias de la evolución. [Usar] **III:** Analizar los mecanismos de la evolución. [Evaluar] **IV:** Aplicar conceptos evolutivos en simulaciones computacionales. [Usar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

5.7 Ecología Básica y Aplicaciones, 8 hr(s)

Competencias: AG-C07,AG-C10

Tópicos: **I:** Interacciones entre organismos. **II:** Ciclos biogeoquímicos. **III:** Dinámicas poblacionales. **IV:** Modelado computacional de sistemas ecológicos.

Objetivos de Aprendizaje **I:** Describir las interacciones ecológicas básicas. [Familiarizarse] **II:** Explicar los ciclos biogeoquímicos. [Usar] **III:** Analizar dinámicas poblacionales. [Evaluar] **IV:** Aplicar modelado computacional a problemas ecológicos. [Usar]

Bibliografía: [Urry et al., 2017, Raven et al., 2017]

6. Bibliografía

[Raven et al., 2017] Raven, P. H., Johnson, G. B., Mason, K. A., Losos, J. B., and Singer, S. R. (2017). *Biology*. McGraw-Hill Education.

[Urry et al., 2017] Urry, L. A., Cain, M. L., Wasserman, S. A., Minorsky, P. V., and Reece, J. B. (2017). *Campbell Biology*. Pearson.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

EX301 Actividades Extracurriculares (Obligatorio)

1. Información general

- **Semestre**: 8vo Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**: Ninguno

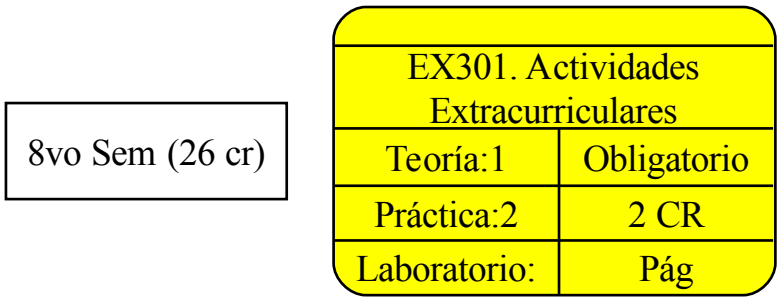


Figura 3.53: Mapa de Conexión. EX301 Actividades Extracurriculares

2. Fundamentación

Este curso registra la participación del estudiante en actividades complementarias a la formación académica, como talleres, congresos, proyección social, voluntariados o competencias técnicas. Su objetivo es fomentar el desarrollo integral, habilidades blandas y compromiso social.

3. Objetivos del curso.

- Participar en actividades que complementen su formación profesional.
- Desarrollar habilidades de comunicación y trabajo en equipo.
- Contribuir a iniciativas de impacto social o técnico.

4. Competencias Generales (*Outcomes*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Familiarizarse)

AG-C06) Aprendizaje a lo largo de la vida: Reconoce la importancia del aprendizaje continuo y la adaptación a nuevas tecnologías. (Familiarizarse)

5. Contenido

5.1 Actividades Extracurriculares, 48 hr(s)

Competencias: AG-C01,AG-C06

Tópicos: **I:** Talleres técnicos o interdisciplinarios. **II:** Asistencia a congresos, seminarios o charlas. **III:** Proyección social (voluntariados, mentorías). **IV:** Competencias académicas (hackathons, ferias científicas).

Objetivos de Aprendizaje I: Demostrar participación en al menos 3 actividades [Familiarizarse]. **II:** Elaborar informes reflexivos sobre las experiencias [Usar]. **III:** Colaborar en equipos para proyectos sociales o técnicos [Evaluar].

Bibliografía: [ACM, 2019, ABET Computing Accreditation Commission, 2022]

6. Bibliografía

[ABET Computing Accreditation Commission, 2022] ABET Computing Accreditation Commission (2022). Abet criteria for student professional development. Technical report, ABET.

[ACM, 2019] ACM (2019). Guidelines for extracurricular activities in computing. Technical report, ACM.

Noveno Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS3I1 Seguridad en Computación (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS231 Redes y Comunicación (7mo Sem-Pág 220)

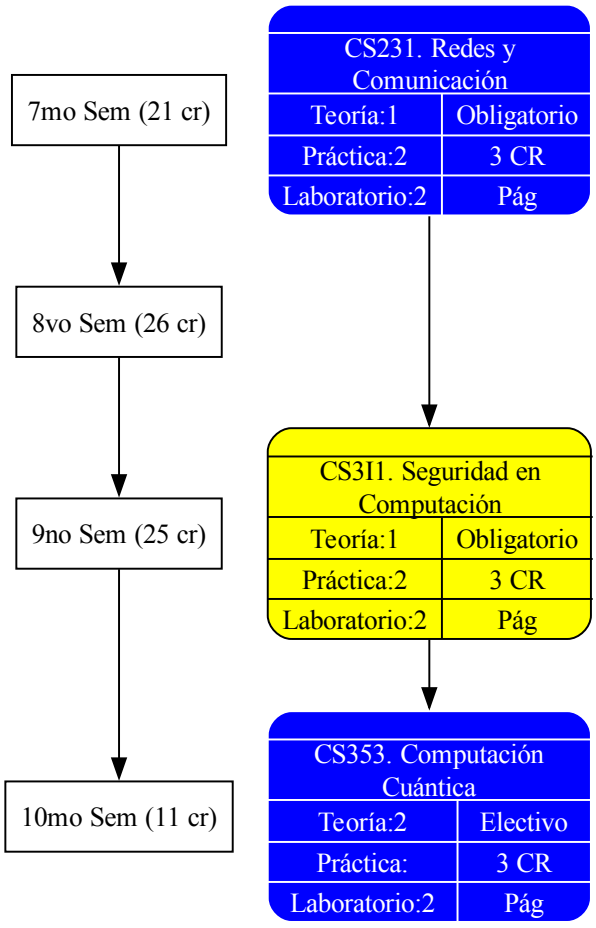


Figura 3.54: Mapa de Conexión. CS3I1 Seguridad en Computación

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Principios y Mentalidad de Seguridad, 6 hr(s)

Competencias: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS3P1 Computación Paralela y Distribuída (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS212 Análisis y Diseño de Algoritmos (5to Sem-Pág 166)
  - CS231 Redes y Comunicación (7mo Sem-Pág 220)

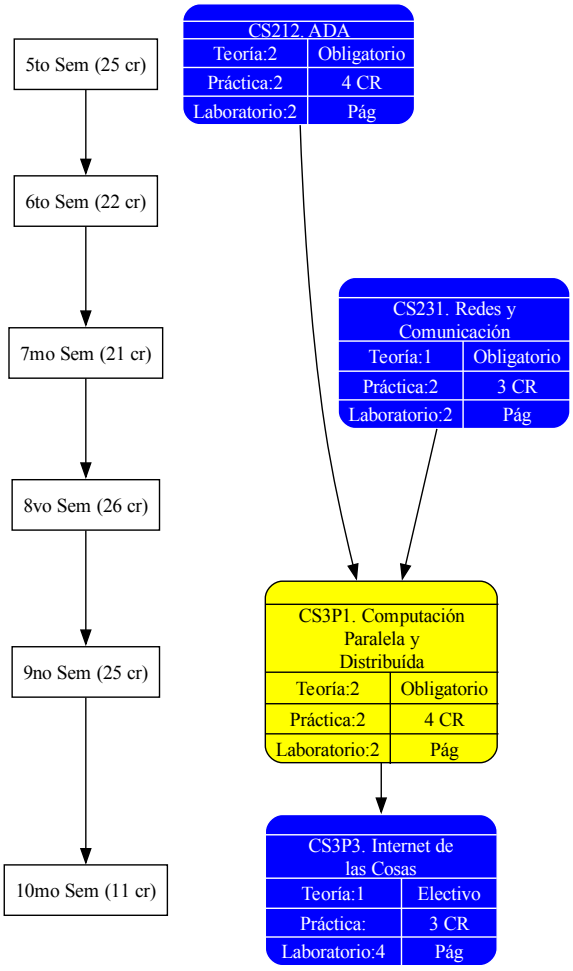


Figura 3.55: Mapa de Conexión. CS3P1 Computación Paralela y Distribuída

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*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Programas: Paralelismo Declarativo, 4 hr(s)

Competencias: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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: AG-C09,AG-C11

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.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS400 Prácticas Pre-profesionales (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 2 horas;
- **Prerrequisitos**:
  - 110Cr

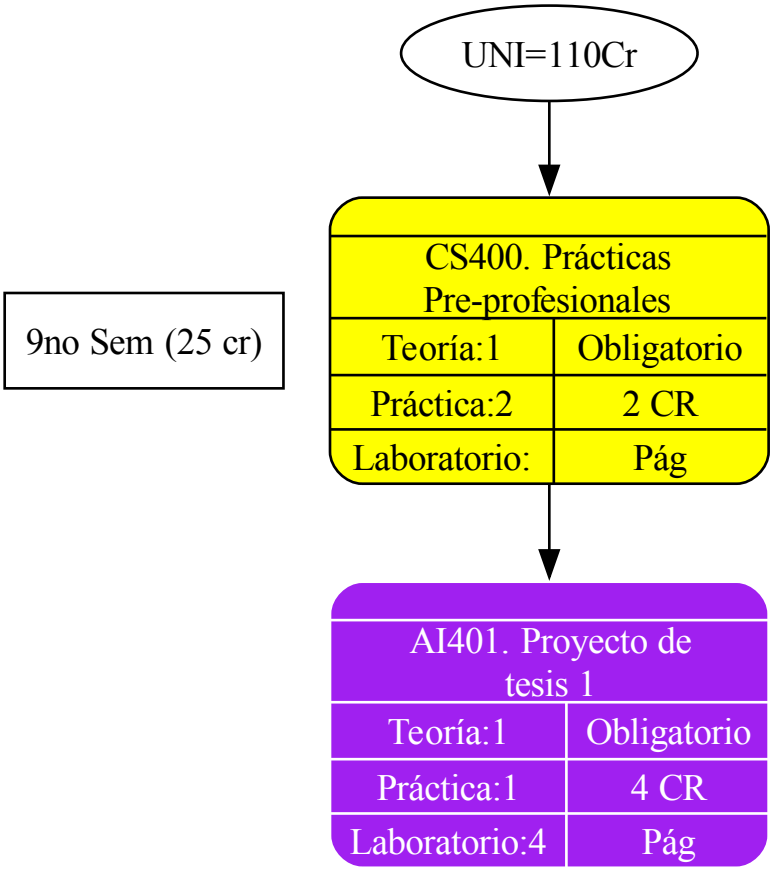


Figura 3.56: Mapa de Conexión. CS400 Prácticas Pre-profesionales

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*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Usar)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Usar)

AG-C06) Aprendizaje a lo largo de la vida: Reconoce la importancia del aprendizaje continuo y la adaptación a nuevas tecnologías. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Evaluación de Desempeño en Entorno Laboral, 48 hr(s)

Competencias: AG-C02,AG-C04,AG-C05,AG-C09

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: [ACM Education Committee, 2022, IEEE Computer Society, 2021]

6. Bibliografía

[ACM Education Committee, 2022] ACM Education Committee (2022). Guide to successful pre-professional internships in computer science. Technical report, Association for Computing Machinery. Official document with standards for internships in computing.

[IEEE Computer Society, 2021] IEEE Computer Society (2021). *Professional Practices in Computing: Ethics and Management*. IEEE Press. Best-practices manual for IT work environments.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS403 Proyecto de tesis 2 (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)

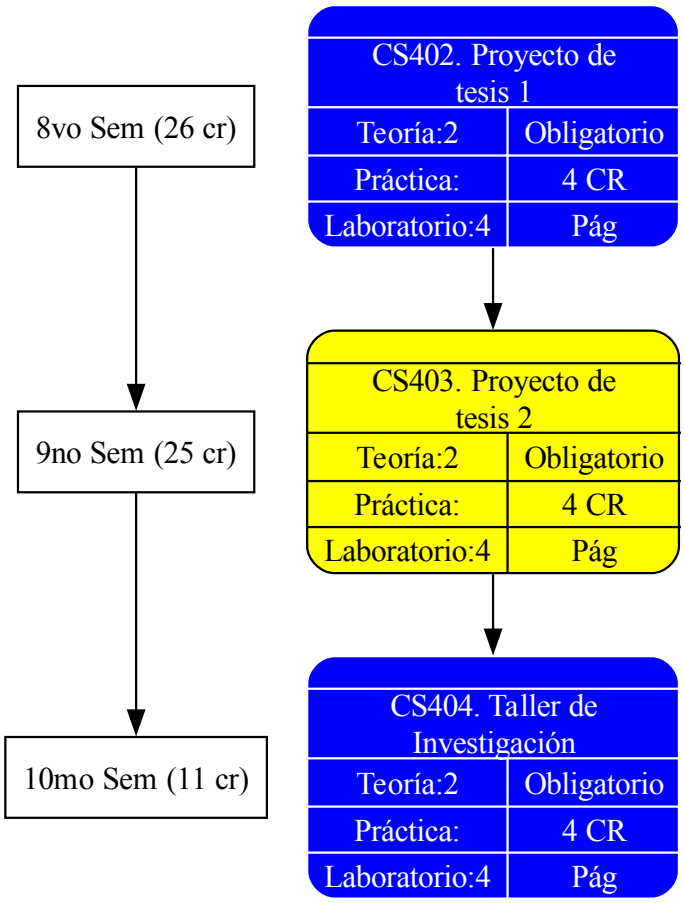


Figura 3.57: Mapa de Conexión. CS403 Proyecto de tesis 2

2. Fundamentación

Este curso tiene por objetivo que el alumno desarrolle su propuesta de investigación y ejecute los primeros experimentos formales, produciendo los capítulos centrales de su tesis bajo la supervisión del asesor.

3. Objetivos del curso.

- Formalizar el plan de tesis en el formato institucional con objetivos, cronograma y análisis de viabilidad.

- Redactar el marco teórico y el capítulo de trabajos relacionados con síntesis crítica de la literatura.
- Diseñar la metodología original e implementar un prototipo funcional de la propuesta de investigación.
- Ejecutar experimentos preliminares comparativos con rigor estadístico e interpretación fundamentada de resultados.
- Presentar y defender el avance de tesis ante el comité evaluador incorporando su retroalimentación.
- Los entregables de este curso son:

  Avance parcial: Plan de tesis en formato institucional, capítulo de marco teórico y trabajos relacionados en LaTeX.

  Final: Capítulos de metodología, propuesta y experimentos preliminares aprobados por el asesor.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Formalización del Plan de Tesis, 10 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Estructura del plan de tesis según el formato institucional. **II:** Definición precisa del problema de investigación y pregunta central de tesis. **III:** Formulación de objetivos generales y específicos verificables. **IV:** Cronograma de actividades con hitos hasta la defensa final de tesis. **V:** Análisis de viabilidad: recursos, tiempos, riesgos y contingencias del proyecto.

Objetivos de Aprendizaje I: Producir el plan de tesis en el formato exigido por la institución académica. [Evaluar] **II:** Definir un problema de investigación original y objetivos verificables. [Evaluar] **III:** Elaborar un cronograma detallado con hitos y análisis de riesgos hasta la defensa de tesis. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.2 Marco Teórico y Estado del Arte Avanzado, 10 hr(s)

Competencias: AG-C04,AG-C08,AG-C09

Tópicos: I: Organización temática y síntesis crítica de la literatura del área de tesis. **II:** Redacción del capítulo de marco teórico en LaTeXcon estructura coherente. **III:** Identificación y justificación de las brechas de investigación del problema. **IV:** Análisis comparativo de enfoques: tablas y diagramas de clasificación de la literatura. **V:** Integración de referencias con BibTeX y gestión de versiones con Git.

Objetivos de Aprendizaje I: Redactar el capítulo de marco teórico con síntesis estructurada y crítica de la literatura. [Evaluar] **II:** Justificar la originalidad del problema en relación con el estado del arte. [Evaluar] **III:** Construir tablas comparativas que evidencien las brechas de investigación. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Diseño de la Metodología Original, 10 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Definición del enfoque metodológico: experimental, analítico o constructivo. **II:** Especificación de la contribución original: algoritmo, sistema, modelo o metodología. **III:** Diseño del protocolo experimental: variables, métricas, datasets y benchmarks. **IV:** Criterios de validación y generalización de resultados de la propuesta. **V:** Alineación de la metodología con los objetivos del plan de tesis.

Objetivos de Aprendizaje I: Diseñar la metodología de investigación alineada con los objetivos. [Evaluar] **II:** Especificar la contribución original del proyecto con criterios verificables. [Evaluar] **III:** Definir el protocolo experimental completo con variables, métricas y benchmarks. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Implementación y Experimentos Preliminares, 12 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Implementación de la propuesta original: prototipo, algoritmo o modelo de tesis. **II:** Reproducción de baselines relevantes del estado del arte para comparación. **III:** Ejecución de experimentos preliminares con análisis estadístico básico. **IV:** Registro y documentación de configuraciones, versiones y resultados. **V:** Identificación de limitaciones y ajuste iterativo de la propuesta.

Objetivos de Aprendizaje I: Implementar un prototipo funcional de la propuesta original de tesis. [Usar] **II:** Ejecutar experimentos preliminares comparativos con baselines definidos. [Usar] **III:** Analizar e interpretar resultados con rigor estadístico apropiado. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Redacción de Capítulos Centrales de la Tesis, 10 hr(s)

Competencias: AG-C02,AG-C04,AG-C09

Tópicos: I: Redacción del capítulo de metodología y diseño experimental en LaTeX. **II:** Redacción del capítulo de trabajos relacionados con análisis crítico comparativo. **III:** Presentación de resultados preliminares: tablas, figuras y análisis comparativo. **IV:** Ciclos de revisión con el asesor: incorporación de retroalimentación estructurada. **V:** Coherencia y consistencia narrativa entre capítulos de la tesis.

Objetivos de Aprendizaje I: Redactar los capítulos de metodología y trabajos relacionados con rigor académico. [Evaluar] **II:** Presentar resultados preliminares con visualizaciones claras y análisis comparativo. [Usar] **III:** Aplicar ciclos de revisión con el asesor para mejorar la calidad del documento. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.6 Presentación del Avance y Planificación Final, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: **I:** Presentación oral del avance de tesis ante el comité evaluador. **II:** Defensa de metodología y resultados preliminares ante preguntas del comité. **III:** Incorporación de retroalimentación del comité en la revisión del documento. **IV:** Planificación de actividades restantes hacia la defensa final de tesis. **V:** Gestión ética: autoría, integridad académica y confidencialidad de datos.

Objetivos de Aprendizaje **I:** Presentar y defender oralmente el avance de tesis ante un comité académico. [Evaluar] **II:** Incorporar la retroalimentación del comité en una revisión fundamentada del documento. [Evaluar] **III:** Planificar las actividades restantes con objetivos claros hacia la defensa final. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library*. Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library*. College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library*. IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI402 Proyecto de tesis 2 (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS401 Metodología de la Investigación (4to Sem-Pág 152)

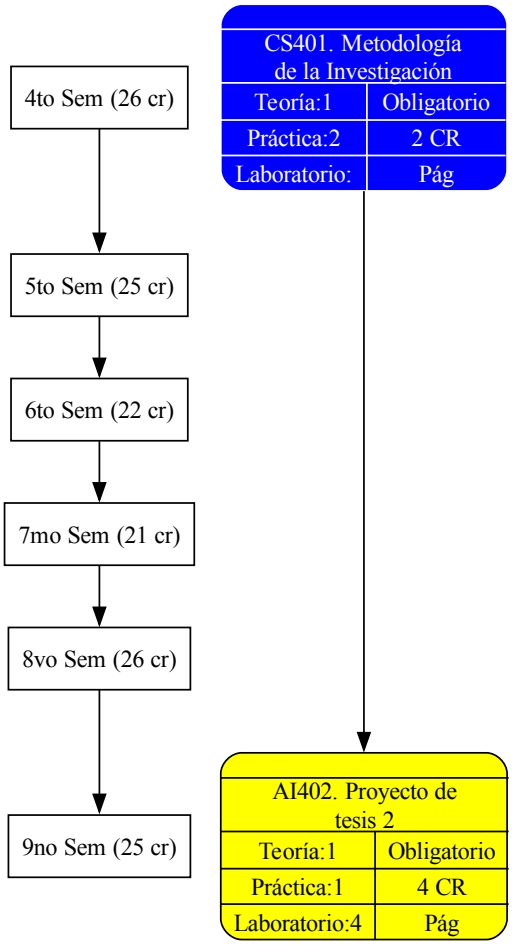


Figura 3.58: Mapa de Conexión. AI402 Proyecto de tesis 2

2. Fundamentación

Este curso tiene por objetivo que el alumno conduzca un estudio profundo del estado del arte de un tópico específico elegido para su proyecto de tesis. El alumno deberá producir un reporte técnico con experimentos preliminares comparativos y formular una propuesta de investigación, culminando junto a un asesor especializado en el área elegida.

3. Objetivos del curso.

- Realizar un estudio profundo del estado del arte en un tópico específico del área de computación.

- Demostrar dominio de la literatura relevante mediante análisis crítico y comparativo de enfoques existentes.
- Diseñar y ejecutar experimentos preliminares que reproduzcan y comparen técnicas del estado del arte.
- Redactar un reporte técnico de calidad con marco teórico, metodología y resultados experimentales.
- Formular una propuesta de investigación con pregunta, objetivos y cronograma.
- Seleccionar un asesor de tesis que domine el área de investigación elegida.
- Los entregables de este curso son:

  Avance parcial: Bibliografía sólida y borrador del marco teórico en formato LaTeX.

  Final: Reporte Técnico con experimentos preliminares comparativos y propuesta de investigación aprobada por el asesor.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Profundización en el Estado del Arte Específico, 10 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Selección y refinamiento del tópico de investigación en coordinación con el asesor. **II:** Búsqueda exhaustiva multi-fuente: IEEE Xplore, ACM Digital Library, Google Scholar, arXiv y bases de datos especializadas. **III:** Identificación de trabajos seminales y artículos más citados del área. **IV:** Construcción del grafo de citaciones y análisis de la evolución temporal del tópico. **V:** Identificación de subáreas, variantes y enfoques principales del campo.

Objetivos de Aprendizaje I: Construir un mapa completo de la literatura de un tópico específico identificando los trabajos fundamentales. [Evaluar] **II:** Analizar la evolución temporal de un área de investigación identificando tendencias y brechas. [Evaluar] **III:** Organizar la literatura en subcategorías coherentes que reflejen la estructura del campo. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.2 Análisis Crítico y Comparativo de Enfoques, 10 hr(s)

Competencias: AG-C08,AG-C09

Tópicos: **I:** Análisis crítico de propuestas: metodología, datasets, métricas y limitaciones de cada enfoque. **II:** Construcción de tablas comparativas de propuestas existentes con criterios objetivos. **III:** Identificación de brechas de investigación (research gaps) no resueltas en el área. **IV:** Formulación de la pregunta de investigación propia a partir del análisis del estado del arte. **V:** Marco teórico: síntesis estructurada y coherente del conocimiento relevante del área.

Objetivos de Aprendizaje I: Comparar sistemáticamente múltiples enfoques usando criterios objetivos y cuantificables. [Evaluar] **II:** Identificar brechas de investigación no resueltas en un área específica de computación. [Evaluar] **III:** Formular una pregunta de investigación propia sustentada en el análisis del estado del arte. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Diseño y Ejecución de Experimentos Preliminares, 12 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Diseño experimental: variables, métricas y posibles benchmarks del área. **II:** Reproducción de resultados de trabajos relevantes del estado del arte. **III:** Implementación de baselines para comparación objetiva de enfoques. **IV:** Análisis estadístico básico: pruebas de significancia e intervalos de confianza. **V:** Gestión de datos experimentales: datasets, configuraciones, versiones y registro de resultados.

Objetivos de Aprendizaje I: Reproducir los resultados de al menos dos trabajos relevantes del estado del arte del área. [Usar] **II:** Diseñar y ejecutar un experimento comparativo preliminar con baselines definidos. [Usar] **III:** Interpretar y reportar resultados experimentales con rigor estadístico apropiado. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Redacción del Reporte Técnico, 12 hr(s)

Competencias: AG-C04,AG-C09

Tópicos: I: Estructura del reporte técnico: introducción, marco teórico, metodología, experimentos, resultados y conclusiones. **II:** Redacción del marco teórico: organización temática y síntesis crítica de la literatura. **III:** Documentación de la metodología y configuración experimental en LaTeX. **IV:** Presentación de resultados: tablas, gráficas y análisis comparativos de alta calidad. **V:** Ciclos de revisión: escritura, revisión con asesor e incorporación de retroalimentación.

Objetivos de Aprendizaje I: Redactar un reporte técnico completo con marco teórico y experimentos preliminares. [Usar] **II:** Estructurar y presentar resultados experimentales con visualizaciones claras y precisas. [Usar] **III:** Aplicar ciclos de revisión para mejorar la calidad, coherencia y rigor del documento. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Propuesta de Investigación y Selección de Asesor, 8 hr(s)

Competencias: AG-C02,AG-C05,AG-C09

Tópicos: I: Estructura de una propuesta de investigación: problema, motivación y objetivos. **II:** Definición del alcance del proyecto de tesis: metodología, contribución esperada y cronograma. **III:** Análisis de viabilidad: recursos, tiempo, riesgos y contingencias del proyecto. **IV:** Criterios para la selección del asesor de tesis: experiencia, disponibilidad y alineación temática. **V:** Proceso de formalización del vínculo investigador-asesor en el programa académico.

Objetivos de Aprendizaje I: Elaborar una propuesta de investigación que defina el problema, objetivos y posible solución. [Evaluar] **II:** Seleccionar un asesor con experiencia demostrada en el área de investigación elegida. [Usar] **III:** Definir un plan de tesis viable con objetivos, alcance, cronograma y análisis de riesgos. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.6 Presentación y Defensa del Estado del Arte, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: **I:** Preparación de la presentación oral del reporte técnico ante el comité evaluador. **II:** Técnicas de defensa ante preguntas y cuestionamientos de un comité académico. **III:** Elaboración de póster técnico del estado del arte para presentación en simposio. **IV:** Incorporación de retroalimentación del comité evaluador en la revisión final del reporte. **V:** Planificación de la transición a CS403: objetivos, contribución esperada y siguientes pasos.

Objetivos de Aprendizaje **I:** Presentar y defender oralmente el reporte técnico ante un comité académico con rigor. [Evaluar] **II:** Incorporar constructivamente la retroalimentación del comité en una revisión del reporte. [Usar] **III:** Planificar la transición hacia el proyecto de tesis (CS403) con objetivos y pasos claros. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library*. Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library*. College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library*. IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial
CB309 Bioinformática (Obligatorio)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Práctica**: 4 horas;
- **Prerrequisitos**:
  - BI101 Biología (8vo Sem-Pág 253)

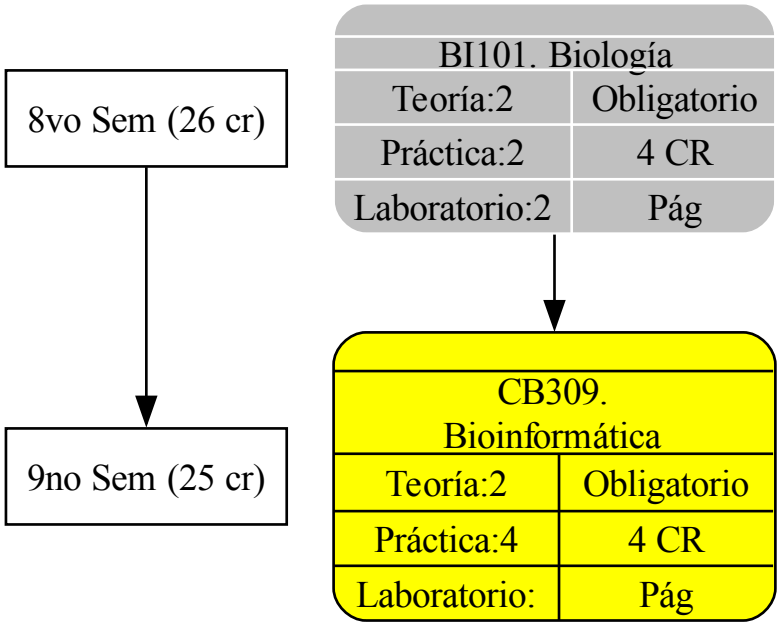


Figura 3.59: Mapa de Conexión. CB309 Bioinformática

2. Fundamentación

El uso de métodos computacionales en las ciencias biológicas se ha convertido en una de las herramientas claves para el campo de la biología molecular, siendo parte fundamental en las investigaciones de esta área.
En Biología Molecular, existen diversas aplicaciones que involucran tanto al ADN, al análisis de proteínas o al secuenciamiento del genoma humano, que dependen de métodos computacionales. Muchos de estos problemas son realmente complejos y tratan con grandes conjuntos de datos.
Este curso puede ser aprovechado para ver casos de uso concretos de varias áreas de conocimiento de Ciencia de la Computación como: Lenguajes de Programación (PL), Algoritmos y Complejidad (AL), Probabilidades y Estadística, Manejo de Información (IM), Inteligencia Artificial (IA).

3. Objetivos del curso.

- Que el alumno tenga un conocimiento sólido de los problemas biológicos moleculares que desafían a la computación.
- Que el alumno sea capaz de abstraer la esencia de los diversos problemas biológicos para plantear soluciones usando sus conocimientos de Ciencia de la Computación

4. Competencias Generales (*Outcomes*)

AG-C10) Indagación: Estudia problemas complejos de computación usando métodos de ciencias de la información. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Introducción a la Biología Molecular, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Revisión de la química orgánica: moléculas y macromoléculas, azúcares, acidos nucleicos, nuclótidos, ARN, ADN, proteínas, aminoácidos y nivels de estructura en las proteinas. **II:** El dogma de la vida: del ADN a las proteinas, transcripción, traducción, síntesis de proteinas **III:** Estudio del genoma: Mapas y secuencias, técnicas específicas

Objetivos de Aprendizaje **I:** Lograr un conocimiento general de los tópicos más importantes en Biología Molecular. [Familiarizarse] **II:** Entender que los problemas biológicos son un desafío al mundo computacional. [Evaluar]

Bibliografía: [Clote and Backofen, 2000, Setubal and Meidanis, 1997]

5.2 Comparación de Secuencias, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Secuencias de nucléotidos y secuencias de aminoácidos. **II:** Alineamiento de secuencias, el problema de alineamiento por pares, búsqueda exhaustiva, Programación dinámica, alineamiento global, alineamiento local, penalización por gaps **III:** Comparación de múltiples secuencias: suma de pares, análisis de complejidad por programación dinámica, heurísticas de alineamiento, algoritmo estrella, algoritmos de alineamiento progresivo.

Objetivos de Aprendizaje **I:** Entender y solucionar el problema de alineamiento de un par de secuencias. [Usar] **II:** Comprender y solucionar el problema de alineamiento de múltiples secuencias. [Usar] **III:** Conocer los diversos algoritmos de alineamiento de secuencias existentes en la literatura. [Familiarizarse]

Bibliografía: [Clote and Backofen, 2000, Setubal and Meidanis, 1997, Pevzner, 2000]

5.3 Árboles Filogenéticos, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Filogenia: Introducción y relaciones filogenéticas. **II:** Arboles Filogenéticos: definición, tipo de árboles, problema de búsqueda y reconstrucción de árboles **III:** Métodos de Reconstrucción: métodos por parsimonia, métodos por distancia, métodos por máxima verosimilitud, confianza de los árboles reconstruidos

Objetivos de Aprendizaje **I:** Comprender el concepto de filogenia, árboles filogenéticos y la diferencia metodológica entre biología y biología molecular. [Familiarizarse] **II:** Comprender el problema de reconstrucción de árboles filogenéticos, conocer y aplicar los principales algoritmos para reconstrucción de árboles filogenéticos. [Evaluar]

Bibliografía: [Clote and Backofen, 2000, Setubal and Meidanis, 1997, Pevzner, 2000]

5.4 Ensamblaje de Secuencias de ADN, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Fundamento biológico: caso ideal, dificultades, métodos alternativos para secuenciamiento de ADN **II:** Modelos formales de ensamblaje: *Shortest Common Superstring*, *Reconstruction*, *Multicontig* **III:** Algoritmos para ensamblaje de secuencias: representación de overlaps, caminos para crear *superstrings*, algoritmo voraz, grafos acíclicos. **IV:** Heurísticas para ensamblaje: búsqueda de sobreposiciones, ordenación de fragmentos, alineamientos y consenso.

Objetivos de Aprendizaje **I:** Comprender el desafío computacional que ofrece el problema de Ensamblaje de Secuencias. [Familiarizarse] **II:** Entender el principio de modelo formal para ensamblaje. [Evaluar] **III:** Conocer las principales heurísticas para el problema de ensambjale de secuencias ADN [Usar]

Bibliografía: [Setubal and Meidanis, 1997, Aluru, 2006]

5.5 Estructuras secundarias y terciarias, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Estructuras moleculares: primaria, secundaria, terciaria, cuaternaria. **II:** Predicción de estructuras secundarias de ARN: modelo formal, energia de pares, estructuras con bases independientes, solución con Programación Dinámica, estructuras con bucles. **III:** *Protein folding*: Estructuras en proteinas, problema de *protein folding*. **IV:** *Protein Threading*: Definiciones, Algoritmo *Branch & Bound*, *Branch & Bound* para *protein threading*. **V:** *Structural Alignment*: definiciones, algoritmo DALI

Objetivos de Aprendizaje **I:** Conocer las estructuras protéicas y la necesidad de métodos computacionales para la predicción de la geometría. [Familiarizarse] **II:** Conocer los algoritmos de solución de problemas de predicción de estructuras secundarias ARN, y de estructuras en proteínas. [Evaluar]

Bibliografía: [Setubal and Meidanis, 1997, Clote and Backofen, 2000, Aluru, 2006]

5.6 Modelos Probabilísticos en Biología Molecular, 4 hr(s)

Competencias: AG-C10,AG-C11

Tópicos: **I:** Probabilidad: Variables aleatorias, Cadenas de Markov, Algoritmo de Metropoli-Hasting, Campos Aleatorios de Markov y Muestreador de Gibbs, Máxima Verosimilitud. **II:** Modelos Ocultos de Markov (HMM), estimación de parámetros, algoritmo de Viterbi y método Baul-Welch, Aplicación en alineamientos de pares y múltiples, en detección de Motifs en proteínas, en ADN eucariótico, en familias de secuencias. **III:** Filogenia Probabilística: Modelos probabilísticos de evolución, verosimilitud de alineamientos, verosimilitud para inferencia, comparación de métodos probailísticos y no probabilísticos

Objetivos de Aprendizaje **I:** Revisar conceptos de Modelos Probabilísticos y comprender su importancia en Biología Molecular Computacional. [Evaluar] **II:** Conocer y aplicar Modelos Ocultos de Markov para varios análisis en Biología Molecular. [Usar] **III:** Conocer la aplicación de modelos probabilísticos en Filogenia y comparalos con modelos no probabilísticos [Evaluar]

Bibliografía: [R et al., 1998, Clote and Backofen, 2000, Aluru, 2006, Krogh et al., 1994]

6. Bibliografía

[Aluru, 2006] Aluru, S., editor (2006). *Handbook of Computational Molecular Biology*. Computer and Information Science Series. Chapman & Hall, CRC, Boca Raton, FL.

[Clote and Backofen, 2000] Clote, P. and Backofen, R. (2000). *Computational Molecular Biology: An Introduction*. John Wiley & Sons Ltd. 279 pages.

[Krogh et al., 1994] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden markov models in computational biology, applications to protein modeling. *J Mol. Biol*, 235:1501–1531.

[Pevzner, 2000] Pevzner, P. A. (2000). *Computational Molecular Biology: an Algorithmic Approach*. The MIT Press, Cambridge, Massachusetts.

[R et al., 1998] R, D., S.R, E., A, K., and G, M. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

[Setubal and Meidanis, 1997] Setubal, J. C. and Meidanis, J. (1997). *Introduction to computational molecular biology*. Boston: PWS Publishing Company.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI268 Visión Computacional (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)

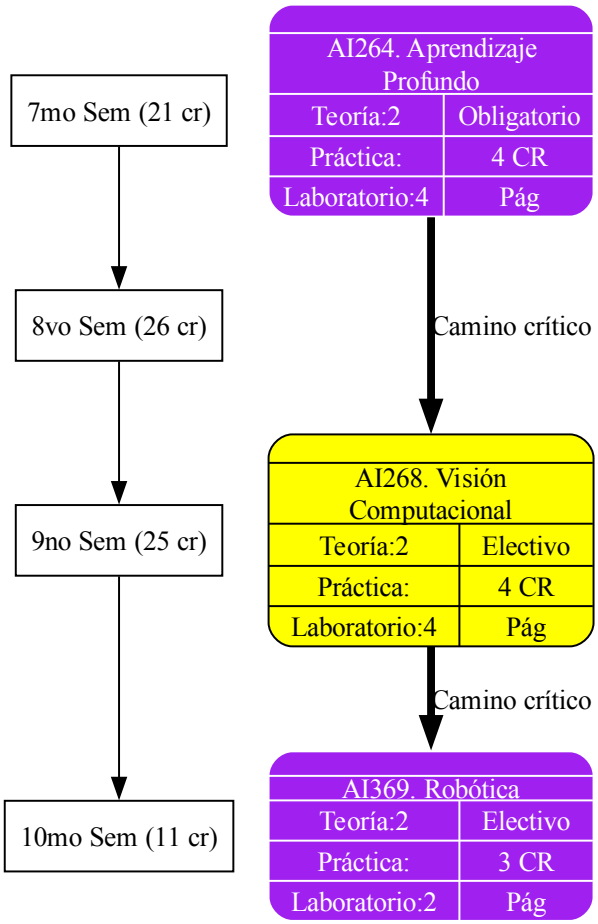


Figura 3.60: Mapa de Conexión. AI268 Visión Computacional

2. Fundamentación

Este curso cubre técnicas fundamentales para el análisis automático de imágenes digitales, esencial en aplicaciones como diagnóstico médico, vehículos autónomos y sistemas de vigilancia. Alinea con los estándares ACM/ IEEE-CS para visión por computadora.

3. Objetivos del curso.

- Implementar algoritmos de extracción de características y reconocimiento de objetos usando OpenCV/Python.

- Evaluar métodos de deep learning para segmentación semántica (ej. Mask R-CNN).

4. Competencias Generales (*Outcomes*)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Fundamentos de Imágenes Digitales, 16 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: I: Filtrado espacial (Gaussiano, Sobel) **II:** Transformaciones geométricas y morfológicas **III:** Espacios de color (RGB, HSV, LAB)

Objetivos de Aprendizaje I: Aplicar operaciones básicas de procesamiento de imágenes [Usar] **II:** Calibrar parámetros de filtros para casos reales [Evaluar]

Bibliografía: [Szeliski, 2010, Gonzalez and Woods, 2018]

5.2 Geometría Epipolar y Reconstrucción, 16 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: I: Matriz fundamental y esencial **II:** Triangulación y estructura desde movimiento **III:** Nube de puntos con Open3D

Objetivos de Aprendizaje I: Implementar pipelines de reconstrucción 3D [Usar] **II:** Documentar resultados técnicos en informes [Evaluar]

Bibliografía: [Hartley and Zisserman, 2004]

5.3 Redes Neuronales para Visión, 16 hr(s)

Competencias: AG-C08,AG-C11

Tópicos: I: Arquitecturas CNN (ResNet, YOLO) **II:** Transfer learning con TensorFlow **III:** Segmentación semántica (U-Net)

Objetivos de Aprendizaje I: Entrenar modelos para clasificación de imágenes [Usar] **II:** Colaborar en equipos para proyectos integrados [Usar]

Bibliografía: [He et al., 2017]

6. Bibliografía

[Gonzalez and Woods, 2018] Gonzalez, R. and Woods, R. (2018). *Digital Image Processing*. Pearson, 4th edition.

[Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. *IEEE ICCV*.

[Szeliski, 2010] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI363 Sistemas Autónomos (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)

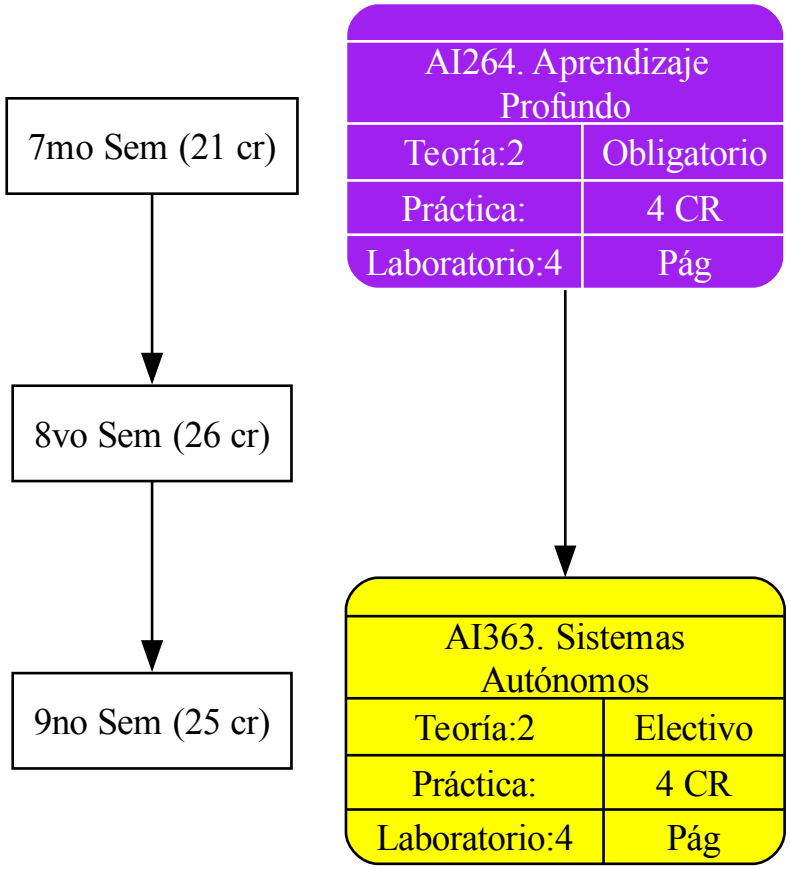


Figura 3.61: Mapa de Conexión. AI363 Sistemas Autónomos

2. Fundamentación

Este curso cubre los principios de sistemas autónomos capaces de percibir, decidir y actuar independientemente, incluyendo arquitecturas de decisión, navegación autónoma y consideraciones de seguridad crítica.

3. Objetivos del curso.

- Implementar pipelines de percepción-decisión-acción

- Diseñar sistemas de planificación de movimientos
- Evaluar riesgos en sistemas autónomos de seguridad crítica

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

AG-C12) Aplica teorías de IA, algoritmos cognitivos y principios de *Data Engineering* para construir sistemas inteligentes autónomos. (Evaluar)

5. Contenido

5.1 Fundamentos de Autonomía, 12 hr(s)

Competencias: AG-C09

Tópicos: I: Arquitecturas de control (sense-plan-act vs behavior-based) **II:** Localización y mapeo simultáneo (SLAM) **III:** Filtros bayesianos (Kalman, Particle)

Objetivos de Aprendizaje I: Implementar filtros de localización [Usar] **II:** Evaluar arquitecturas de control [Evaluar]

Bibliografía: [Thrun et al., 2005]

5.2 Toma de Decisiones Autónomas, 24 hr(s)

Competencias: AG-C09

Tópicos: I: Redes neuronales para control end-to-end **II:** Planificación de trayectorias (A*, RRT*) **III:** Aprendizaje por refuerzo en robótica

Objetivos de Aprendizaje I: Entrenar redes para control vehicular [Usar] **II:** Diseñar sistemas híbridos lógica-IA [Evaluar]

Bibliografía: [Bojarski et al., 2016]

5.3 Seguridad y Ética, 8 hr(s)

Competencias: AG-C09

Tópicos: I: Fallas en cascada en sistemas autónomos **II:** Mecanismos de "kill switch"éticos **III:** Regulaciones para vehículos autónomos

Objetivos de Aprendizaje I: Analizar casos de fallas críticas [Familiarizarse] **II:** Implementar salvaguardas básicas [Usar]

Bibliografía: [Amodei et al., 2016]

6. Bibliografía

[Amodei et al., 2016] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. In *NeurIPS*.

[Bojarski et al., 2016] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. (2016). End to end learning for self-driving cars. *arXiv:1604.07316*.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI365 Modelos Generativos Avanzados en IA (Electivo)

1. Información general

- **Semestre**: 9no Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - AI264 Aprendizaje Profundo (7mo Sem-Pág 228)

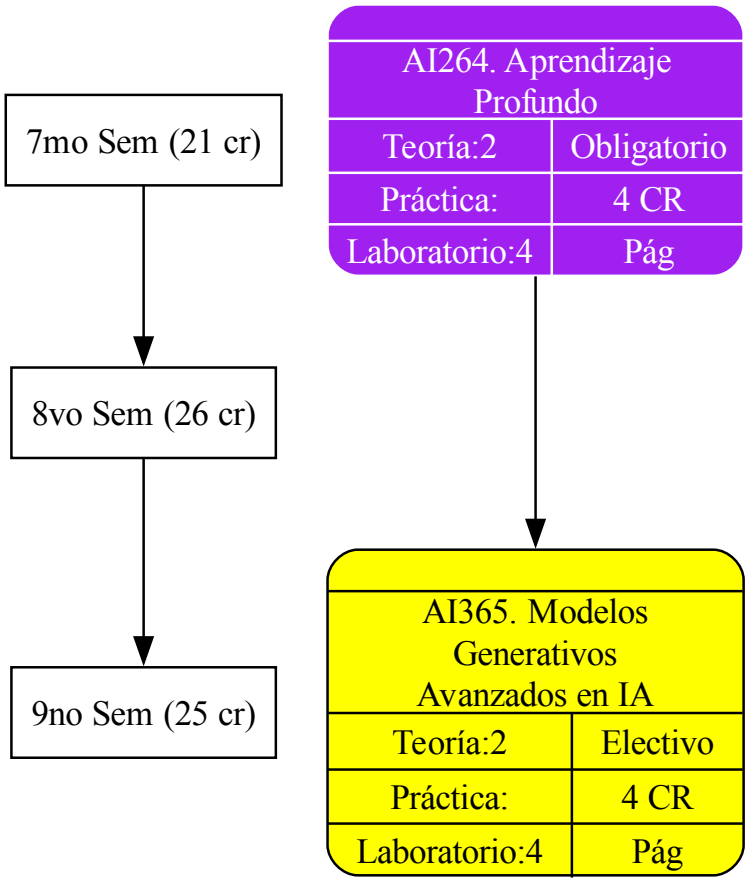


Figura 3.62: Mapa de Conexión. AI365 Modelos Generativos Avanzados en IA

2. Fundamentación

Este curso explora modelos generativos modernos (GANs, Diffusion Models, LLMs) para creación de contenido multimodal, abordando tanto fundamentos matemáticos como aplicaciones prácticas y consideraciones éticas en generación sintética.

3. Objetivos del curso.

- Implementar pipelines de generación de imágenes/texto/video

- Diseñar sistemas de edición asistida por IA
- Evaluar riesgos de contenido sintético

4. Competencias Generales (*Outcomes*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Evaluar)

5. Contenido

5.1 Fundamentos Generativos, 12 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Teoría de GANs (minimax, Wasserstein) **II:** Autoregressive models (PixelCNN, Transformers) **III:** Métricas de evaluación (FID, Inception Score)

Objetivos de Aprendizaje I: Entrenar GANs básicas [Usar] **II:** Analizar modos de falla en generación [Evaluar]

Bibliografía: [Goodfellow et al., 2014]

5.2 Modelos de Difusión, 24 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Procesos de difusión (forward/reverse) **II:** Stable Diffusion y arquitecturas latentes **III:** Técnicas de control (ControlNet, LoRA)

Objetivos de Aprendizaje I: Implementar samplers de difusión [Usar] **II:** Fine-tuning para dominios específicos [Evaluar]

Bibliografía: [Sohl-Dickstein et al., 2023, Rombach et al., 2022]

5.3 Ética en Medios Sintéticos, 8 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Deepfakes y desinformación **II:** Derechos de autor en contenido generado **III:** Marcado de autenticidad (C2PA, watermarking)

Objetivos de Aprendizaje I: Detectar contenido sintético [Familiarizarse] **II:** Implementar sistemas de trazabilidad [Usar]

Bibliografía: [Sohl-Dickstein et al., 2023]

6. Bibliografía

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. In *NeurIPS*.

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *CVPR*.

[Sohl-Dickstein et al., 2023] Sohl-Dickstein, J., Song, Y., and Ermon, S. (2023). *Diffusion Models: A Comprehensive Practical Guide*. AI Press.

Décimo Semestre

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS404 Taller de Investigación (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS403 Proyecto de tesis 2 (9no Sem-Pág 276)

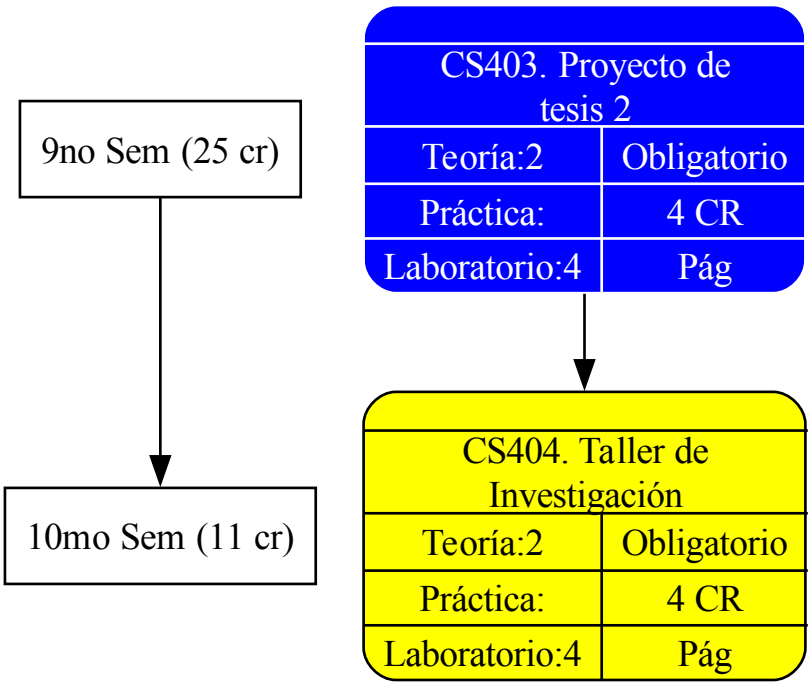


Figura 3.63: Mapa de Conexión. CS404 Taller de Investigación

2. Fundamentación

Este curso tiene por objetivo que el alumno concluya su tesis con calidad suficiente para la sustentación pública, completando experimentos, capítulos finales, revisiones con el asesor y la entrega formal.

3. Objetivos del curso.

- Concluir la parte experimental de la tesis con rigor estadístico y análisis completo de resultados.
- Redactar los capítulos finales de la tesis: conclusiones, contribuciones y trabajo futuro.
- Asegurar que el documento cumpla con el formato institucional y tenga coherencia narrativa completa.
- Incorporar las revisiones del asesor hasta obtener un documento aprobado para sustentación.

- Preparar y realizar la presentación formal de la tesis ante el comité de la facultad.
- Los entregables de este curso son:

  Parcial: Borrador avanzado con experimentos concluidos, capítulos de introducción, marco teórico, propuesta y resultados en LaTeX.

  Final: Documento de tesis completo, aprobado por el asesor y listo para sustentación pública.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Conclusión de Experimentos y Análisis de Resultados, 12 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Ejecución completa del protocolo experimental definido en el plan de tesis. **II:** Análisis estadístico final: pruebas de significancia, intervalos de confianza y efecto. **III:** Comparación objetiva con baselines y estado del arte usando métricas establecidas. **IV:** Interpretación de resultados: hallazgos, limitaciones y amenazas a la validez. **V:** Documentación reproducible: código, datasets, configuraciones y resultados.

Objetivos de Aprendizaje I: Completar la ejecución del protocolo experimental con todos los casos definidos. [Evaluar] **II:** Analizar resultados con rigor estadístico e interpretar hallazgos con respecto a la hipótesis. [Evaluar] **III:** Documentar de forma reproducible los experimentos, configuraciones y resultados. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.2 Redacción de Capítulos Finales de la Tesis, 12 hr(s)

Competencias: AG-C02,AG-C04,AG-C09

Tópicos: I: Redacción del capítulo de resultados: tablas, figuras y análisis comparativo de alta calidad. **II:** Redacción del capítulo de conclusiones: hallazgos, contribuciones originales y limitaciones. **III:** Redacción del capítulo de trabajo futuro: extensiones y problemas abiertos. **IV:** Redacción de la introducción final: motivación, problema y resumen de contribuciones. **V:** Redacción del resumen ejecutivo (abstract) en español e inglés.

Objetivos de Aprendizaje I: Redactar el capítulo de resultados con visualizaciones claras y análisis comparativo. [Evaluar] **II:** Formular las conclusiones y contribuciones originales de la tesis con precisión. [Evaluar] **III:** Escribir el abstract y la introducción final articulando el problema y los aportes. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.3 Formato, Integración y Calidad del Documento, 10 hr(s)

Competencias: AG-C02,AG-C04,AG-C09

Tópicos: I: Verificación del cumplimiento del formato institucional de tesis en LaTeX. **II:** Revisión de calidad de figuras, tablas y ecuaciones en el documento final. **III:** Consistencia narrativa y terminológica entre todos los capítulos del documento. **IV:** Revisión y actualización de la bibliografía: cobertura, formato BibTeX y citas correctas. **V:** Generación del documento final: PDF, portada, índices y apéndices.

Objetivos de Aprendizaje I: Verificar que el documento cumple íntegramente con el formato institucional de tesis. [Evaluar] **II:** Corregir la consistencia narrativa y terminológica de todos los capítulos. [Evaluar] **III:** Producir el documento PDF final con portada, índices y bibliografía completos. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Revisiones con el Asesor y Ciclos de Corrección, 10 hr(s)

Competencias: AG-C02,AG-C08,AG-C09

Tópicos: I: Dinámica de revisión estructurada: entrega, revisión y respuesta al asesor. **II:** Incorporación de observaciones del asesor: priorización y respuesta argumentada. **III:** Revisión de la solidez de la propuesta y la validez de los experimentos según el asesor. **IV:** Manejo de desacuerdos académicos con el asesor de forma constructiva y ética. **V:** Criterios de aprobación del asesor para la autorización de sustentación.

Objetivos de Aprendizaje I: Incorporar sistemáticamente las observaciones del asesor con respuestas argumentadas. [Evaluar] **II:** Mejorar la solidez científica del documento a través de ciclos de revisión. [Evaluar] **III:** Obtener la aprobación formal del asesor para la sustentación del documento. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Preparación y Ensayo de la Sustentación, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: I: Diseño de la presentación oral: estructura, narrativa y manejo del tiempo. **II:** Preparación de diapositivas técnicas: figuras, resultados y comparaciones clave. **III:** Técnicas de defensa ante preguntas del jurado: argumentación y manejo de objeciones. **IV:** Ensayo de la sustentación con el asesor y/o pares académicos. **V:** Gestión del estrés y comunicación efectiva en contextos de evaluación académica.

Objetivos de Aprendizaje I: Diseñar y exponer una presentación oral clara y estructurada del trabajo de tesis. [Evaluar] **II:** Defender la metodología y resultados ante preguntas críticas del jurado. [Evaluar] **III:** Demostrar dominio completo del tema mediante ensayos de sustentación previos. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.6 Entrega Formal y Pre-Sustentación, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: I: Proceso institucional de entrega de la tesis: documentación y plazos reglamentarios. **II:** Pre-sustentación ante autoridades de la facultad: evaluación del borrador final. **III:** Incorporación de correcciones del comité pre-sustentador en el documento. **IV:** Trámites administrativos: designación de jurado y programación de la sustentación. **V:** Ética en la diseminación: acceso abierto, repositorio institucional y derechos de autoría.

Objetivos de Aprendizaje I: Entregar el documento de tesis completo cumpliendo todos los requisitos institucionales. [Evaluar] **II:** Presentar el borrador final ante el comité pre-sustentador e incorporar sus correcciones. [Evaluar] **III:** Completar los trámites administrativos para la programación de la sustentación pública. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library*. Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library*. IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI403 Taller de Investigación (Obligatorio)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 4
- **Horas del curso**: **Teoría**: 1 horas; **Práctica**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS402 Proyecto de tesis 1 (8vo Sem-Pág 239)

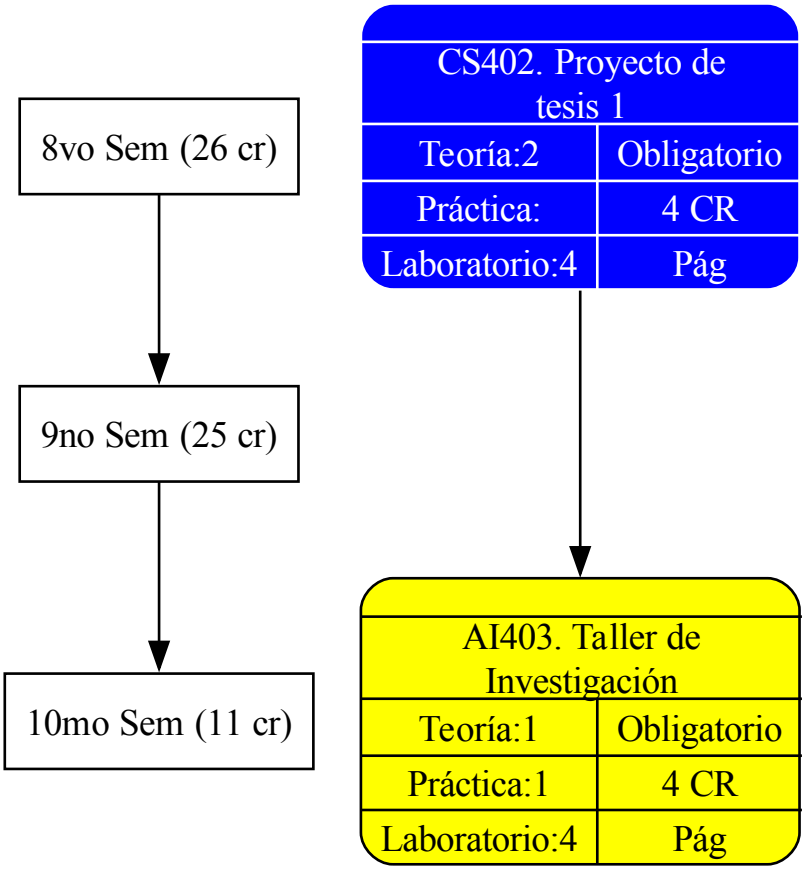


Figura 3.64: Mapa de Conexión. AI403 Taller de Investigación

2. Fundamentación

Este curso tiene por objetivo que el alumno desarrolle su propuesta de investigación y ejecute los primeros experimentos formales, produciendo los capítulos centrales de su tesis bajo la supervisión del asesor.

3. Objetivos del curso.

- Formalizar el plan de tesis en el formato institucional con objetivos, cronograma y análisis de viabilidad.

- Redactar el marco teórico y el capítulo de trabajos relacionados con síntesis crítica de la literatura.
- Diseñar la metodología original e implementar un prototipo funcional de la propuesta de investigación.
- Ejecutar experimentos preliminares comparativos con rigor estadístico e interpretación fundamentada de resultados.
- Presentar y defender el avance de tesis ante el comité evaluador incorporando su retroalimentación.
- Los entregables de este curso son:

  Avance parcial: Plan de tesis en formato institucional, capítulo de marco teórico y trabajos relacionados en LaTeX.

  Final: Capítulos de metodología, propuesta y experimentos preliminares aprobados por el asesor.

4. Competencias Generales (*Outcomes*)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Evaluar)

AG-C04) Comunicación: Se comunica de forma efectiva en actividades complejas de computación. (Evaluar)

AG-C05) Gestión de Proyectos: Aplica los principios de gestión en computación para gestionar proyectos. (Evaluar)

AG-C08) Análisis de Problemas: Identifica, formula y analiza problemas complejos de computación. (Evaluar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Evaluar)

5. Contenido

5.1 Formalización del Plan de Tesis, 10 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: **I:** Estructura del plan de tesis según el formato institucional. **II:** Definición precisa del problema de investigación y pregunta central de tesis. **III:** Formulación de objetivos generales y específicos verificables. **IV:** Cronograma de actividades con hitos hasta la defensa final de tesis. **V:** Análisis de viabilidad: recursos, tiempos, riesgos y contingencias del proyecto.

Objetivos de Aprendizaje **I:** Producir el plan de tesis en el formato exigido por la institución académica. [Evaluar] **II:** Definir un problema de investigación original y objetivos verificables. [Evaluar] **III:** Elaborar un cronograma detallado con hitos y análisis de riesgos hasta la defensa de tesis. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.2 Marco Teórico y Estado del Arte Avanzado, 10 hr(s)

Competencias: AG-C04,AG-C08,AG-C09

Tópicos: **I:** Organización temática y síntesis crítica de la literatura del área de tesis. **II:** Redacción del capítulo de marco teórico en LaTeXcon estructura coherente. **III:** Identificación y justificación de las brechas de investigación del problema. **IV:** Análisis comparativo de enfoques: tablas y diagramas de clasificación de la literatura. **V:** Integración de referencias con BibTeX y gestión de versiones con Git.

Objetivos de Aprendizaje I: Redactar el capítulo de marco teórico con síntesis estructurada y crítica de la literatura. [Evaluar] **II:** Justificar la originalidad del problema en relación con el estado del arte. [Evaluar] **III:** Construir tablas comparativas que evidencien las brechas de investigación. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008, CiteSeer.IST, 2008]

5.3 Diseño de la Metodología Original, 10 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Definición del enfoque metodológico: experimental, analítico o constructivo. **II:** Especificación de la contribución original: algoritmo, sistema, modelo o metodología. **III:** Diseño del protocolo experimental: variables, métricas, datasets y benchmarks. **IV:** Criterios de validación y generalización de resultados de la propuesta. **V:** Alineación de la metodología con los objetivos del plan de tesis.

Objetivos de Aprendizaje I: Diseñar la metodología de investigación alineada con los objetivos. [Evaluar] **II:** Especificar la contribución original del proyecto con criterios verificables. [Evaluar] **III:** Definir el protocolo experimental completo con variables, métricas y benchmarks. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.4 Implementación y Experimentos Preliminares, 12 hr(s)

Competencias: AG-C05,AG-C08,AG-C09

Tópicos: I: Implementación de la propuesta original: prototipo, algoritmo o modelo de tesis. **II:** Reproducción de baselines relevantes del estado del arte para comparación. **III:** Ejecución de experimentos preliminares con análisis estadístico básico. **IV:** Registro y documentación de configuraciones, versiones y resultados. **V:** Identificación de limitaciones y ajuste iterativo de la propuesta.

Objetivos de Aprendizaje I: Implementar un prototipo funcional de la propuesta original de tesis. [Usar] **II:** Ejecutar experimentos preliminares comparativos con baselines definidos. [Usar] **III:** Analizar e interpretar resultados con rigor estadístico apropiado. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.5 Redacción de Capítulos Centrales de la Tesis, 10 hr(s)

Competencias: AG-C02,AG-C04,AG-C09

Tópicos: I: Redacción del capítulo de metodología y diseño experimental en LaTeX. **II:** Redacción del capítulo de trabajos relacionados con análisis crítico comparativo. **III:** Presentación de resultados preliminares: tablas, figuras y análisis comparativo. **IV:** Ciclos de revisión con el asesor: incorporación de retroalimentación estructurada. **V:** Coherencia y consistencia narrativa entre capítulos de la tesis.

Objetivos de Aprendizaje I: Redactar los capítulos de metodología y trabajos relacionados con rigor académico. [Evaluar] **II:** Presentar resultados preliminares con visualizaciones claras y análisis comparativo. [Usar] **III:** Aplicar ciclos de revisión con el asesor para mejorar la calidad del documento. [Evaluar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

5.6 Presentación del Avance y Planificación Final, 8 hr(s)

Competencias: AG-C02,AG-C04,AG-C05

Tópicos: **I:** Presentación oral del avance de tesis ante el comité evaluador. **II:** Defensa de metodología y resultados preliminares ante preguntas del comité. **III:** Incorporación de retroalimentación del comité en la revisión del documento. **IV:** Planificación de actividades restantes hacia la defensa final de tesis. **V:** Gestión ética: autoría, integridad académica y confidencialidad de datos.

Objetivos de Aprendizaje **I:** Presentar y defender oralmente el avance de tesis ante un comité académico. [Evaluar] **II:** Incorporar la retroalimentación del comité en una revisión fundamentada del documento. [Evaluar] **III:** Planificar las actividades restantes con objetivos claros hacia la defensa final. [Usar]

Bibliografía: [IEEE-Computer Society, 2008, ACM, 2008]

6. Bibliografía

[ACM, 2008] ACM (2008). *Digital Library*. Association for Computing Machinery. http://portal.acm.org/dl.cfm.

[CiteSeer.IST, 2008] CiteSeer.IST (2008). *Scientific Literature Digital Library*. College of Information Sciences and Technology, Penn State University. http://citeseer.ist.psu.edu.

[IEEE-Computer Society, 2008] IEEE-Computer Society (2008). *Digital Library*. IEEE-Computer Society. http://www.computer.org/publications/dlib.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

AI369 Robótica (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - AI268 Visión Computacional (9no Sem-Pág 288)

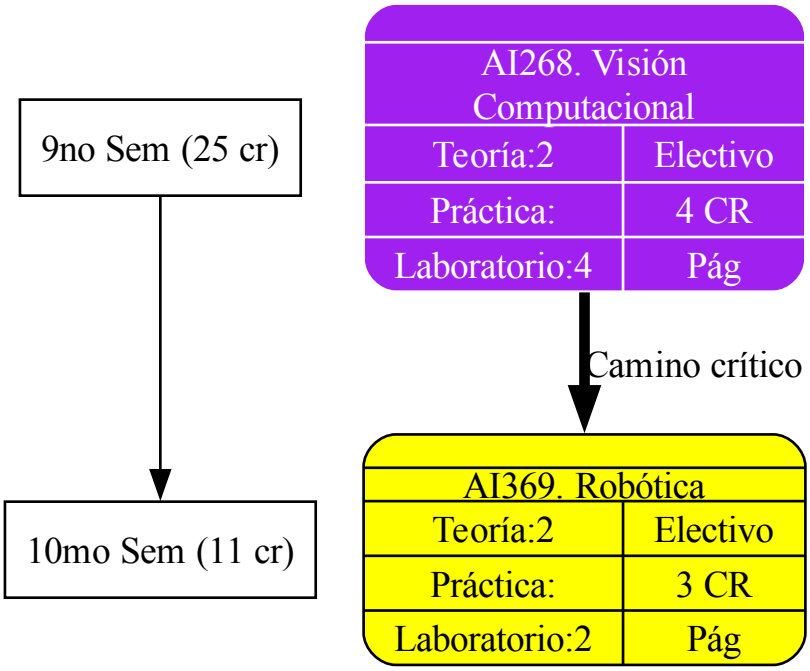


Figura 3.65: Mapa de Conexión. AI369 Robótica

2. Fundamentación

La Robótica es la manifestación física de la Inteligencia Artificial. Este curso introduce los conceptos fundamentales del diseño de robots, incluyendo cinemática, percepción y control. Los estudiantes explorarán cómo los robots interactúan con el mundo físico a través de sensores y actuadores, y cómo toman decisiones autónomas usando técnicas probabilísticas de mapeo y localización. El curso cierra la brecha entre algoritmos abstractos y ejecución física.

3. Objetivos del curso.

- Comprender los fundamentos mecánicos y matemáticos de la cinemática del robot.
- Dominar técnicas de fusión de sensores para percepción robótica.
- Implementar algoritmos de navegación y planificación de trayectorias.
- Analizar la integración del aprendizaje automático en el control robótico.
- Diseñar y simular agentes autónomos para entornos específicos.

4. Competencias Generales (*Outcomes*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Evaluar)

5. Contenido

5.1 Robótica, 12 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: **I:** Visión general: problemas y progreso: *a*) Sistemas robóticos de última generación, incluyendo sus sensores y una visión general de su procesamiento sensorial *b*) Arquitecturas de control de robots, por ejemplo, control deliberativo versus reactivo y vehículos de Braitenberg *c*) Modelado del mundo y modelos del mundo *d*) Incertidumbre inherente en la percepción y en el control **II:** Sensores y efectores: *a*) Sensores: por ejemplo, LIDAR, sonar, visión, profundidad, estereoscópico, cámaras de eventos, micrófonos, háptica *b*) Efectores: por ejemplo, ruedas, brazos, pinzas **III:** Marcos de coordenadas, traslación y rotación (2D y 3D) **IV:** Espacio de configuración y mapas ambientales **V:** Interpretación de datos sensoriales inciertos **VI:** Localización y mapeo **VII:** Navegación y control **VIII:** Cinemática directa e inversa **IX:** Planificación de trayectorias de movimiento y optimización de trayectorias **X:** Manipulación y agarre **XI:** Control y dinámica de articulaciones **XII:** Control basado en visión **XIII:** Coordinación y colaboración múltiple de robots **XIV:** Interacción humano-robot: *a*) Espacios de trabajo compartidos *b*) Equipos humano-robot e HRI físico *c*) Robots de asistencia social *d*) Predicción de movimiento/tarea/meta *e*) Colaboración y comunicación (explícita vs implícita, verbal o simbólica vs no verbal o visual) *f*) Confianza **XV:** Aplicaciones y Problemas Sociales, Económicos y éticos: *a*) Implicaciones sociales, económicas, de derecho al trabajo *b*) Implicaciones éticas y de privacidad de las aplicaciones robóticas *c*) Responsabilidad en robótica autónoma *d*) Armas autónomas y ética *e*) Supervisión y control humano

Objetivos de Aprendizaje **I:** 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] **II:** Integrar sensores, actuadores y software en un robot diseñado para emprender una tarea específica [Aplicar] **III:** Programar un robot para realizar tareas simples usando arquitecturas de control deliberativas, reactivas y/o híbridas [Aplicar] **IV:** Implementar algoritmos fundamentales de planificación de movimiento dentro de un espacio de configuración de robot [Aplicar] **V:** Caracterizar las incertidumbres asociadas con sensores y actuadores robóticos comunes; articular estrategias para mitigar estas incertidumbres [Caracterizar] **VI:** Listar las diferencias entre las representaciones del entorno externo de los robots, incluyendo sus fortalezas y limitaciones [Listar/Enumerar] **VII:** 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] **VIII:** Describir al menos un enfoque para coordinar las acciones y percepción de varios robots para realizar una única tarea [Describir] **IX:** 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] **X:** Analizar los problemas sociales, económicos y éticos de una aplicación robótica del mundo real [Analizar]

Bibliografía: [Siegwart et al., 2011, Craig, 2017]

5.2 Percepción y Visión por Computadora, 12 hr(s)

Competencias: AG-C09,AG-C11

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: [Thrun et al., 2005, Siegwart et al., 2011]

5.3 Robótica, 24 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Visión general: problemas y progreso: *a*) Sistemas robóticos de última generación, incluyendo sus sensores y una visión general de su procesamiento sensorial *b*) Arquitecturas de control de robots, por ejemplo, control deliberativo versus reactivo y vehículos de Braitenberg *c*) Modelado del mundo y modelos del mundo *d*) Incertidumbre inherente en la percepción y en el control **II:** Sensores y efectores: *a*) Sensores: por ejemplo, LIDAR, sonar, visión, profundidad, estereoscópico, cámaras de eventos, micrófonos, háptica *b*) Efectores: por ejemplo, ruedas, brazos, pinzas **III:** Marcos de coordenadas, traslación y rotación (2D y 3D) **IV:** Espacio de configuración y mapas ambientales **V:** Interpretación de datos sensoriales inciertos **VI:** Localización y mapeo **VII:** Navegación y control **VIII:** Cinemática directa e inversa **IX:** Planificación de trayectorias de movimiento y optimización de trayectorias **X:** Manipulación y agarre **XI:** Control y dinámica de articulaciones **XII:** Control basado en visión **XIII:** Coordinación y colaboración múltiple de robots **XIV:** Interacción humano-robot: *a*) Espacios de trabajo compartidos *b*) Equipos humano-robot e HRI físico *c*) Robots de asistencia social *d*) Predicción de movimiento/tarea/meta *e*) Colaboración y comunicación (explícita vs implícita, verbal o simbólica vs no verbal o visual) *f*) Confianza **XV:** Aplicaciones y Problemas Sociales, Económicos y éticos: *a*) Implicaciones sociales, económicas, de derecho al trabajo *b*) Implicaciones éticas y de privacidad de las aplicaciones robóticas *c*) Responsabilidad en robótica autónoma *d*) Armas autónomas y ética *e*) Supervisión y control humano

Objetivos de Aprendizaje I: 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] **II:** Integrar sensores, actuadores y software en un robot diseñado para emprender una tarea específica [Aplicar] **III:** Programar un robot para realizar

tareas simples usando arquitecturas de control deliberativas, reactivas y/o híbridas [Aplicar] **IV:** Implementar algoritmos fundamentales de planificación de movimiento dentro de un espacio de configuración de robot [Aplicar] **V:** Caracterizar las incertidumbres asociadas con sensores y actuadores robóticos comunes; articular estrategias para mitigar estas incertidumbres [Caracterizar] **VI:** Listar las diferencias entre las representaciones del entorno externo de los robots, incluyendo sus fortalezas y limitaciones [Listar/Enumerar] **VII:** 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] **VIII:** Describir al menos un enfoque para coordinar las acciones y percepción de varios robots para realizar una única tarea [Describir] **IX:** 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] **X:** Analizar los problemas sociales, económicos y éticos de una aplicación robótica del mundo real [Analizar]

Bibliografía: [Thrun et al., 2005, Siegwart et al., 2011]

6. Bibliografía

[Craig, 2017] Craig, J. J. (2017). *Introduction to Robotics: Mechanics and Control*. Pearson, 4th edition.

[Siegwart et al., 2011] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press, 2nd edition.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS281 Computación en la Sociedad (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 2
- **Horas del curso**: **Teoría**: 2 horas;
- **Prerrequisitos**: Ninguno

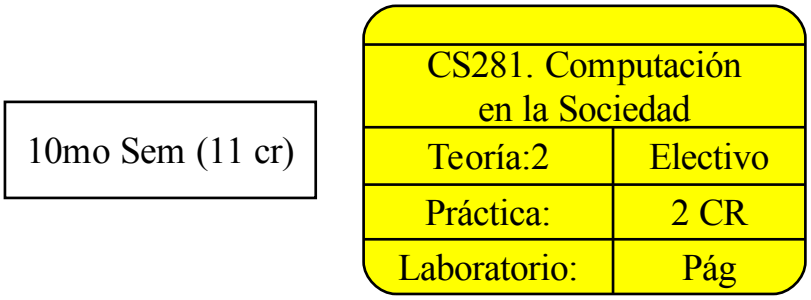


Figura 3.66: Mapa de Conexión. CS281 Computación en la Sociedad

2. Fundamentación

Este curso explora la relación multifacética entre las tecnologías de computación y la sociedad. Proporciona una perspectiva histórica sobre la evolución del campo mientras equipa a los estudiantes con marcos de análisis ético para navegar desafíos contemporáneos. Los temas incluyen propiedad intelectual, privacidad, responsabilidad profesional, sostenibilidad y el impacto socioeconómico de la automatización. Los estudiantes desarrollarán las habilidades de comunicación y sensibilidad ética requeridas para la práctica profesional en un entorno diverso y globalizado.

3. Objetivos del curso.

- Aplicar análisis ético a dilemas profesionales en computación.
- Comprender los marcos legales para Propiedad Intelectual y Privacidad.
- Evaluar la sostenibilidad ambiental y el impacto social de las TI.
- Comunicarse efectivamente respecto a problemas profesionales y éticos.

4. Competencias Generales (*Outcomes*)

AG-C01) El Profesional y el Mundo: Analiza y evalúa el impacto de las soluciones a problemas complejos de computación en el desarrollo sostenible de la sociedad. (Usar)

AG-C02) Ética: Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional de la computación. (Usar)

5. Contenido

5.1 Fundamentos e Historia Temprana de la Informática, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: La historia de la informática: hardware, software y humano/organizacional. **II:** El papel de la historia en el presente, incluyendo dentro de diferentes contextos sociales, y la relevancia de esta historia en el futuro. **III:** 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). **IV:** Edad II (Computación moderna temprana): ENIAC, UNIVAC, Bombas (Bletchley Park y descifradores de códigos), compañías informáticas (por ejemplo, IBM), mainframes, etc.

Objetivos de Aprendizaje I: 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)] **II:** 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] **III:** Identificar tendencias formativas y consecuentes en la historia del campo de la informática [Analizar] **IV:** Identificar las contribuciones de individuos u organizaciones pioneras (laboratorios de investigación, compañías informáticas, oficinas gubernamentales) en el campo de la informática [Analizar]

Bibliografía: [Quinn, 2019, Baase and Henry, 2017]

5.2 Impacto Social e IA, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: 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. **II:** 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). **III:** 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. **IV:** Crecimiento y control de internet, datos, informática e inteligencia artificial.

Objetivos de Aprendizaje I: Describir las diferentes formas en que la tecnología informática (redes, computación móvil, inteligencia artificial) media la interacción social a nivel personal y colectivo [Describir] **II:** Interpretar el contexto social de un diseño dado y su implementación [Interpretar] **III:** Analizar la eficacia de un diseño e implementación dados utilizando datos empíricos [Analizar] **IV:** Comprender las implicaciones del uso de la tecnología (por ejemplo, redes sociales) para diferentes identidades, culturas y comunidades [Explicar] **V:** Describir el papel de internet en facilitar la comunicación entre ciudadanos, gobiernos y entre sí [Describir] **VI:** Analizar los efectos de la dependencia de la informática en la implementación de la democracia (por ejemplo, entrega de servicios sociales, votación electrónica) [Analizar] **VII:** 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]

Bibliografía: [Quinn, 2019]

5.3 Fundamentos y Marcos del Análisis Ético, 4 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Evitar falacias y tergiversaciones en la argumentación. **II:** Teorías éticas y toma de decisiones (marcos filosóficos y sociales, por ejemplo [1]). **III:** Reconocimiento del papel que juega la cultura en nuestra comprensión, adopción, diseño y uso de la tecnología informática.

IV: Por qué la ética es importante en la informática y cómo la ética es similar y diferente de las leyes y normas sociales. **V:** 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.

Objetivos de Aprendizaje **I:** Describir cómo un contexto cultural dado impacta la toma de decisiones [Describir] **II:** Expresar el uso de ejemplos y analogías en la argumentación ética [Analizar] **III:** Analizar (y evitar) falacias lógicas básicas en un argumento [Analizar] **IV:** Analizar un argumento para identificar premisas y conclusión [Analizar] **V:** 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] **VI:** Justificar una decisión tomada sobre bases éticas [Justificar] **VII:** Discutir las ventajas y desventajas de los marcos éticos tradicionales [Debatir] **VIII:** Analizar dilemas éticos relacionados con la creación y uso de tecnología desde múltiples perspectivas utilizando marcos éticos [Analizar]

Bibliografía: [Quinn, 2019, Baase and Henry, 2017]

5.4 Valores, Naturaleza y Actualización Profesional, 4 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: **I:** Valores de la comunidad y las leyes por las que vivimos. **II:** La naturaleza de ser un profesional, incluyendo cuidado, atención, disciplina, responsabilidad fiduciaria y tutoría (mentoring). **III:** 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.

Objetivos de Aprendizaje **I:** Identificar problemas éticos que surgen en el diseño de software, prácticas de desarrollo y despliegue de software [Analizar] **II:** Discutir cómo abordar problemas éticos en situaciones específicas [Debatir] **III:** Describir los mecanismos que típicamente existen para que un profesional se mantenga actualizado en asuntos éticos [Describir]

Bibliografía: [Baase and Henry, 2017]

5.5 Fundamentos de la Propiedad Intelectual, 4 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: **I:** Derechos de propiedad intelectual. **II:** Fundamentos legales para la protección de la propiedad intelectual. **III:** Fundamentos filosóficos de la propiedad intelectual. **IV:** Formas de propiedad intelectual (por ejemplo, derechos de autor, patentes, secretos comerciales, marcas registradas) y los derechos que protegen.

Objetivos de Aprendizaje **I:** Defender los usos legales y éticos de materiales con derechos de autor [Defender] **II:** Discutir las bases filosóficas de la propiedad intelectual en un contexto apropiado (por ejemplo, país) [Debatir] **III:** Contrastar las protecciones y obligaciones de los derechos de autor, patentes, secretos comerciales y marcas registradas [Contrastar] **IV:** Describir la justificación para la protección legal de la propiedad intelectual en el contexto apropiado (por ejemplo, país) [Describir]

Bibliografía: [Quinn, 2019]

5.6 Derechos de PI Digital y Derechos de Autor, 6 hr(s)

Competencias:

Tópicos: **I:** Propiedad intelectual digital intangible (IDIP). **II:** Limitaciones a las protecciones de derechos de autor, incluyendo uso justo (fair use) y la doctrina de la primera venta (first sale doctrine). **III:** Leyes y tratados de propiedad intelectual que impactan la aplicación de los derechos de autor.

Objetivos de Aprendizaje I: Describir y criticar la legislación y los precedentes destinados a las infracciones de derechos de autor digitales [Describir] **II:** Identificar ejemplos contemporáneos de propiedad intelectual digital intangible [Analizar] **III:** Discutir si un uso de material con derechos de autor es probable que sea uso justo (fair use) [Debatir] **IV:** 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] **V:** Discutir la naturaleza de las leyes anti-elusión (anti-circumvention) en el contexto de la protección de derechos de autor [Debatir]

Bibliografía: [Tavani, 2013]

5.7 Datos, Vigilancia y Tecnología de Privacidad, 4 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: 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. **II:** Concepciones de anonimato, seudonimidad e identidad. **III:** Soluciones basadas en tecnología para la protección de la privacidad (por ejemplo, encriptación de extremo a extremo y privacidad diferencial).

Objetivos de Aprendizaje I: Evaluar soluciones a las amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Evaluar] **II:** 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] **III:** Distinguir los conceptos y objetivos del anonimato y la seudonimidad [Distinguir] **IV:** Describir las ramificaciones de las protecciones de privacidad basadas en tecnología, incluyendo la privacidad diferencial y la encriptación de extremo a extremo [Describir]

Bibliografía: [Baase and Henry, 2017]

5.8 Huella Ambiental de los Sistemas Informáticos, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Impactos ambientales, sociales y culturales de las decisiones de implementación (por ejemplo, objetivos de sostenibilidad, sesgo/resultados algorítmicos, viabilidad económica y consumo de recursos). **II:** 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).

Objetivos de Aprendizaje I: 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)] **II:** Describir los impactos sociales y ambientales globales del uso y disposición de computadoras [Describir] **III:** 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] **IV:** 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)]

Bibliografía: [Quinn, 2019]

5.9 Modelos y Estrategias Económicas, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: 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). **II:** Estrategias de precios y despliegue: obsolescencia programada (planned obsolescence), suscripciones, freemium, licencias de software, código abierto (open-source), software libre (free software), adware. **III:** 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.

Objetivos de Aprendizaje I: Resumir los efectos sociales de los modelos económicos (por ejemplo, las economías del conocimiento y de la atención) [Resumir] **II:** Describir las diferencias y similitudes de estrategias competitivas (por ejemplo, suscripción vs freemium vs gratis) [Describir] **III:** Discutir ejemplos de brechas digitales [Debatir]

Bibliografía: [Baase and Henry, 2017]

5.10 Delitos y Ataques Informáticos, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Delitos informáticos, recurso legal para criminales informáticos e impacto en víctimas y sociedad. **II:** Ingeniería social, fraude habilitado por la informática, robo de identidad y recuperación de estos. **III:** Terrorismo cibernético, hacking criminal y hacktivismo. **IV:** Malware, virus, gusanos. **V:** Ataques a infraestructura crítica como redes eléctricas y oleoductos.

Objetivos de Aprendizaje I: Enumerar ejemplos clásicos de delitos informáticos e incidentes de ingeniería social con impacto social [Listar/Enumerar] **II:** Identificar problemas con las leyes que se aplican a los delitos informáticos [Analizar] **III:** Describir la motivación y ramificaciones del terrorismo cibernético, robo de datos, hacktivismo, ransomware y otros ataques [Describir] **IV:** Examinar los problemas éticos y legales que rodean el mal uso del acceso y varias violaciones de seguridad [Examinar]

Bibliografía: [Baase and Henry, 2017]

5.11 Identidad y Diversidad en la Informática, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Cómo la identidad impacta y es impactada por las tecnologías y entornos informáticos (académicos y profesionales). **II:** Los beneficios de los equipos de desarrollo diversos y los impactos de los equipos que no son diversos.

Objetivos de Aprendizaje I: Definir y distinguir equidad, igualdad, diversidad, inclusión y accesibilidad [Definir] **II:** 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] **III:** 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)]

Bibliografía: [Quinn, 2019]

5.12 Comunicación Técnica y Profesional, 2 hr(s)

Competencias: AG-C01,AG-C02

Tópicos: I: Comunicación oral, escrita y electrónica en equipo y grupo. **II:** Materiales de comunicación técnica (por ejemplo, código fuente y documentación, tutoriales, materiales de referencia, documentación de API). **III:** Comunicarse con diferentes partes interesadas, como clientes, liderazgo o el público. **IV:** Compensaciones (tradeoffs) en factores competitivos que afectan los canales y elecciones de comunicación.

Objetivos de Aprendizaje I: 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] **II:** Analizar documentación técnica escrita para verificar precisión técnica, concisión, falta de ambigüedad y conciencia de la audiencia [Analizar] **III:** Componer y entregar una presentación formal consciente de la audiencia, accesible y organizada [Componer] **IV:** Identificar y describir cualidades de la comunicación efectiva (por ejemplo, virtual, cara a cara, intragrupo, documentos compartidos) [Analizar] **V:** Elegir una forma apropiada de comunicar preocupaciones éticas delicadas [Evaluar (valorar)]

Bibliografía: [Quinn, 2019]

6. Bibliografía

[Baase and Henry, 2017] Baase, S. and Henry, T. (2017). *A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology*. Pearson, 5th edition.

[Quinn, 2019] Quinn, M. J. (2019). *Ethics for the Information Age*. Pearson, 8th edition.

[Tavani, 2013] Tavani, H. T. (2013). *Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing*. Wiley, Hoboken, NJ, 4th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS351 Tópicos en Computación Gráfica (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS251 Computación Gráfica (7mo Sem-Pág 225)

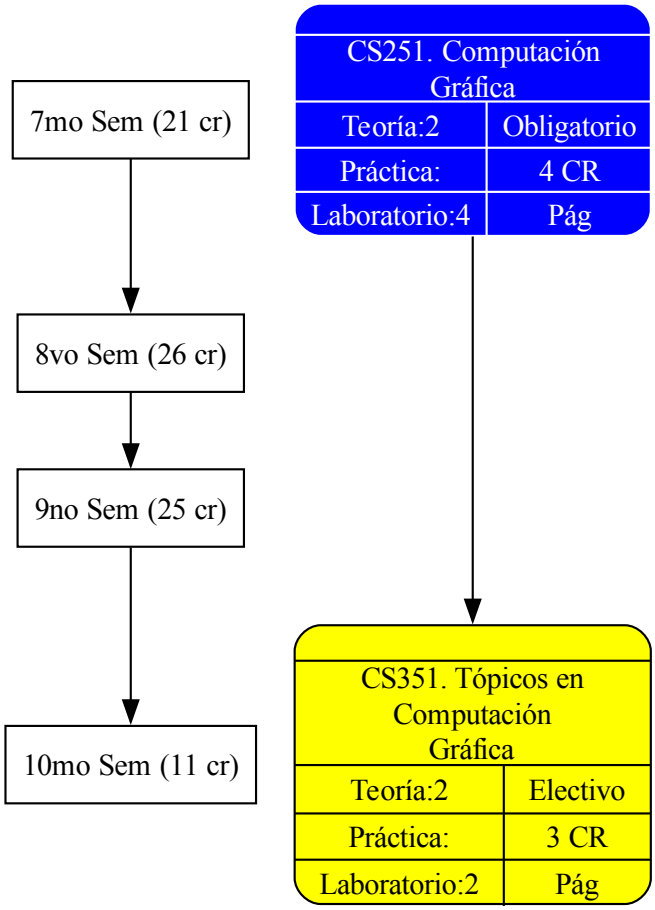


Figura 3.67: Mapa de Conexión. CS351 Tópicos en Computación Gráfica

2. Fundamentación

Los gráficos por computadora modernos van más allá de la simple generación de imágenes, abarcando interacción en tiempo real, simulación física y entornos inmersivos. Este curso explora temas avanzados en la canalización de gráficos, enfocándose en sombreado programable, modelado geométrico complejo y animación por computadora. Además, aborda la intersección de los gráficos con la computación física y las implicaciones éticas de la generación de contenido sintético.

3. Objetivos del curso.

- Dominar la programación avanzada de sombreadores y el renderizado no fotorrealista.
- Implementar simulaciones basadas en física y animaciones complejas.
- Diseñar experiencias inmersivas e interactivas usando computación tangible.
- Evaluar el impacto ético y social de las imágenes generadas por computadora.

4. Competencias Generales (*Outcomes*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Sombreado y Renderizado Avanzado, 12 hr(s)

Competencias: AG-C09,AG-C11

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, 2021a, Marschner and Shirley, 2021b]

5.2 Modelado Geométrico, 10 hr(s)

Competencias: AG-C09,AG-C11

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, 2021a]

5.3 Animación por Computadora, 8 hr(s)

Competencias: AG-C09,AG-C11

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

Objetivos de Aprendizaje I: Aplicar principios de animación para crear movimiento creíble [Aplicar] **II:** Implementar animación por fotogramas clave con técnicas de interpolación [Implementar] **III:** Crear animaciones procedurales como sistemas de partículas o simulaciones basadas en física [Crear] **IV:** Implementar animación de personajes usando sistemas esqueléticos y skinning [Implementar] **V:** Procesar datos de captura de movimiento para retargeting de animación [Implementar]

Bibliografía: [Parent, 2012]

5.4 Simulación, 8 hr(s)

Competencias: AG-C09,AG-C11

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

Objetivos de Aprendizaje I: Implementar simulaciones basadas en física para cuerpos rígidos o fluidos [Implementar] **II:** Aplicar métodos de integración numérica para simulaciones dinámicas [Aplicar] **III:** Optimizar simulaciones para rendimiento en tiempo real [Crear] **IV:** Desarrollar simulaciones para aplicaciones específicas como videojuegos o visualización científica [Crear]

Bibliografía: [Parent, 2012]

5.5 Inmersión, 5 hr(s)

Competencias: AG-C09,AG-C11

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

Objetivos de Aprendizaje I: Explicar los conceptos centrales y diferencias entre VR y AR [Explicar] **II:** Evaluar tecnologías de visualización inmersiva y sus aplicaciones [Evaluar] **III:** Implementar soluciones de seguimiento y entrada para entornos inmersivos [Implementar] **IV:** Diseñar experiencias inmersivas que maximicen la presencia y minimicen la incomodidad [Diseñar]

Bibliografía: [LaValle, 2023]

5.6 Interacción, 5 hr(s)

Competencias: AG-C09,AG-C11

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

Objetivos de Aprendizaje I: Comparar diferentes paradigmas de interacción y su idoneidad para varias tareas [Comparar] **II:** Diseñar interfaces de usuario para aplicaciones intensivas en gráficos [Diseñar] **III:** Implementar sistemas de interacción multimodal incorporando múltiples canales de retroalimentación [Implementar] **IV:** Realizar evaluaciones de usabilidad de técnicas de interacción [Evaluar]

Bibliografía: [LaValle, 2023]

5.7 Computación Tangible/Física, 8 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: **I:** Interfaces tangibles y computación física **II:** Sensores y actuadores para sistemas interactivos **III:** Sistemas embebidos para arte y diseño interactivo **IV:** Instalaciones y entornos interactivos

Objetivos de Aprendizaje **I:** Diseñar interfaces tangibles que unan interacciones físicas y digitales [Diseñar] **II:** Integrar sensores y actuadores para crear sistemas responsivos [Integrar] **III:** Desarrollar sistemas embebidos para proyectos de arte o diseño interactivo [Crear] **IV:** Crear instalaciones interactivas que involucren a los usuarios en espacios físicos [Crear]

Bibliografía: [Noble, 2012]

5.8 Sociedad, ética y Profesión, 8 hr(s)

Competencias: AG-C09,AG-C11

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

Objetivos de Aprendizaje **I:** Analizar problemas éticos relacionados con representación, accesibilidad y privacidad en gráficos e interacción [Analizar] **II:** Discutir las implicaciones sociales de las tecnologías inmersivas y la visualización de datos [Debatir] **III:** Describir prácticas y estándares profesionales en industrias de gráficos e interactivas [Describir]

Bibliografía: [Marschner and Shirley, 2021a]

6. Bibliografía

[LaValle, 2023] LaValle, S. M. (2023). *Virtual Reality*. Cambridge University Press.

[Marschner and Shirley, 2021a] Marschner, S. and Shirley, P. (2021a). *Fundamentals of Computer Graphics*. CRC Press, 5th edition.

[Marschner and Shirley, 2021b] Marschner, S. and Shirley, P. (2021b). *Fundamentals of Computer Graphics*. CRC Press, 5th edition. Same book as Marschner2021; alternate citation key using lead author Shirley.

[Noble, 2012] Noble, J. (2012). *Programming Interactivity: A Designer's Guide to Processing, Arduino, and OpenFrameworks*. O'Reilly Media, 2nd edition.

[Parent, 2012] Parent, R. (2012). *Computer Animation: Algorithms and Techniques*. Morgan Kaufmann, 3rd edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS353 Computación Cuántica (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS3I1 Seguridad en Computación (9no Sem-Pág 258)
  - MA307 Matemática aplicada a la computación (6to Sem-Pág 214)

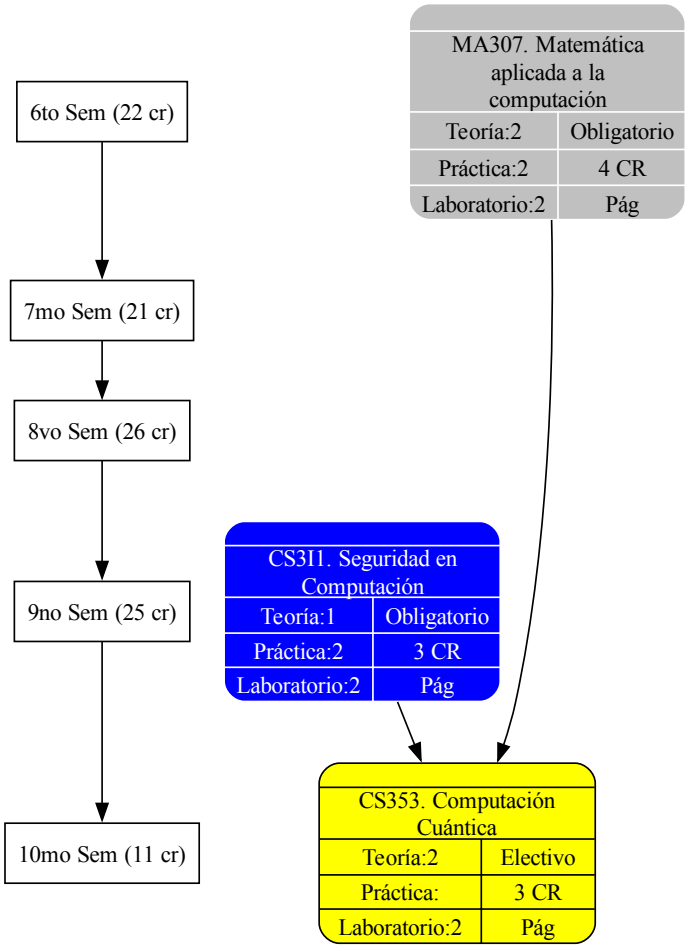


Figura 3.68: Mapa de Conexión. CS353 Computación Cuántica

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*)

AG-C10) Indagación: Estudia problemas complejos de computación usando métodos de ciencias de la información. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Conceptos fundamentales, 10 hr(s)

Competencias: AG-C10,AG-C11

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: AG-C10,AG-C11

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: AG-C10,AG-C11

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: AG-C10,AG-C11

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: AG-C10,AG-C11

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. Accessed: 2026-02-13. The original page has been moved or updated.

[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. Ph219/CS219 course, California Institute of Technology. Material available online. The original URL `http://theory.caltech.edu/~preskill/ph219/` is no longer active (February 2026). Check the author's website for the current version.

[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 de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS392 Tópicos en Ingeniería de Software (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS291 Ingeniería de Software I (5to Sem-Pág 178)

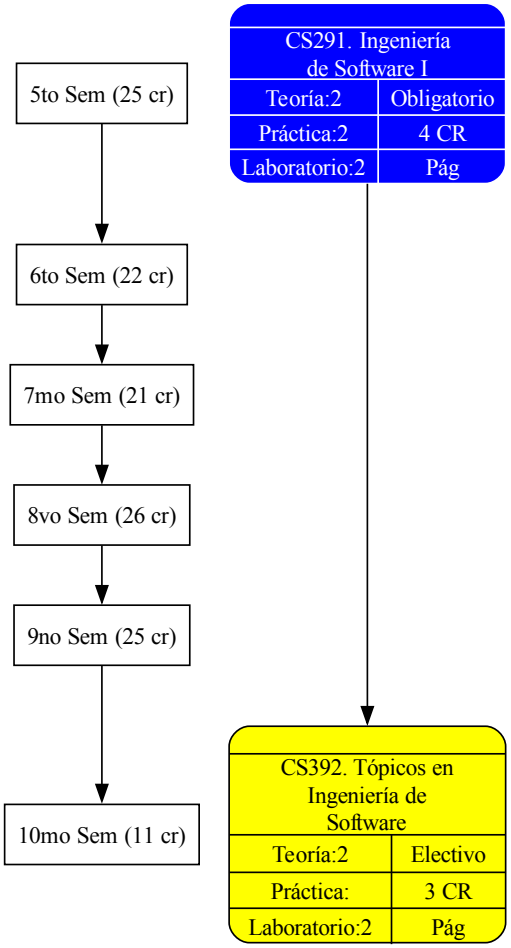


Figura 3.69: Mapa de Conexión. CS392 Tópicos en Ingeniería de Software

2. Fundamentación

El desarrollo de software moderno requiere prácticas de ingeniería rigurosas para garantizar la calidad, mantenibilidad y confiabilidad. Este curso explora temas avanzados en el ciclo de vida del software, enfocándose en la colaboración en equipo, verificación formal y refactorización avanzada. Los estudiantes dominarán el uso de herramientas profesionales para diseñar, construir y validar sistemas complejos mientras aseguran altos estándares de confiabilidad y corrección formal.

3. Objetivos del curso.

- Aplicar técnicas avanzadas de construcción y refactorización de software para mejorar la calidad del código.
- Usar métodos formales y estrategias de validación para garantizar la confiabilidad del sistema.
- Gestionar requisitos complejos y diseñar arquitecturas que cumplan con las necesidades de las partes interesadas.
- Liderar equipos de software usando herramientas profesionales y flujos de trabajo colaborativos.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Colaboración y Dinámica de Equipo, 4 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** 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 **II:** Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos. **III:** Programación cooperativa: *a*) Programación por parejas (pair programming) o Enjambre (swarming) *b*) Revisión de código *c*) Colaboración mediante control de versiones **IV:** 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 *a*) Ventajas del trabajo en equipo *b*) Riesgos y complejidad de dicha colaboración **V:** Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo

Objetivos de Aprendizaje **I:** Seguir prácticas efectivas de comunicación en equipo [Analizar] **II:** 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] **III:** Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar] **IV:** Colaborar efectivamente en desarrollo/programación cooperativa [Analizar] **V:** Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer] **VI:** Redactar y seguir una agenda para una reunión de equipo [Componer] **VII:** 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]

Bibliografía: [Sommerville, 2015]

5.2 Control de Versiones y CI/CD, 4 hr(s)

Competencias: AG-C03,AG-C09

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:** 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 **IV:** 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. **V:** Flujo de revisión colaborativa como puerta de calidad *a*) Revisión obligatoria de *pull requests* antes de fusionar *b*) Propiedad del código mediante reglas de tipo CODEOWNERS *c*) Convenciones de mensajes de *commit* (*conventional commits*) *d*) Reglas de protección de ramas que exigen verificaciones antes de fusionar **VI:** Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI

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:** Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] **V:** 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] **VI:** 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] **VII:** Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]

Bibliografía: [Pressman and Maxim, 2014]

5.3 Ingeniería de Requisitos, 6 hr(s)

Competencias: AG-C03,AG-C09

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:** Propiedades de los requisitos, incluyendo consistencia, validez, completitud y viabilidad. **III:** 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. **IV:** 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 **V:** 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 **VI:** Comunicar y/o formalizar especificaciones de requisitos.

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]

Bibliografía: [Sommerville, 2015]

5.4 Principios y Arquitectura de Software, 6 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Principios de diseño de sistemas. Confiabilidad del Sistema *a*) Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) *b*) Separación de preocupaciones (separation of concerns) *c*) Ocultación de información (information hiding) *d*) Acoplamiento y cohesión (coupling and cohesion) **II:** Arquitectura de software. Confiabilidad del Sistema *a*) Paradigmas de diseño 1) Descomposición funcional de arriba hacia abajo/diseño por capas 2) Arquitectura orientada a datos 3) Análisis y diseño orientado a objetos 4) Diseño dirigido por eventos *b*) Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) *c*) Identificar límites y dependencias de componentes **III:** Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema **IV:** 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

Objetivos de Aprendizaje **I:** Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] **II:** 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] **III:** Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] **IV:** Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]

Bibliografía: [Sommerville, 2015]

5.5 Prácticas de Codificación, 6 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Pruebas prácticas a pequeña escala Prácticas de Desarrollo de Software *a*) Pruebas unitarias *b*) 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. **II:** Documentación Prácticas de Desarrollo de Software *a*) Documentación de la interfaz: describir los requisitos de la interfaz, potencialmente incluyendo contratos (formales o informales), pre y post condiciones, invariantes. *b*) 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.) 1) Aclarar el flujo de datos, cómputo, etc., centrándose en lo que el código es. 2) Identificar partes sutiles/complicadas del código y refactorizar para que sean autoexplicativas si es posible o proporcionar comentarios apropiados para aclarar. **III:** Estilo de codificación Prácticas de Desarrollo de Software *a*) Guías de estilo *b*) Comentarios *c*) Nomenclatura (*naming*) **IV:** "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 *a*) Prácticas de codificación defensiva *b*) Prácticas y principios de codificación

segura *c*) Usar mecanismos de manejo de excepciones para hacer los programas más robustos, tolerantes a fallos **V:** Depuración (debugging) Prácticas de Desarrollo de Software **VI:** Registro (logging) **VII:** Uso de bibliotecas y frameworks desarrollados por otros Prácticas de Desarrollo de Software

Objetivos de Aprendizaje I: Escribir pruebas unitarias apropiadas para un componente pequeño (varias funciones, un solo tipo, etc.) [Escribir] **II:** Escribir comentarios apropiados de interfaz y (si es necesario) de implementación para un componente pequeño [Escribir] **III:** Describir técnicas, patrones de codificación y mecanismos para implementar diseños para lograr propiedades deseadas como confiabilidad, eficiencia y robustez [Describir] **IV:** Escribir código robusto utilizando mecanismos de manejo de excepciones [Escribir] **V:** Describir prácticas de codificación segura y defensiva [Describir] **VI:** Seleccionar y usar un estándar de codificación definido en un proyecto de software pequeño [Usar] **VII:** Comparar y contrastar estrategias de integración, incluyendo integración de arriba hacia abajo (top-down), de abajo hacia arriba (bottom-up) y sándwich [Comparar] **VIII:** Describir el proceso de analizar e implementar cambios en la base de código desarrollada para un proyecto específico [Describir] **IX:** Describir el proceso de analizar e implementar cambios en una base de código existente grande [Describir]

Bibliografía: [Fowler, 2018]

5.6 Compatibilidad y Versionado de API, 6 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Ley de Hyrum / Ley de las Interfaces Implícitas **II:** Compatibilidad con versiones anteriores *a*) La compatibilidad no es una propiedad de una sola entidad, es una propiedad de una relación. *b*) 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. **III:** Control de versiones (versioning) *a*) Versionado Semántico (SemVer) *b*) Desarrollo basado en tronco (*trunk-based development*)

Objetivos de Aprendizaje I: Identificar tanto el comportamiento explícito como el implícito de una interfaz e identificar riesgos potenciales de la Ley de Hyrum [Analizar] **II:** Identificar cambios que pueden considerarse ampliamente ”compatibles con versiones anteriores”, potencialmente con declaraciones explícitas sobre qué uso está o no soportado [Analizar] **III:** Evaluar si un cambio propuesto es lo suficientemente seguro dada la metodología de control de versiones en uso para un proyecto dado [Evaluar]

Bibliografía: [Fowler, 2018]

5.7 Conceptos de verificación y validación, 6 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Conceptos de verificación y validación *a*) Verificación: ¿Estamos construyendo la cosa correctamente? *b*) Validación: ¿Construimos la cosa correcta? **II:** Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? **III:** Objetivos de las pruebas *a*) Usabilidad *b*) Confiabilidad *c*) Conformidad con la especificación *d*) Rendimiento *e*) Seguridad

Objetivos de Aprendizaje I: Explicar por qué las pruebas son importantes [Explicar] **II:** Distinguir entre validación y verificación de programas [Distinguir] **III:** Describir diferentes objetivos de las pruebas [Describir]

Bibliografía: [Pressman and Maxim, 2014]

5.8 Herramientas y Análisis de Pruebas, 3 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Herramientas y automatización de verificación y validación *a*) Análisis estático *b*) Cobertura de código *c*) Fuzzing *d*) Análisis dinámico y contención de fallos (sanitizadores, etc.) *e*) Registro y seguimiento de fallos **II:** Planificación y generación de pruebas *a*) Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos *b*) Uso de números aleatorios y pseudoaleatorios en las pruebas **III:** Pruebas de sistemas asíncronos, paralelos y concurrentes **IV:** Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) **V:** Desarrollo guiado por comportamiento (BDD) *a*) Expresar criterios de aceptación como escenarios Gherkin Given/When/Then *b*) Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo **VI:** Pruebas de mutación y análisis estático de calidad de código *a*) Pruebas de mutación para evaluar la efectividad de una suite de pruebas *b*) Detección estática de *code smells* y complejidad ciclomática excesiva

Objetivos de Aprendizaje **I:** Describir y comparar diferentes herramientas para verificación y validación [Describir] **II:** Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] **III:** Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] **IV:** Describir enfoques para la estimación de fallos [Describir] **V:** Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] **VI:** Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] **VII:** 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] **VIII:** Describir técnicas para la verificación y validación de artefactos que no son código [Describir] **IX:** Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] **X:** Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]

Bibliografía: [Pressman and Maxim, 2014]

5.9 Pruebas de Rendimiento y Evaluación Comparativa, 3 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Pruebas de rendimiento y evaluación comparativa *a*) Rendimiento (*throughput*) y latencia *b*) Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) *c*) Aceleración (*speedup*) y escalado (*scaling*) 1) Ley de Amdahl 2) Ley de Gustafson 3) Escalado suave (*soft*) y débil (*weak*) *d*) Identificar y medir figuras de mérito (*figures of merit*) *e*) Cuellos de botella de rendimiento comunes 1) Limitado por cómputo (*compute-bound*) 2) Limitado por ancho de banda de memoria 3) Limitado por latencia *f*) Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) 1) Estimación de la incertidumbre 2) Intervalos de confianza *g*) Análisis y presentación (gráficos, etc.) *h*) Técnicas de medición de tiempo (*timing*)

Objetivos de Aprendizaje **I:** Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] **II:** Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] **III:** Describir cuellos de botella de rendimiento comunes [Describir] **IV:** Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] **V:** Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] **VI:** 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]

Bibliografía: [Bondi, 2015, Gregg, 2020]

5.10 Conceptos de Confiabilidad, 5 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Concepto de confiabilidad como probabilidad de fallo o tiempo medio entre fallos, y fallas (faults) como causa de fallos (failures) **II:** Identificar requisitos de confiabilidad para diferentes tipos de software **III:** 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.) **IV:** Confiabilidad del software, confiabilidad del sistema y comportamiento de fallos **V:** Ciclo de inyección y eliminación de defectos, y diferentes enfoques para la eliminación de defectos **VI:** 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.

Objetivos de Aprendizaje **I:** Describir cómo determinar el nivel de confiabilidad requerido por un sistema de software [Describir] **II:** Explicar los problemas que existen para lograr niveles muy altos de confiabilidad [Explicar] **III:** Comprender enfoques para minimizar fallas que pueden aplicarse en cada etapa del ciclo de vida del software [Explicar]

Bibliografía: [Pressman and Maxim, 2014]

5.11 Métodos Formales, 5 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Especificación formal de interfaces *a*) Especificación de pre y post condiciones *b*) Lenguajes formales para escribir y analizar pre y post condiciones. **II:** áreas de problemas bien atendidas por los métodos formales *a*) Programación libre de bloqueos (*lock-free*), condiciones de carrera de datos *b*) Sistemas asíncronos y distribuidos, bloqueos (*deadlock*), bloqueos vivos (*livelock*), etc. **III:** Comparación con otras herramientas y técnicas para la detección de defectos *a*) Pruebas *b*) *Fuzzing* **IV:** Enfoques formales para el modelado y análisis de software *a*) Verificadores de modelos (model checkers) *b*) Buscadores de modelos (model finders)

Objetivos de Aprendizaje **I:** 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] **II:** Aplicar técnicas de especificación y análisis formal a diseños y programas de software de baja complejidad [Aplicar] **III:** Explicar los beneficios y desventajas potenciales del uso de lenguajes de especificación formal [Explicar]

Bibliografía: [Sommerville, 2015]

6. Bibliografía

[Bondi, 2015] Bondi, A. B. (2015). *Foundations of Software and System Performance Engineering: Process, Performance Modeling, Requirements, Testing, Scalability, and Practice*. Addison-Wesley, Upper Saddle River, NJ.

[Fowler, 2018] Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 2nd edition.

[Gregg, 2020] Gregg, B. (2020). *Systems Performance: Enterprise and the Cloud*. Addison-Wesley Professional, Boston, MA, 2nd edition.

[Pressman and Maxim, 2014] Pressman, R. S. and Maxim, B. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 8th edition.

[Sommerville, 2015] Sommerville, I. (2015). *Software Engineering*. Pearson, 10th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS393 Sistemas de Infomación (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 2 horas; **Laboratorio**: 2 horas;
- **Prerrequisitos**:
  - CS291 Ingeniería de Software I (5to Sem-Pág 178)

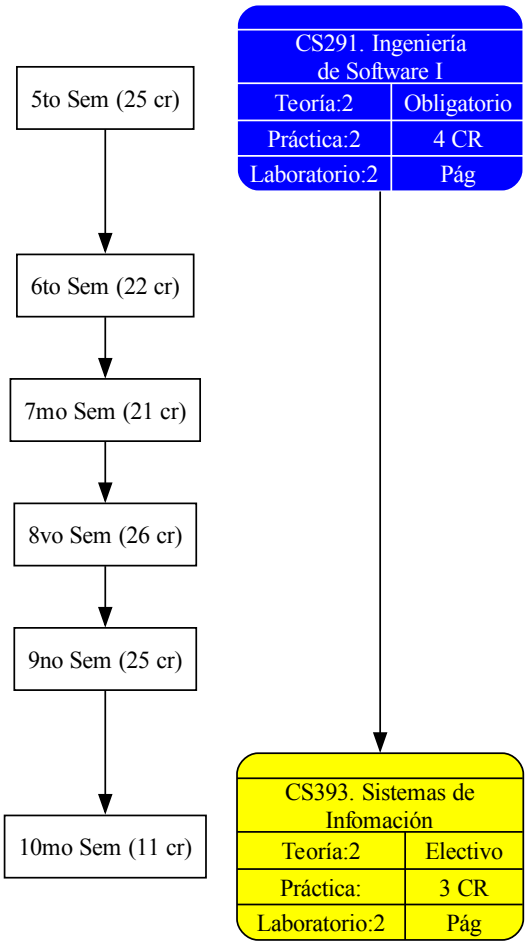


Figura 3.70: Mapa de Conexión. CS393 Sistemas de Infomación

2. Fundamentación

Este curso proporciona una visión general completa de cómo los sistemas de información respaldan los procesos empresariales y la toma de decisiones dentro de las organizaciones. Cubre el rol estratégico de la tecnología de la información, el ciclo de vida de los sistemas de información, estrategias de gestión de datos y la integración de aplicaciones empresariales para lograr ventajas competitivas en una economía digital globalizada.

3. Objetivos del curso.

- Comprender los componentes fundamentales y los tipos de Sistemas de Información.
- Analizar el papel de los SI en la estrategia organizacional y la toma de decisiones.
- Evaluar tecnologías emergentes y su impacto en los modelos de negocio.

4. Competencias Generales (*Outcomes*)

AG-C03) Trabajo Individual y en Equipo: Se desempeña efectivamente como individuo y como miembro o líder en equipos diversos. (Usar)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

5. Contenido

5.1 Foundations of Information Systems, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Componentes de los Sistemas de Información: Hardware, Software, Datos, Redes. **II:** Tipos de SI: TPS, MIS, DSS y ESS. **III:** Ventaja competitiva a través de la Tecnología de la Información. **IV:** Procesos empresariales e integración de Sistemas de Información.

Objetivos de Aprendizaje **I:** Identificar los componentes centrales de un sistema de información [Familiarizarse]. **II:** Explicar cómo los SI respaldan los diversos niveles de la jerarquía organizacional [Usar]. **III:** Analizar un caso de negocio para identificar usos estratégicos de TI [Evaluar].

Bibliografía: [Laudon and Laudon, 2021, Stair and Reynolds, 2017]

5.2 Data Management and Business Intelligence, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Sistemas de gestión de bases de datos y almacenes de datos. **II:** Análisis de Big Data en un contexto empresarial. **III:** Inteligencia Empresarial (BI) y herramientas de Visualización de Datos. **IV:** Sistemas de Gestión del Conocimiento.

Objetivos de Aprendizaje **I:** Diseñar una arquitectura de datos básica para un requisito empresarial [Usar]. **II:** Evaluar la importancia de la calidad de datos y la gobernanza [Evaluar].

Bibliografía: [Laudon and Laudon, 2021, Valacich and Schneider, 2021]

5.3 Enterprise Systems and E-Commerce, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: **I:** Sistemas de Planificación de Recursos Empresariales (ERP). **II:** Gestión de la Cadena de Suministro (SCM) y Gestión de Relaciones con el Cliente (CRM). **III:** Modelos de comercio electrónico: B2B, B2C y C2C. **IV:** Comercio móvil y sistemas de pago digital.

Objetivos de Aprendizaje **I:** Describir las áreas funcionales de un sistema ERP [Familiarizarse]. **II:** Evaluar el impacto del comercio electrónico en los modelos de negocio tradicionales [Evaluar].

Bibliografía: [Stair and Reynolds, 2017, Valacich and Schneider, 2021]

5.4 IS Development and Security, 12 hr(s)

Competencias: AG-C03,AG-C09

Tópicos: I: Ciclo de Vida de Desarrollo de Sistemas (SDLC) y metodologías Ágiles. **II:** Externalización y servicios de SI basados en la nube. **III:** Amenazas a la seguridad de la información y gestión de riesgos. **IV:** Problemas éticos y sociales en Sistemas de Información.

Objetivos de Aprendizaje I: Comparar diferentes metodologías de desarrollo de sistemas [Usar]. **II:** Formular pautas éticas para la privacidad y seguridad de los datos [Evaluar].

Bibliografía: [Laudon and Laudon, 2021, Stair and Reynolds, 2017]

6. Bibliografía

[Laudon and Laudon, 2021] Laudon, K. C. and Laudon, J. P. (2021). *Management Information Systems: Managing the Digital Firm*. Pearson, 17th edition.

[Stair and Reynolds, 2017] Stair, R. and Reynolds, G. (2017). *Principles of Information Systems*. Cengage Learning, 13th edition.

[Valacich and Schneider, 2021] Valacich, J. and Schneider, C. (2021). *Information Systems Today: Managing in the Digital World*. Pearson, 9th edition.

Universidad Nacional de Ingeniería (UNI)
Facultad de Ciencias
Escuela Profesional de Inteligencia Artificial

CS3P3 Internet de las Cosas (Electivo)

1. Información general

- **Semestre**: 10mo Sem. **Créditos**: 3
- **Horas del curso**: **Teoría**: 1 horas; **Laboratorio**: 4 horas;
- **Prerrequisitos**:
  - CS3P1 Computación Paralela y Distribuída (9no Sem-Pág 264)

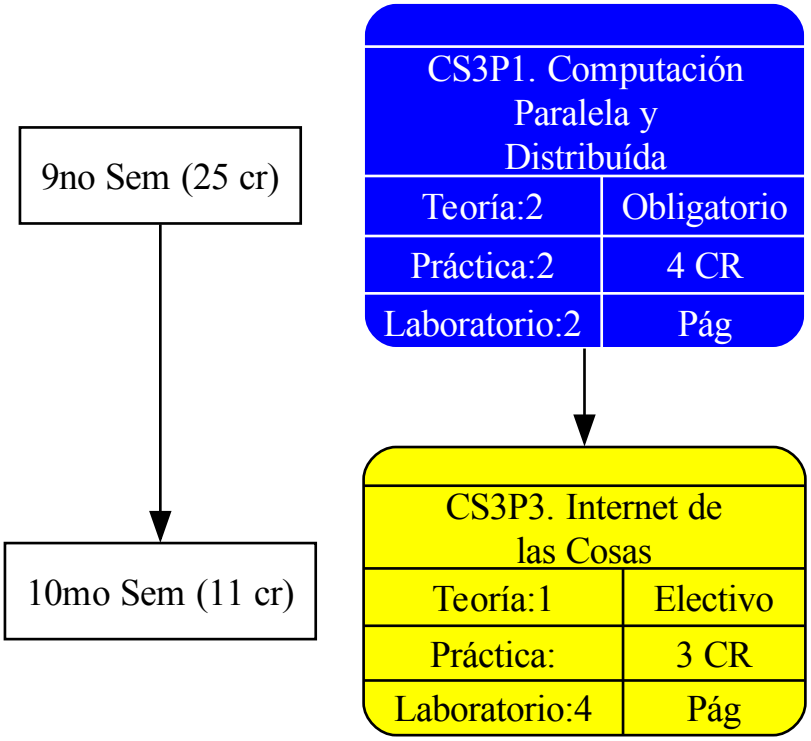


Figura 3.71: Mapa de Conexión. CS3P3 Internet de las Cosas

2. Fundamentación

El Internet de las Cosas (IoT) ha transformado cómo interactuamos con el mundo físico al conectar miles de millones de dispositivos. Este curso explora los fundamentos arquitectónicos del IoT, haciendo énfasis en la movilidad de dispositivos, la comunicación inalámbrica y las implicaciones sociales de la percepción generalizada. Los estudiantes analizarán los desafíos técnicos de los sistemas restringidos y las responsabilidades éticas asociadas con la recolección de datos a gran escala en entornos interconectados.

3. Objetivos del curso.

- Comprender la arquitectura de los sistemas IoT y el rol de la movilidad en el diseño de redes.
- Evaluar los impactos sociales y éticos de las tecnologías IoT.

- Implementar protocolos de comunicación para dispositivos restringidos y móviles.
- Analizar los riesgos de seguridad en el ciclo de vida de un ecosistema IoT.

4. Competencias Generales (*Outcomes*)

AG-C09) Diseño y Desarrollo de Soluciones: Diseña, implementa y evalúa soluciones para problemas complejos de computación. (Usar)

AG-C11) Uso de Herramientas: Aplica herramientas modernas de computación en la resolución de problemas. (Usar)

5. Contenido

5.1 Soporte de Movilidad, 12 hr(s)

Competencias: AG-C09,AG-C11

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: [Hanes et al., 2017, Buyya and Dastjerdi, 2016]

5.2 Ética, Ley y Sociedad en Seguridad, 12 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: 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. **II:** 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.

Objetivos de Aprendizaje I: Analizar los impactos sociales de la vigilancia generalizada por IoT [Evaluar]. **II:** Aplicar principios de privacidad por diseño a arquitecturas de datos de IoT [Usar].

Bibliografía: [Hanes et al., 2017]

5.3 Aplicaciones en Red, 24 hr(s)

Competencias: AG-C09,AG-C11

Tópicos: I: Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación **II:** Protocolos para redes restringidas (CoAP, MQTT, LoRaWAN).

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: [Hanes et al., 2017]

6. Bibliografía

[Buyya and Dastjerdi, 2016] Buyya, R. and Dastjerdi, A. V. (2016). *Internet of Things: Principles and Paradigms*. Morgan Kaufmann.

[Hanes et al., 2017] Hanes, D., Salgueiro, G., and Grossetete, P. (2017). *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press.

Capítulo 4

Laboratorios

BIC101. Introducción a la Computación (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.

BQU101. Química I (Obligatorio) 1er Sem, Lab: 2 hr(s)

Laboratorio de Química:

- **Equipos**: Balanzas analíticas, microscopios, espectrofotómetros, kits de vidriería.
- **Seguridad**: Campanas extractoras, duchas de emergencia, kits para derrames.
- **Reactivos**: Ácidos/bases, soluciones tampón, solventes orgánicos, indicadores de pH.

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.

BFI101. Física I (Obligatorio) 2do Sem, Lab: 4 hr(s)

Laboratorio de Física:

- **Equipos y Herramientas**: Bancos ópticos, mesas de fuerzas, péndulos, carros dinámicos, espectrómetros, sensores PASCO y módulos de medición (multímetros, fuentes, vernier).
- **Hardware y Tecnología**: Laptops de uso académico, sensores y módulos Arduino para adquisición de datos.
- **Software**: Arduino IDE, MATLAB para análisis, Microsoft Office y software especializado en adquisición de datos experimentales.
- **Servicios y Experimentos**: Prácticas de Física I y II (Mecánica, Termodinámica, Electromagnetismo, Ondas), capacitación en instrumentación y apoyo a proyectos de investigación.

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.

CS2B1. Desarrollo Basado en Plataformas (Obligatorio) 3er 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.

AI161. IA Aplicada (Obligatorio) 3er 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.

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.

CS211. Teoría de la Computación (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.

CS221. Arquitectura de Computadores (Obligatorio) 4to Sem, Lab: 4 hr(s)

Laboratorio de Redes y Ciberseguridad:
- **Hardware**: 30 PCs + racks de red (switches/routers Cisco, clusters Raspberry Pi).
- **Sistemas Operativos**: Linux (Kali, CentOS) / Windows Server.
- **Software**: Wireshark, GNS3, VMware ESXi, emuladores 8086.
- **Enfoque**: Simulación de redes, pruebas de penetración, protocolos IoT.

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.

AI261. Probabilidad y Estadistica para IA (Obligatorio) 4to 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.

CS212. Análisis y Diseño de Algoritmos (Obligatorio) 5to 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.

CS261. Inteligencia Artificial (Obligatorio) 5to 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.

CS291. Ingeniería de Software I (Obligatorio) 5to 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.

CS2S1. Sistemas Operativos (Obligatorio) 5to 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.

AI162. Política y gobernanza de la IA (Obligatorio) 5to 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.

CS311. Algoritmos para Problemas Complejos (Obligatorio) 6to 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.

AI262. Procesamiento de Lenguaje Natural (Obligatorio) 6to 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.

AI263. Introducción a Machine Learning (Obligatorio) 6to 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.

AI267. Interaccion Humano-IA (Obligatorio) 6to 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.

MA307. Matemática aplicada a la computación (Obligatorio) 6to 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.

CS231. Redes y Comunicación (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.

CS251. Computación Gráfica (Obligatorio) 7mo 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) 7mo 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.

AI265. Aprendizaje por Refuerzo (Obligatorio) 7mo 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.

AI266. IA para Ingenieria de Software (Obligatorio) 7mo 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.

CS402. Proyecto de tesis 1 (Obligatorio) 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.

AI362. Inteligencia Artificial para edge computing (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.

AI364. Inteligencia Artificial para Ciberseguridad I (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.

AI366. Sistemas de IA Distribuidos (Obligatorio) 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.

AI401. Proyecto de tesis 1 (Obligatorio) 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.

BI101. Biología (Obligatorio) 8vo Sem, Lab: 2 hr(s)

Laboratorio de Biología/Bioinformática:

- **Equipos**: Microscopios (compuestos/digitales), centrifugadoras, máquinas PCR.
- **Bioinformática**: 10 PCs con PyMol, BLAST, R/Bioconductor.
- **Muestras**: Portaobjetos preparados (microbiología, histología), kits de extracción de ADN.

CS3I1. Seguridad en Computación (Obligatorio) 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.

CS3P1. Computación Paralela y Distribuída (Obligatorio) 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.

CS403. Proyecto de tesis 2 (Obligatorio) 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.

AI402. Proyecto de tesis 2 (Obligatorio) 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.

AI268. Visión Computacional (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.

AI363. Sistemas Autónomos (Electivo) 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.

AI365. Modelos Generativos Avanzados en IA (Electivo) 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.

CS404. Taller de Investigación (Obligatorio) 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.

AI403. Taller de Investigación (Obligatorio) 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.

AI369. Robótica (Electivo) 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.

CS351. Tópicos en Computación Gráfica (Electivo) 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.

CS353. Computación Cuántica (Electivo) 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.

CS392. Tópicos en Ingeniería de Software (Electivo) 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.

CS393. Sistemas de Infomación (Electivo) 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.

CS3P3. Internet de las Cosas (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.

Capítulo 5

Profesores & Cursos

Cursos asignados por profesor

No hay profesores asignados en este documento.

Profesor asignado por curso

- Primer Semestre
  - BIC101. Introducción a la Computación (Pág 49)
    - No hay profesor asignado a esta materia
  - BMA101. Álgebra Lineal (Pág 54)
    - No hay profesor asignado a esta materia
  - BMA102. Cálculo Diferencial (Pág 57)
    - No hay profesor asignado a esta materia
  - BQU101. Química I (Pág 60)
    - No hay profesor asignado a esta materia
  - BEI101. Inglés I (Pág 65)
    - No hay profesor asignado a esta materia
- Segundo Semestre
  - CS100. Introducción a la Ciencia de la Computación (Pág 69)
    - No hay profesor asignado a esta materia
  - CS112. Programación Orientada a Objetos I (Pág 80)
    - No hay profesor asignado a esta materia
  - CS1D1. Estructuras Discretas (Pág 85)
    - No hay profesor asignado a esta materia
  - BMA103. Cálculo Integral (Pág 88)
    - No hay profesor asignado a esta materia
  - BFI101. Física I (Pág 90)
    - No hay profesor asignado a esta materia
  - BEI102. Inglés II (Pág 95)
    - No hay profesor asignado a esta materia
- Tercer Semestre
  - CS113. Programación Orientada a Objetos II (Pág 99)
    - No hay profesor asignado a esta materia
  - CS2B1. Desarrollo Basado en Plataformas (Pág 105)
    - No hay profesor asignado a esta materia
  - AI161. IA Aplicada (Pág 109)
    - No hay profesor asignado a esta materia
  - STA251. Estadística y Probabilidades (Pág 113)
    - No hay profesor asignado a esta materia
  - FG106. Teatro (Pág 118)

- No hay profesor asignado a esta materia
- BEI201. Inglés III (Pág 122)
  - No hay profesor asignado a esta materia
- EC101. Economía General (Pág 126)
  - No hay profesor asignado a esta materia
- FG120. Realidad Nacional (Pág 129)
  - No hay profesor asignado a esta materia

- Cuarto Semestre
  - CS210. Algoritmos y Estructuras de Datos (Pág 132)
    - No hay profesor asignado a esta materia
  - CS211. Teoría de la Computación (Pág 137)
    - No hay profesor asignado a esta materia
  - CS221. Arquitectura de Computadores (Pág 142)
    - No hay profesor asignado a esta materia
  - CS271. Bases de Datos I (Pág 147)
    - No hay profesor asignado a esta materia
  - CS401. Metodología de la Investigación (Pág 152)
    - No hay profesor asignado a esta materia
  - AI261. Probabilidad y Estadistica para IA (Pág 156)
    - No hay profesor asignado a esta materia
  - FG350. Liderazgo y Desempeño (Pág 159)
    - No hay profesor asignado a esta materia
  - BEI202. Inglés IV (Pág 162)
    - No hay profesor asignado a esta materia
- Quinto Semestre
  - CS212. Análisis y Diseño de Algoritmos (Pág 166)
    - No hay profesor asignado a esta materia
  - CS261. Inteligencia Artificial (Pág 171)
    - No hay profesor asignado a esta materia
  - CS291. Ingeniería de Software I (Pág 178)
    - No hay profesor asignado a esta materia
  - CS2S1. Sistemas Operativos (Pág 184)
    - No hay profesor asignado a esta materia
  - AI162. Política y gobernanza de la IA (Pág 189)
    - No hay profesor asignado a esta materia
  - MA202. Métodos Numéricos (Pág 191)
    - No hay profesor asignado a esta materia
  - BEI203. Inglés V (Pág 196)
    - No hay profesor asignado a esta materia
- Sexto Semestre
  - CS311. Algoritmos para Problemas Complejos (Pág 200)
    - No hay profesor asignado a esta materia
  - AI262. Procesamiento de Lenguaje Natural (Pág 203)
    - No hay profesor asignado a esta materia
  - AI263. Introducción a Machine Learning (Pág 206)
    - No hay profesor asignado a esta materia
  - AI267. Interaccion Humano-IA (Pág 209)
    - No hay profesor asignado a esta materia
  - MA307. Matemática aplicada a la computación (Pág 214)
    - No hay profesor asignado a esta materia
  - FG211. Ética Profesional (Pág 217)
    - No hay profesor asignado a esta materia
- Séptimo Semestre
  - CS231. Redes y Comunicación (Pág 220)
    - No hay profesor asignado a esta materia
  - CS251. Computación Gráfica (Pág 225)

- No hay profesor asignado a esta materia
- AI264. Aprendizaje Profundo (Pág 228)
  - No hay profesor asignado a esta materia
- AI265. Aprendizaje por Refuerzo (Pág 231)
  - No hay profesor asignado a esta materia
- AI266. IA para Ingenieria de Software (Pág 234)
  - No hay profesor asignado a esta materia
- AI400. Metodología de la Investigación (Pág 236)
  - No hay profesor asignado a esta materia

- Octavo Semestre
  - CS402. Proyecto de tesis 1 (Pág 239)
    - No hay profesor asignado a esta materia
  - AI362. Inteligencia Artificial para edge computing (Pág 243)
    - No hay profesor asignado a esta materia
  - AI364. Inteligencia Artificial para Ciberseguridad I (Pág 245)
    - No hay profesor asignado a esta materia
  - AI366. Sistemas de IA Distribuidos (Pág 247)
    - No hay profesor asignado a esta materia
  - AI401. Proyecto de tesis 1 (Pág 249)
    - No hay profesor asignado a esta materia
  - BI101. Biología (Pág 253)
    - No hay profesor asignado a esta materia
  - EX301. Actividades Extracurriculares (Pág 256)
    - No hay profesor asignado a esta materia
- Noveno Semestre
  - CS3I1. Seguridad en Computación (Pág 258)
    - No hay profesor asignado a esta materia
  - CS3P1. Computación Paralela y Distribuída (Pág 264)
    - No hay profesor asignado a esta materia
  - CS400. Prácticas Pre-profesionales (Pág 274)
    - No hay profesor asignado a esta materia
  - CS403. Proyecto de tesis 2 (Pág 276)
    - No hay profesor asignado a esta materia
  - AI402. Proyecto de tesis 2 (Pág 280)
    - No hay profesor asignado a esta materia
  - CB309. Bioinformática (Pág 284)
    - No hay profesor asignado a esta materia
  - AI268. Visión Computacional (Pág 288)
    - No hay profesor asignado a esta materia
  - AI363. Sistemas Autónomos (Pág 290)
    - No hay profesor asignado a esta materia
  - AI365. Modelos Generativos Avanzados en IA (Pág 292)
    - No hay profesor asignado a esta materia
- Décimo Semestre
  - CS404. Taller de Investigación (Pág 294)
    - No hay profesor asignado a esta materia
  - AI403. Taller de Investigación (Pág 298)
    - No hay profesor asignado a esta materia
  - AI369. Robótica (Pág 302)
    - No hay profesor asignado a esta materia
  - CS281. Computación en la Sociedad (Pág 306)
    - No hay profesor asignado a esta materia
  - CS351. Tópicos en Computación Gráfica (Pág 312)
    - No hay profesor asignado a esta materia
  - CS353. Computación Cuántica (Pág 316)
    - No hay profesor asignado a esta materia

- CS392. Tópicos en Ingeniería de Software (Pág 320)
  - No hay profesor asignado a esta materia
- CS393. Sistemas de Infomación (Pág 327)
  - No hay profesor asignado a esta materia
- CS3P3. Internet de las Cosas (Pág 330)
  - No hay profesor asignado a esta materia