Universidad Nacional de Ingeniería (UNI)

Facultad de Ciencias

Escuela Profesional de Ciencia de la Computación



Libro de Sílabos

– 2027-I–

Lima: 20 de octubre de 2026

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. Jose Alfredo Herrera-Quispe** Doctor en Ciencia de la Computación, UNSA (2013) Docente del Departamento de Ciencia de la Computación

Mag. Sidney Weisser Valer Quispe
Maestro en Ciencia con mención en Telemática, FIEE-UNI (2026)
Docente del Departamento de Ciencia de la Computación
email: *svalerq@uni.edu.pe*

Mag. Johnny Quispe Obregón
Docente del Departamento de Ciencia de la Computación
email: *johnny.quispe.o@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*

Índice general

Primer Semestre **8**
1.1. BIC101. Introducción a la Computación 8
1.2. BMA101. Algebra Lineal 15
1.3. BMA102. Cálculo Diferencial 18
1.4. BCH101. Química I 21
1.5. BEI101. Inglés I 27

Segundo Semestre **31**
2.1. CS100. Introducción a la Ciencia de la Computación 31
2.2. CS112. Programación Orientada a Objetos I 50
2.3. CS1D1. Estructuras Discretas 57
2.4. BMA103. Cálculo Integral 60
2.5. BFI101. Física I 63
2.6. BEI102. Inglés II 69

Tercer Semestre **74**
3.1. CS113. Programación Orientada a Objetos II 74
3.2. CS2B1. Desarrollo Basado en Plataformas 83
3.3. AI161. IA Aplicada 90
3.4. BMA104. Cálculo Diferencial e Integral Avanzado 94
3.5. STA251. Estadística y Probabilidades 98
3.6. BEI201. Inglés III 104

Cuarto Semestre **109**
4.1. CS210. Algoritmos y Estructuras de Datos 109
4.2. CS211. Teoría de la Computación 116
4.3. CS221. Arquitectura de Computadores 124
4.4. CS271. Bases de Datos I 131
4.5. CS401. Metodología de la Investigación 137
4.6. MA202. Métodos Numéricos 142
4.7. BEI202. Inglés IV 148

Quinto Semestre **153**
5.1. CS212. Análisis y Diseño de Algoritmos 153
5.2. CS261. Inteligencia Artificial 161
5.3. CS272. Bases de Datos II 175
5.4. CS291. Ingeniería de Software I 181
5.5. CS2S1. Sistemas Operativos 190

5.6. FG106. Teatro . . . 197
5.7. BEI203. Inglés V . . . 202

Sexto Semestre **207**
6.1. CS231. Redes y Comunicación . . . 207
6.2. CS311. Algoritmos para Problemas Complejos . . . 213
6.3. CS312. Estructuras de Datos Avanzadas . . . 217
6.4. CS342. Compiladores . . . 221
6.5. AI263. Introducción al Aprendizaje de Máquina . . . 226
6.6. FI201. Física Computacional . . . 229

Séptimo Semestre **232**
7.1. CS251. Computación Gráfica . . . 232
7.2. CS292. Ingeniería de Software II . . . 236
7.3. CS2H1. Experiencia de Usuario (UX) . . . 245
7.4. AI264. Aprendizaje Profundo . . . 252
7.5. FG211. Ética Profesional . . . 255
7.6. FG350. Liderazgo y Desempeño . . . 258

Octavo Semestre **262**
8.1. CS281. Computación en la Sociedad . . . 262
8.2. CS3I1. Seguridad en Computación . . . 270
8.3. CS3P1. Computación Paralela y Distribuída . . . 279
8.4. CS402. Proyecto de tesis 1 . . . 295
8.5. EX301. Actividades Extracurriculares . . . 299
8.6. ET201. Formación de Empresas de Base Tecnológica I . . . 301
8.7. AI268. Visión Computacional . . . 305
8.8. CS391. Ingeniería de Software III . . . 315
8.9. CS393. Sistemas de Información . . . 322
8.10. CS394. Arquitectura de Software y DevOps para Software en la Nube . . . 325
8.11. CS395. Calidad de Software . . . 336

Noveno Semestre **348**
9.1. CS370. Big Data . . . 348
9.2. CS400. Prácticas Pre-profesionales . . . 353
9.3. CS403. Proyecto de tesis 2 . . . 355
9.4. AI365. Modelos Generativos Avanzados en IA . . . 359
9.5. CB309. Bioinformática . . . 362
9.6. ET301. Formación de Empresas de Base Tecnológica II . . . 366
9.7. AI369. Robótica . . . 369
9.8. CS351. Tópicos en Computación Gráfica . . . 374
9.9. CS353. Computación Cuántica . . . 379
9.10. CS392. Tópicos en Ingeniería de Software . . . 383
9.11. CS3P3. Internet de las Cosas . . . 395

Décimo Semestre **398**
10.1. CS3P2. Cloud Computing 398
10.2. CS404. Taller de Investigación 404
10.3. AI367. Tópicos en Inteligencia Artificial 408
10.4. AI368. Computación Evolutiva 420

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BIC101. Introducción a la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BIC101. Introducción a la Computación
2.2 Semestre	:	1er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Introducción a la Programación y su Historia (4 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Breve historia de la computación y los lenguajes de programación. ▪ Visión general del lenguaje de programación Python. ▪ Entornos y herramientas de programación. ▪ Estructura básica y ejecución de programas. ▪ Escritura, ejecución y depuración de programas simples.	▪ Identificar tendencias importantes en la historia de la computación. [Familiarizarse] ▪ Discutir el contexto histórico de los paradigmas de lenguajes de programación. [Familiarizarse] ▪ Comparar la vida diaria antes y después de las computadoras personales e Internet. [Evaluar] ▪ Escribir, compilar y ejecutar un programa simple en Python. [Usar]
Lecturas : [**Zelle2010**], [**Guttag2013**], [**Brooks2019**]	

Unidad 2: Fundamentos de Sistemas de Tipos (6 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Un tipo como un conjunto de valores junto con un conjunto de operaciones: • Tipos primitivos (por ejemplo, números, booleanos) • Tipos compuestos construidos a partir de otros tipos (por ejemplo, registros/estructuras, uniones, arreglos, listas, funciones, referencias usando operaciones de conjuntos) ▪ Seguridad de tipos y errores causados por usar valores inconsistentemente dados sus tipos pretendidos ▪ Objetivos y limitaciones del tipado estático y dinámico: detectar y eliminar errores lo antes posible. ▪ Tipos de datos primitivos en Python (int, float, str, bool). ▪ Conversión de tipos y casting.	▪ Describir, tanto para un tipo primitivo como para uno compuesto, los valores que tienen ese tipo [Familiarizarse] ▪ Describir ejemplos de errores de programa detectados por un sistema de tipos [Familiarizarse] ▪ Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] ▪ Explicar el sistema de tipado dinámico de Python. [Familiarizarse]
Lecturas : [**Zelle2010**], [**Guttag2013**]	

Unidad 3: Fundamentos de Programación Orientada a Objetos (14 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Programación imperativa como subconjunto de la programación orientada a objetos. ▪ Definición de clases: campos, métodos y constructores. ▪ Propiedades dinámicas vs estáticas. ▪ Variables, expresiones y asignaciones. ▪ Operaciones de entrada y salida. ▪ Sentencias condicionales (if, elif, else). ▪ Estructuras de iteración (bucles for, bucles while). ▪ Bucles anidados y sentencias de control de bucle. ▪ Fundamentos del manejo de excepciones.	▪ Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Usar] ▪ Predecir y validar el flujo de control en un programa usando despacho dinámico [Evaluar] ▪ Usar mecanismos de encapsulación orientados a objetos como interfaces y miembros privados [Usar] ▪ Implementar programas usando estructuras condicionales e iterativas. [Usar] ▪ Depurar programas con errores de sintaxis y lógicos. [Usar] ▪ Diseñar e implementar algoritmos simples usando bucles. [Usar]
Lecturas : [**Zelle2010**], [**Guttag2013**], [**Brooks2019**]	

Unidad 4: Funciones y paso de parámetros (10 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición e invocación de funciones. ▪ Mecanismos de paso de parámetros (por valor, por referencia). ▪ Valores de retorno y múltiples retornos. ▪ Alcance y tiempo de vida de variables (local, global, nonlocal). ▪ Fundamentos de recursión. ▪ Expresiones lambda y funciones de orden superior. ▪ Funciones como objetos de primera clase. ▪ Documentación y prueba de funciones.	▪ Definir e implementar funciones con paso de parámetros adecuado. [Usar] ▪ Explicar la diferencia entre paso por valor y paso por referencia. [Familiarizarse] ▪ Implementar soluciones recursivas a problemas simples. [Usar] ▪ Usar expresiones lambda para construcciones de programación funcional. [Usar] ▪ Diseñar funciones con alcance y documentación apropiados. [Evaluar]
Lecturas : [**Zelle2010**], [**Guttag2013**], [**Pragmatic2009**]	

Unidad 5: Estructuras de datos básicas (10 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Listas y operaciones con listas (indexación, segmentación, mutabilidad). ▪ Tuplas e inmutabilidad. ▪ Diccionarios y mapeo clave-valor. ▪ Conjuntos y operaciones con conjuntos. ▪ Cadenas como secuencias. ▪ Comprensión de listas y expresiones generadoras. ▪ Algoritmos básicos de búsqueda y ordenamiento. ▪ Introducción a la complejidad algorítmica para operaciones básicas.	▪ Implementar programas usando listas, diccionarios y conjuntos. [Usar] ▪ Elegir estructuras de datos apropiadas para problemas dados. [Evaluar] ▪ Explicar las compensaciones entre diferentes estructuras de datos. [Familiarizarse] ▪ Implementar operaciones básicas de búsqueda y ordenamiento en listas. [Usar]
Lecturas : [**Zelle2010**], [**Guttag2013**]	

Unidad 6: Fundamentos de Programación Orientada a Objetos (10 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. ▪ Despacho dinámico: definición de llamada a método. ▪ Idiomas orientados a objetos para encapsulación: • Privacidad, ocultación de datos y visibilidad de miembros de clase. • Interfaces que revelan solo firmas de métodos. • Clases base abstractas, rasgos (*traits*) y mixins. ▪ Clases y objetos en Python. ▪ Atributos y métodos. ▪ Fundamentos de herencia y polimorfismo. ▪ Encapsulamiento y ocultación de información.	▪ Construir una jerarquía de clases simple utilizando subclases que permita reutilizar código para subclases distintas [Usar] ▪ 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] ▪ 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] ▪ Diseñar e implementar clases simples en Python. [Usar] ▪ Explicar los beneficios de la POO para la organización del código. [Familiarizarse]
Lecturas : [**Zelle2010**], [**Guttag2013**]	

Unidad 7: Estrategias Algorítmicas (6 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas: • Fuerza Bruta (por ejemplo, búsqueda lineal, ordenamiento por selección, viajante de comercio, mochila) • Disminuir y Vencer: ◦ Por una Constante (por ejemplo, ordenamiento por inserción, ordenamiento topológico) ◦ Por un Factor Constante (por ejemplo, búsqueda binaria) ◦ Por un Tamaño Variable (por ejemplo, Euclides) • Dividir y Vencer (por ejemplo, búsqueda binaria, quicksort, mergesort, Strassen) • Voraz (por ejemplo, Dijkstra, Kruskal, Mochila) • Transformar y Vencer: ◦ Simplificación de instancia (por ejemplo, encontrar duplicados mediante preordenamiento de lista) ◦ Cambio de representación (por ejemplo, heapsort) ◦ Reducción de problema (por ejemplo, mínimo común múltiplo, programación lineal) ◦ Programación dinámica (por ejemplo, Floyd, Marshall, Bellman-Ford) • Compromisos espacio vs tiempo (por ejemplo, hash) ▪ Iteración vs recursión (por ejemplo, factorial, búsqueda en árbol) ▪ Estrategias de descomposición de problemas. ▪ Técnicas de diseño de algoritmos. ▪ Estrategias de depuración y prueba. ▪ Seguimiento de la ejecución del programa.	▪ Para cada uno de los paradigmas en esta unidad: • Explicar sus características definitorias • Explicar un ejemplo que demuestre el paradigma incluyendo cómo este ejemplo satisface las características del paradigma. [Familiarizarse] ▪ Dar ejemplos de algoritmos iterativos y recursivos que resuelvan el mismo problema, explicar los beneficios y desventajas de cada enfoque [Usar] ▪ Evaluar si un enfoque voraz conduce a una solución óptima [Evaluar] ▪ Aplicar técnicas sistemáticas de depuración para localizar y corregir errores. [Usar] ▪ Analizar la complejidad algorítmica para problemas simples. [Familiarizarse]
Lecturas : [**Guttag2013**], [**Cormen2009**]	

Unidad 8: Métodos Modernos de Desarrollo y Uso de Herramientas (4 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Uso de entornos de programación modernos (IDEs, notebooks). ▪ Mejores prácticas de búsqueda y reutilización de código. ▪ Uso de bibliotecas estándar y APIs. ▪ Técnicas de depuración y prueba. ▪ Fundamentos de control de versiones. ▪ Documentación de código y guías de estilo. ▪ Entornos virtuales y gestión de paquetes.	▪ Usar herramientas de desarrollo modernas efectivamente. [Usar] ▪ Aplicar principios de reutilización de código mediante bibliotecas y APIs. [Usar] ▪ Depurar y probar programas sistemáticamente. [Usar] ▪ Seguir estándares de codificación y prácticas de documentación. [Familiarizarse] ▪ Gestionar paquetes y entornos de Python. [Familiarizarse]
Lecturas : [**Zelle2010**], [**Pragmatic2009**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BMA101. Algebra Lineal (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BMA101. Algebra Lineal
2.2 Semestre	:	1er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	3 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Matrices y Sistemas de Ecuaciones Lineales (19 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Álgebra de matrices: suma, multiplicación por escalar, producto de matrices, transpuesta, tipos de matrices e inversa de una matriz ▪ Resolución de sistemas de ecuaciones lineales mediante eliminación gaussiana y Gauss-Jordan; forma escalonada, rango y consistencia de sistemas ▪ Determinantes: definición, propiedades, regla de Cramer e interpretación geométrica ▪ Descomposiciones matriciales: LU, QR y descomposición en valores singulares (SVD)	▪ Identificar tipos de matrices y realizar operaciones básicas de álgebra matricial, incluyendo el cálculo de la inversa de una matriz [Familiarizarse] ▪ Resolver sistemas de ecuaciones lineales mediante eliminación gaussiana o Gauss-Jordan, determinando el rango y la consistencia del sistema [Usar] ▪ Aplicar la regla de Cramer y las descomposiciones LU, QR y SVD para resolver sistemas lineales y factorizar matrices [Evaluar]
Lecturas : [**Strang2016**], [**Lay2016**]	

Unidad 2: Espacios Vectoriales y Transformaciones Lineales (14 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Espacios y subespacios vectoriales; bases, dimensión e independencia lineal ▪ Transformaciones lineales, núcleo e imagen, teorema rango-nulidad y representaciones matriciales	▪ Identificar si un conjunto con operaciones dadas forma un espacio vectorial y determinar una base [Familiarizarse] ▪ Determinar el núcleo, la imagen y la representación matricial de una aplicación lineal, verificando el teorema de rango-nulidad [Usar] ▪ Aplicar transformaciones lineales para modelar y resolver problemas de cambio de base y composición de aplicaciones [Evaluar]
Lecturas : [**Strang2016**], [**Lay2016**]	

Unidad 3: Valores Propios, Producto Interno y Formas Cuadráticas (15 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Valores y vectores propios; diagonalización y el teorema espectral ▪ Espacios con producto interno, ortogonalidad, proceso de Gram-Schmidt y mínimos cuadrados ▪ Formas cuadráticas y definitud: clasificación en positiva, negativa e indefinida mediante valores propios	▪ Identificar los valores propios y vectores propios de una matriz y determinar si es diagonalizable [Familiarizarse] ▪ Calcular la diagonalización de matrices mediante el teorema espectral y aplicar el proceso de Gram-Schmidt para obtener bases ortogonales [Usar] ▪ Aplicar los métodos de mínimos cuadrados y la clasificación de formas cuadráticas mediante valores propios para resolver problemas de aproximación y optimización [Evaluar]
Lecturas : [**Strang2016**], [**Lay2016**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BMA102. Cálculo Diferencial (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BMA102. Cálculo Diferencial
2.2 Semestre	:	1er Semestre.
2.3 Créditos	:	5
2.4 Horas	:	4 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Límites y Continuidad (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición épsilon-delta de un límite y técnicas de demostración formal ▪ Leyes de los límites, límites laterales y límites en el infinito ▪ Continuidad: tipos, propiedades y el Teorema del Valor Intermedio ▪ Regla de L'Hôpital y formas indeterminadas ▪ Límites de sucesiones: convergencia, Teorema del Sándwich y Bolzano-Weierstrass	▪ Enunciar la definición épsilon-delta de un límite y explicar su significado geométrico [Familiarizarse] ▪ Evaluar límites usando leyes de límites, la regla de L'Hôpital y análisis asintótico [Usar] ▪ Demostrar continuidad o discontinuidad de una función y aplicar el Teorema del Valor Intermedio [Evaluar]
Lecturas : [**Stewart2015**], [**LarsonEdwards2014**]	

Unidad 2: Cálculo Diferencial (36 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ La derivada como límite; interpretaciones geométricas y físicas; diferenciabilidad ▪ Reglas del producto, cociente y cadena; diferenciación implícita; tasas relacionadas ▪ Teorema de Rolle, Teorema del Valor Medio y criterios de monotonía ▪ Derivadas de orden superior, polinomios de Taylor y estimación del residuo ▪ Extremos: pruebas de la primera y segunda derivada, trazado de curvas y optimización aplicada	▪ Interpretar la derivada como tasa de cambio y como pendiente de la recta tangente [Familiarizarse] ▪ Aplicar las reglas de diferenciación y la diferenciación implícita para calcular derivadas de funciones compuestas e implícitas [Usar] ▪ Resolver problemas de optimización y trazar curvas usando el Teorema del Valor Medio y las pruebas de derivadas [Evaluar]
Lecturas : [**Stewart2015**], [**LarsonEdwards2014**]	

Unidad 3: Ecuaciones Paramétricas y Coordenadas Polares (6 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Curvas paramétricas y su cálculo diferencial, aplicadas a trayectorias en cinemática ▪ Coordenadas polares y su conversión a coordenadas cartesianas	▪ Calcular la velocidad y aceleración de una trayectoria descrita paramétricamente [Usar] ▪ Convertir entre coordenadas polares y cartesianas al modelar geometrías de ingeniería [Usar]
Lecturas : [**Stewart2016**], [**Thomas2014**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BCH101. Química I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BCH101. Química I
2.2 Semestre	:	1er Semestre.
2.3 Créditos	:	5
2.4 Horas	:	4 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Materia y Energía (8 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Clasificación de la materia: elementos, compuestos, mezclas homogéneas y heterogéneas ▪ Propiedades físicas y químicas, y cambios físicos y químicos de la materia ▪ Sistema Internacional de unidades, cifras significativas y análisis dimensional ▪ Formas de energía, calor y temperatura; ley de conservación de la masa y la energía ▪ El método científico aplicado a la química: observación, hipótesis y experimentación	▪ Clasificar una muestra de materia como elemento, compuesto o mezcla, y distinguir mezclas homogéneas de heterogéneas [Familiarizarse] ▪ Aplicar el análisis dimensional y las reglas de cifras significativas en cálculos de mediciones químicas [Usar] ▪ Relacionar los cambios de energía observados con procesos físicos y químicos específicos, aplicando la ley de conservación de la energía [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 2: Estructura Atómica (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Evolución histórica del modelo atómico: Dalton, Thomson, Rutherford y Bohr ▪ Partículas subatómicas, número atómico, número de masa e isótopos ▪ Configuración electrónica, números cuánticos y el principio de Aufbau ▪ Organización de la tabla periódica: grupos, periodos y bloques ▪ Tendencias periódicas: radio atómico, energía de ionización, afinidad electrónica y electronegatividad	▪ Describir la evolución histórica del modelo atómico y las contribuciones de cada modelo a la comprensión actual del átomo [Familiarizarse] ▪ Determinar la configuración electrónica de un átomo o ion a partir de su posición en la tabla periódica [Usar] ▪ Predecir tendencias periódicas (radio atómico, energía de ionización) para comparar dos o más elementos [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 3: Enlace Químico (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Enlace iónico: formación, energía reticular y propiedades de los compuestos iónicos ▪ Enlace covalente, estructuras de Lewis y el concepto de resonancia ▪ Geometría molecular mediante la teoría de repulsión de pares de electrones de valencia (RPECV) ▪ Electronegatividad, polaridad de enlaces y polaridad molecular ▪ Fuerzas intermoleculares: dipolo-dipolo, puente de hidrógeno y fuerzas de dispersión de London	▪ Dibujar estructuras de Lewis correctas, incluyendo casos de resonancia, para moléculas e iones poliatómicos [Usar] ▪ Predecir la geometría molecular y la polaridad de una molécula aplicando la teoría RPECV [Usar] ▪ Relacionar el tipo de fuerza intermolecular presente en una sustancia con sus propiedades físicas observadas (punto de ebullición, solubilidad) [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 4: Estequiometría (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Fórmulas químicas, masa molar y el concepto de mol ▪ Balanceo de ecuaciones químicas y su interpretación cuantitativa ▪ Cálculos estequiométricos mol-mol, masa-masa y masa-mol ▪ Reactivo limitante, reactivo en exceso y rendimiento porcentual ▪ Molaridad, preparación de disoluciones y estequiometría de disoluciones	▪ Balancear ecuaciones químicas por el método de tanteo, verificando la conservación de la masa [Familiarizarse] ▪ Calcular las cantidades de reactivos y productos en una reacción química usando relaciones molares [Usar] ▪ Determinar el reactivo limitante y el rendimiento porcentual de una reacción a partir de datos experimentales [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 5: Estados de la Materia (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Leyes de los gases: Boyle, Charles, Avogadro y la ecuación del gas ideal ▪ Teoría cinético-molecular de los gases y sus aplicaciones ▪ Cambios de fase, curvas de calentamiento y diagramas de fase ▪ Sólidos cristalinos y amorfos; tipos de sólidos cristalinos ▪ Propiedades coligativas de las disoluciones: disminución de la presión de vapor, elevación ebulloscópica, descenso crioscópico y presión osmótica	▪ Aplicar la ecuación del gas ideal y las leyes de los gases para resolver problemas de presión, volumen y temperatura [Usar] ▪ Interpretar un diagrama de fase para identificar los estados y transiciones de una sustancia a distintas condiciones [Familiarizarse] ▪ Calcular las propiedades coligativas de una disolución (presión de vapor, punto de ebullición, punto de congelación) a partir de su concentración [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 6: Química Ambiental y del Agua (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Composición química del agua natural y parámetros de calidad (pH, dureza, oxígeno disuelto, DBO) ▪ Fundamentos de tratamiento de agua: coagulación, floculación y desinfección ▪ Principales contaminantes del aire y del agua y sus fuentes ▪ Ciclos biogeoquímicos del carbono, nitrógeno y fósforo ▪ Química atmosférica y el efecto invernadero	▪ Interpretar parámetros de calidad del agua para evaluar su aptitud de uso [Usar] ▪ Describir los procesos básicos de tratamiento de agua potable y residual [Familiarizarse] ▪ Identificar las principales fuentes y efectos de los contaminantes del aire y del agua [Familiarizarse] ▪ Explicar el mecanismo químico del efecto invernadero [Usar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 7: Polímeros y Geosintéticos (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Clasificación de polímeros: termoplásticos, termoestables y elastómeros ▪ Reacciones de polimerización: adición y condensación ▪ Propiedades mecánicas y químicas de los polímeros relevantes para ingeniería ▪ Geosintéticos: geotextiles, geomembranas y geomallas, y sus aplicaciones ▪ Degradación y durabilidad de polímeros en condiciones ambientales	▪ Clasificar un polímero según su estructura y comportamiento térmico [Usar] ▪ Identificar si una reacción de polimerización es por adición o por condensación [Familiarizarse] ▪ Seleccionar el tipo de geosintético adecuado según la aplicación de ingeniería civil requerida [Usar] ▪ Evaluar la durabilidad de un polímero frente a la exposición ambiental [Evaluar]
Lecturas : [**ChangGoldsby2016**], [**MamloukZaniewski2017**]	

Unidad 8: Equilibrio Químico (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ La constante de equilibrio K_c y K_p, y su relación con la estequiometría de la reacción ▪ El cociente de reacción Q y la predicción del sentido de una reacción ▪ Principio de Le Chatelier: efecto de la concentración, la presión y la temperatura ▪ Tablas ICE para el cálculo de concentraciones en el equilibrio	▪ Escribir la expresión de la constante de equilibrio para una reacción dada [Usar] ▪ Calcular concentraciones en el equilibrio usando tablas ICE [Usar] ▪ Predecir el desplazamiento de un equilibrio ante cambios de concentración, presión o temperatura usando el principio de Le Chatelier [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 9: Equilibrio Iónico e Hidrólisis de Sales (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Teorías ácido-base de Arrhenius, Brønsted-Lowry y Lewis ▪ Cálculo de pH y pOH para ácidos y bases fuertes ▪ Equilibrio de ácidos y bases débiles: K_a, K_b y grado de disociación ▪ Hidrólisis de sales y predicción del carácter ácido, básico o neutro de una disolución ▪ Disoluciones amortiguadoras (buffer) y la ecuación de Henderson-Hasselbalch	▪ Calcular el pH de disoluciones de ácidos y bases fuertes y débiles [Usar] ▪ Predecir el carácter ácido-base de una disolución de una sal a partir de la hidrólisis de sus iones [Evaluar] ▪ Calcular el pH de una disolución amortiguadora usando la ecuación de Henderson-Hasselbalch [Usar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 10: Sales, Solubilidad y Equilibrios de Precipitación (6 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ El producto de solubilidad K_{sp} y su relación con la solubilidad molar ▪ Predicción de precipitación mediante la comparación del cociente de solubilidad con K_{sp} ▪ Efecto del ion común sobre la solubilidad	▪ Calcular K_{sp} a partir de la solubilidad molar de una sal, y viceversa [Usar] ▪ Predecir si se formará un precipitado al mezclar dos disoluciones dadas sus concentraciones y el K_{sp} correspondiente [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 11: Electroquímica y Electrólisis (8 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Reacciones de oxidación-reducción y balanceo por el método del ion-electrón ▪ Celdas galvánicas, potenciales estándar de electrodo y la ecuación de Nernst ▪ Celdas electrolíticas y electrólisis; leyes de Faraday ▪ Corrosión electroquímica de metales y métodos de protección ▪ Corrosión del acero de refuerzo en concreto armado	▪ Balancear ecuaciones redox por el método del ion-electrón en medio ácido o básico [Usar] ▪ Calcular el potencial de una celda galvánica y predecir la espontaneidad de una reacción redox [Usar] ▪ Aplicar las leyes de Faraday para calcular la masa depositada o el tiempo en un proceso de electrólisis [Usar] ▪ Analizar las condiciones que favorecen la corrosión de un metal o del acero de refuerzo en concreto, y proponer métodos de protección [Evaluar]
Lecturas : [**BrownLeMayBursten2017**], [**ChangGoldsby2016**]	

Unidad 12: Química del Cemento (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Composición química del cemento Portland: silicatos, aluminatos y ferroaluminatos de calcio ▪ Reacciones de hidratación del cemento y desarrollo de resistencia ▪ Rol de cada compuesto (C_3S, C_2S, C_3A, C_4AF) en las propiedades del cemento ▪ Materiales cementantes suplementarios: puzolanas, escoria y cenizas volantes	▪ Describir la composición química del cemento Portland y sus principales compuestos [Familiarizarse] ▪ Explicar el proceso de hidratación del cemento y su relación con el desarrollo de resistencia [Usar] ▪ Evaluar el efecto de materiales cementantes suplementarios sobre las propiedades del concreto [Evaluar]
Lecturas : [**MamloukZaniewski2017**], [**ChangGoldsby2016**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BEI101. Inglés I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BEI101. Inglés I
2.2 Semestre	:	1er Semestre.
2.3 Créditos	:	2
2.4 Horas	:	4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Bienvenido a la Computación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Verbo To Be: Afirmativo, Negativo e Interrogativo. ▪ Pronombres personales y adjetivos posesivos. ▪ Expresiones numéricas (1-100): Puertos, direcciones y fundamentos binarios. ▪ Expresiones sociales en entornos académicos y de laboratorio.	▪ 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.").
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 2: Partes y Sistemas (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Posesivo 's y sustantivos plurales aplicados a componentes (the CPU's speed, the system's memory). ▪ Componentes de hardware y sus relaciones. ▪ Estructuras Adjetivo + Sustantivo para descripciones técnicas. ▪ Interacciones en el laboratorio: Preguntar sobre equipos y herramientas.	▪ El estudiante describe componentes de hardware y sus relaciones utilizando posesivos y frases nominales (e.g., "the motherboard's slots", "the server's processors").
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 3: Cómo Funciona (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Simple I (Tercera persona: He/She/It) para describir procesos. ▪ Vocabulario: Cómo operan programas, algoritmos y dispositivos. ▪ Expresar la hora e interpretar relojes y horarios de sistemas. ▪ Verbos de operación: runs, compiles, executes, stores, transmits.	▪ 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.").
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 4: Lo que Hacen los Ingenieros (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Simple II (I/You/We/They) para describir rutinas profesionales. ▪ Preguntas y negaciones con 'Do/Does' en contextos técnicos. ▪ Roles en ingeniería: developer, analyst, researcher, systems engineer. ▪ Adverbios de frecuencia para describir ciclos de trabajo y pruebas.	▪ El estudiante describe las tareas diarias y los roles profesionales de ingenieros y científicos computacionales utilizando el presente simple con precisión.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 5: Dentro de una Computadora (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Existencial: There is / There are aplicado a descripciones de arquitecturas. ▪ Preposiciones de lugar: componentes en una placa madre, archivos en un directorio. ▪ Vocabulario: memoria, almacenamiento, buses, ranuras, puertos, capas. ▪ Descripción de topología de red e infraestructura física.	▪ El estudiante describe la organización espacial de componentes de hardware y sistemas de archivos, permitiéndole interpretar y producir descripciones técnicas de arquitecturas.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 6: ¿Pueden Pensar las Computadoras? (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Verbo modal Can/Can't para describir capacidades y limitaciones computacionales. ▪ Adverbios de modo aplicados al rendimiento de sistemas. ▪ Pasado del verbo 'To Be' (Was/Were): sistemas computacionales tempranos y pioneros. ▪ Expresiones básicas para soporte técnico y resolución de problemas.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 7: Historia de la Computación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pasado Simple: Verbos regulares e irregulares para narrar la historia tecnológica. ▪ Expresiones de tiempo pasado: invented, developed, published, launched. ▪ Figuras clave: Turing, Babbage, Lovelace, von Neumann. ▪ Vocabulario: Generaciones de computadoras, hitos, inventos.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 8: Mis Metas Profesionales en Computación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción al inglés técnico para computación e ingeniería. ▪ Lectura y comprensión de especificaciones técnicas simples. ▪ Trabajo en equipos de ingeniería multiculturales: normas de comunicación. ▪ Presentación oral: "Mi rol en el campo de la computación".	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS100. Introducción a la Ciencia de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS100. Introducción a la Ciencia de la Computación
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Familiarity)

7. TEMAS

Unidad 1: Pensamiento Computacional (Parte I) (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Explicación del sistema de evaluación. ▪ Instrucciones generales del curso. ▪ Definición de Computación. ▪ La Computación como un binomio Humano-Computadora. ▪ Distorsiones en la definición de computación. ▪ La Computación como la automatización de la abstracción. ▪ Computación e Ingeniería: similitudes y diferencias. ▪ Resolución de problemas algorítmicos. ▪ Dinámica: Comprender la ejecución de un algoritmo a velocidad humana.	▪ Aplicar los conceptos fundamentales de la computación en situaciones de la vida real. [Usar] ▪ Identificar distorsiones de la Computación en situaciones de la vida real. [Usar] ▪ Identificar claramente al menos 3 contextos de uso de la palabra .[En]gineer.[en] inglés. [Evaluar] ▪ Identificar las limitaciones humanas para resolver problemas computacionales. [Usar]
Lecturas : [**Brookshear2019**]	

Unidad 2: Pensamiento Computacional (Parte II) (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descomposición (Dividir un problema en piezas más pequeñas). ▪ Abstracción (Enfocarse en lo importante). ▪ Reconocimiento de patrones (Identificar secuencias similares basadas en problemas previos). ▪ Diseño de algoritmos.	▪ Aplicar el Pensamiento Computacional a problemas del mundo real. [Usar]
Lecturas : [**Brookshear2019**]	

Unidad 3: Conceptos Generales (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Numeración binaria vs. decimal. ▪ Representación de caracteres: la tabla ASCII. ▪ Búsqueda binaria. ▪ Complejidad computacional de un algoritmo.	▪ Aplicar varios sistemas de numeración a problemas del mundo real. [Usar] ▪ Comprender la representación interna de caracteres en las tablas ASCII y UTF-8. [Familiarizarse] ▪ Aplicar la estrategia algorítmica de Divide y Vencerás. [Usar] ▪ Determinar análisis básicos de complejidad algorítmica. [Usar]
Lecturas : [**Brookshear2019**]	

Unidad 4: Representación de Datos a Nivel de Máquina (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general e historia de la arquitectura de computadoras ▪ Bits, bytes y palabras ▪ Representaciones sin signo, con signo y complemento a dos ▪ Representación de datos numéricos y bases numéricas: • Punto fijo • Punto flotante ▪ Representación de datos no numéricos ▪ Representación de registros, arreglos y tipos de datos UTF	▪ Discutir por qué todo en las computadoras son datos, incluyendo instrucciones [Debatir] ▪ Explicar cómo las representaciones numéricas de longitud fija pueden afectar la exactitud y precisión [Explicar] ▪ Describir cómo se almacenan los enteros negativos en representaciones de magnitud con signo y complemento a dos [Describir] ▪ Discutir cómo diferentes formatos pueden representar datos numéricos [Debatir] ▪ Explicar la representación a nivel de bits de datos no numéricos, como caracteres, cadenas, registros y arreglos [Explicar] ▪ Traducir datos numéricos de un formato a otro [Traducir] ▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 5: Abstracción y Representación de Programas (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Programas que toman (otros) programas como entrada, como intérpretes, compiladores, verificadores de tipos, generadores de documentación ▪ Componentes de un lenguaje: • Definiciones de alfabetos, delimitadores, oraciones, sintaxis y semántica • Sintaxis vs semántica ▪ Programa como un conjunto de oraciones significativas no ambiguas ▪ Abstracciones básicas de programación: constantes, variables, declaraciones (incluyendo declaraciones anidadas), comando, expresión, asignación, selección, iteración definida e indefinida, iteradores, función, procedimiento, módulos, manejo de excepciones ▪ Tipos de variables: estáticas, locales, no locales (*nonlocal*), globales; necesidad y problemas con variables no locales y globales. ▪ Reglas de ámbito (*scope*): estático vs dinámico; visibilidad de variables; efectos secundarios (*side-effects*). ▪ Entorno (*environment*) vs almacén (*store*) y sus propiedades ▪ Abstracción de datos y de control ▪ Mecanismos para intercambio de información entre unidades de programa como procedimientos, funciones y módulos: variables no locales, variables globales, paso de parámetros, importación-exportación entre módulos ▪ Representación de instrucciones de bajo nivel como instrucciones de máquina virtual, lenguaje ensamblador y representación binaria	▪ Ilustrar el ámbito de variables y visibilidad usando programas simples [Aplicar] ▪ Ilustrar diferentes tipos de paso de parámetros usando un lenguaje de programación pseudo simple [Aplicar] ▪ Explicar el efecto secundario usando variables globales y no locales y cómo corregir tales programas [Explicar] ▪ Explicar cómo los programas que procesan otros programas tratan a los otros programas como sus datos de entrada [Explicar] ▪ Describir una gramática y un árbol de sintaxis abstracta para un lenguaje pequeño [Describir]
Lecturas : [**Brookshear2019**]	

Unidad 6: Fundamentos de Criptografía (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Basic concepts on Cryptography ▪ Public-keys, Private-keys	▪ Explicar el papel de la criptografía en el soporte de la seguridad y la privacidad [Explicar] ▪ Discutir los riesgos de inventar los propios métodos criptográficos [Debatir] ▪ Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Debatir] ▪ Implementar y criptoanalizar cifrados clásicos [Implementar]
Lecturas : [**Brookshear2019**]	

Unidad 7: Gestión de Memoria (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento ▪ Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas y marcos ▪ Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad • Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes) ▪ Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento) ▪ Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia	▪ Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar] ▪ Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir] ▪ Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar] ▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 8: Rol y Propósito de los Sistemas Operativos (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación. ▪ Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos) ▪ Influencias de seguridad, redes, multimedia, computación paralela y distribuida	▪ Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar] ▪ Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar] ▪ Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 9: Principios del Sistema Operativo (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diseño y enfoques de software del sistema operativo (por ejemplo, monolítico, en capas, modular, microkernel, unikernel) ▪ Abstracciones, procesos y recursos ▪ La evolución del vínculo entre la arquitectura del hardware y las funciones del sistema operativo ▪ Aprovechamiento de interrupciones desde el nivel de hardware: rutinas de servicio e implementaciones. Ver también: Organización de Máquina a Nivel de Ensamblador • Interrupciones de temporizador para implementar segmentos de tiempo • Interrupciones de E/S para poner hilos bloqueados en espera sin sondeo	▪ 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] ▪ Categorizar llamadas al sistema por propósito [Categorizar] ▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 10: Fundamentos de Redes y Comunicaciones (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Importancia de las redes en la informática contemporánea, y desafíos asociados. Ver también: Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad ▪ Organización de Internet (por ejemplo, usuarios, Proveedores de Servicios de Internet, sistemas autónomos, proveedores de contenido, redes de entrega de contenido) ▪ Capas y sus roles (aplicación, transporte, red, enlace de datos y física) ▪ Elementos de red (por ejemplo, routers, switches, hubs, puntos de acceso y hosts) ▪ Conceptos básicos de colas (por ejemplo, relación con latencia, congestión, niveles de servicio, etc.)	▪ Articular la organización de Internet [Articular] ▪ Listar y definir la terminología de red apropiada [Listar/Enumerar] ▪ Describir la estructura en capas de una arquitectura de red típica [Describir]
Lecturas : [**Brookshear2019**]	

Unidad 11: Aplicaciones en Red (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Esquemas de denominación y direccionamiento (por ejemplo, DNS e Identificadores Uniformes de Recursos) ▪ Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación ▪ Diversidad de demandas de aplicaciones en red (por ejemplo, latencia, ancho de banda y tolerancia a pérdidas). Ver también: Comunicación, Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social, Equidad Digital y Accesibilidad, Impacto Social e IA ▪ Cobertura de protocolos de capa de aplicación (por ejemplo, HTTP) ▪ Interacciones con APIs de TCP, UDP y Socket. Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades	▪ Definir los principios de denominación, direccionamiento y localización de recursos [Definir] ▪ Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar]
Lecturas : [**Brookshear2019**]	

Unidad 12: Marco de Análisis de Complejidad (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Marco de Análisis de Complejidad: • Rendimiento de un algoritmo en el mejor caso, caso promedio y peor caso • Mediciones empíricas y relativas (Orden de Crecimiento) • Tamaño de entrada y operaciones primitivas • Eficiencia de tiempo y espacio ▪ Mediciones empíricas de rendimiento ▪ Compromisos tiempo-espacio en algoritmos	▪ 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] ▪ 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] ▪ Evaluar informalmente la clase de complejidad fundamental de algoritmos simples [Evaluar] ▪ 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] ▪ Explicar ejemplos que ilustren los compromisos tiempo-espacio de algoritmos [Explicar] ▪ Explicar cómo el balance del árbol afecta la eficiencia de las operaciones de árbol de búsqueda binaria [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 13: Notación Asintótica y Clases de Complejidad (1 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis de complejidad asintótica (cotas promedio y peor caso): • Notaciones formales Big-O, Big-Omega y Big-Theta • Clases de Complejidad Fundamentales y Ejemplos/Problemas Representativos: ◦ $O(1)$ Constante (por ejemplo, acceso a arreglo) ◦ $O(\log_2 n)$ Logarítmica (por ejemplo, búsqueda binaria) ◦ $O(n)$ Lineal (por ejemplo, búsqueda lineal) ◦ $O(n \log_2 n)$ Log Lineal (por ejemplo, mergesort) ◦ $O(n^2)$ Cuadrática (por ejemplo, ordenamiento por selección) ◦ $O(n^c)$ Polinomial (por ejemplo, $O(n^3)$ eliminación gaussiana) ◦ $O(2^n)$ Exponencial (por ejemplo, Mochila, Satisfactibilidad (SAT), Viajante de Comercio (TSP), todos los subconjuntos) ◦ $O(n!)$ Factorial (por ejemplo, circuito hamiltoniano, todas las permutaciones)	▪ Usando ejemplos, explicar cada una de las clases de complejidad fundamentales en esta unidad [Explicar] ▪ Para cada clase de complejidad fundamental en esta unidad, explicar un algoritmo que demuestre la complejidad de tiempo de ejecución asociada [Explicar] ▪ Explicar a una audiencia no técnica la importancia de los algoritmos tratables versus intratables usando una explicación intuitiva de la complejidad Big-O [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 14: Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (1 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Notaciones Little-o, Little-Omega y Little Theta	▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 15: Estructuras de Datos Fundamentales (3 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: • Operaciones de diccionario (insertar, eliminar, encontrar) ▪ Arreglos: • Numéricos vs no numéricos, cadenas de caracteres • Unidimensionales (vector) vs multidimensionales (matriz) ▪ Registros/Estructuras/Tuplas y Objetos ▪ Listas enlazadas (por razones históricas): • Simples vs Dobles y Lineales vs Circulares ▪ Pilas ▪ Colas y deques: • Cola de prioridad basada en montículo ▪ Tablas/mapas hash: • Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) ▪ árboles: • Binarios, n-arios y árboles de búsqueda • Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo)	▪ Para cada ADT/Estructura de Datos en esta unidad: • Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. • Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Explicar] ▪ Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Explicar] ▪ Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 16: Algoritmos Fundamentales (3 horas)

Resultados esperados: AG-C02,AG-C08,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Grafos (por ejemplo, [no]dirigidos, [a]cíclicos, [no]conexos y [no]ponderados): • Representación de grafos: lista de adyacencia vs matriz ▪ Algoritmos de búsqueda: • Complejidad $O(n)$ (por ejemplo, búsqueda lineal/secuencial en arreglo/lista) • Complejidad $O(\log_2 n)$ (por ejemplo, búsqueda binaria) • Complejidad $O(\log_b n)$ (por ejemplo, búsqueda en árbol no informada en profundidad/amplitud) ▪ Algoritmos de ordenamiento (por ejemplo, estables, inestables): • Complejidad $O(n^2)$ (por ejemplo, inserción, selección) • Complejidad $O(n \log n)$ (por ejemplo, quicksort, merge, timesort) ▪ Algoritmos de grafos: • Camino más corto (por ejemplo, Dijkstra, Floyd) • árbol de expansión mínima (por ejemplo, Prim, Kruskal) ▪ Algoritmos de ordenamiento: • Complejidad $O(n \log n)$ heapsort • Pseudo $O(n)$ complejidad (por ejemplo, bucket, counting, radix) ▪ Algoritmos de grafos: • Clausura transitiva (por ejemplo, Warshall) • Ordenamiento topológico	▪ Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar] ▪ Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar] ▪ 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] ▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 17: Algoritmos Avanzados (2 horas)

Resultados esperados: AG-C02,AG-C08,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Algoritmos de criptografía (por ejemplo, SHA-256) ▪ Algoritmos paralelos	▪ Una apreciación de la computación cuántica y su aplicación a ciertos problemas [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 18: Conceptos Fundamentales de Sistemas de Bases de Datos (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propósito y ventajas de los sistemas de bases de datos ▪ Componentes de los sistemas de bases de datos ▪ Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) ▪ Gestión de transacciones ▪ Normalización ▪ Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) ▪ Bases de datos distribuidas/sistemas basados en la nube ▪ Uso de un lenguaje de consulta declarativo	▪ Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Analizar] ▪ Enumerar los componentes de un sistema de base de datos (relacional) [Enumerar] ▪ Defender el valor de la independencia de datos [Defender] ▪ Componer una consulta simple de selección-proyección-unión en SQL [Componer] ▪ Describir las ventajas de eliminar datos duplicados repetidos [Describir] ▪ Esbozar cómo MapReduce usa paralelismo para procesar datos eficientemente [Esquematizar/Esbozar]
Lecturas : [**Brookshear2019**]	

Unidad 19: Fundamentos de Programación Orientada a Objetos (4 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Programación imperativa como subconjunto de la programación orientada a objetos. ▪ Diseño orientado a objetos: • Descomposición en objetos que poseen estado y tienen comportamiento. • Diseño de jerarquía de clases para modelado. ▪ Definición de clases: campos, métodos y constructores. ▪ Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. ▪ Despacho dinámico: definición de llamada a método. ▪ Idiomas orientados a objetos para encapsulación: • Privacidad, ocultación de datos y visibilidad de miembros de clase. • Interfaces que revelan solo firmas de métodos. • Clases base abstractas, rasgos (*traits*) y mixins.	▪ Enumerar las diferencias entre los paradigmas de programación imperativa y orientada a objetos [Enumerar] ▪ Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Componer] ▪ Usar mecanismos de encapsulación orientados a objetos como interfaces y miembros privados [Usar]
Lecturas : [**Brookshear2019**]	

Unidad 20: Control de Versiones y CI/CD (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Conceptos y mecanismos de integración de herramientas. Prácticas de Desarrollo de Software ▪ Uso de las facilidades de un IDE moderno: depuración, refactorización, búsqueda/indexación, asistentes de código basados en ML, etc. Prácticas de Desarrollo de Software	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ 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] ▪ Demostrar la capacidad de usar herramientas de software para apoyar el desarrollo de un producto de software de tamaño mediano [Demostrar]
Lecturas : [**Brookshear2019**]	

Unidad 21: Ingeniería de Requisitos (2 horas)

Resultados esperados: AG-C02,AG-C08,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Describir requisitos funcionales usando, por ejemplo, casos de uso o historias de usuario. • Usar al menos un método para documentar y estructurar requisitos funcionales. • Comprender cómo el método apoya el diseño y la implementación. • Fortalezas y debilidades de usar un enfoque específico. ▪ Obtención (*elicitation*) de requisitos. • Fuentes de requisitos, por ejemplo, usuarios, administradores o personal de soporte. • Métodos de recopilación de requisitos, por ejemplo, encuestas, entrevistas o análisis de comportamiento. ▪ Requisitos no funcionales, por ejemplo, seguridad, usabilidad o rendimiento, también llamados Atributos de Calidad. Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social ▪ Identificación y gestión de riesgos, incluyendo consideraciones éticas en torno al producto propuesto. Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo ▪ Comunicar y/o formalizar especificaciones de requisitos. ▪ Prototipado de una herramienta tanto para obtener como para validar/confirmar requisitos. ▪ Evolución del producto: cuando cambian los requisitos, cómo entender qué efecto tiene eso y qué cambios deben realizarse. ▪ Estimación de esfuerzo: • Aprender técnicas para estimar mejor el esfuerzo requerido para completar una tarea; • Practicar la estimación y compararla con cuánto tiempo toman las tareas; • 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.	▪ Comparar diferentes métodos de obtención de requisitos a lo largo de múltiples ejes [Comparar] ▪ 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] ▪ Identificar qué comportamientos son requeridos, permitidos o prohibidos a partir de un conjunto dado de requisitos y una lista de comportamientos candidatos [Analizar] ▪ Recopilar un conjunto de requisitos para un sistema de software simple [Analizar] ▪ Identificar áreas de un sistema de software que deben cambiarse, dada una descripción del sistema y un conjunto de nuevos requisitos a implementar [Analizar] ▪ Identificar los requisitos funcionales y no funcionales en un conjunto de requisitos [Analizar] ▪ Estimar el tiempo para completar un conjunto de tareas, luego comparar las estimaciones con el tiempo real tomado [Estimar] ▪ 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] ▪ Escribir una especificación de requisitos para un sistema de software simple [Escribir]

Lecturas : [**Brookshear2019**]

Unidad 22: Fundamentos de Aprendizaje Automático (1 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: • Aprendizaje supervisado: ◦ Clasificación ◦ Regresión • Aprendizaje por refuerzo • Aprendizaje no supervisado: ◦ Agrupamiento ▪ Ideas fundamentales: • Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. • Fuentes de error e indecidibilidad en el aprendizaje automático ▪ Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: • Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente	▪ 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] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] ▪ Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 23: Redes Neuronales y Ética (1 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redes neuronales básicas: • Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos • Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) ▪ ética para el Aprendizaje Automático: • Enfocarse en datos reales, escenarios reales y estudios de caso • Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas ▪ Aprendizaje profundo: • Redes feed-forward profundas (solo intuición, sin matemáticas) • Redes neuronales convolucionales (solo intuición, sin matemáticas) • Visualización de representaciones de características aprendidas de redes profundas • Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) ▪ ética para el Aprendizaje Automático: • Continuar enfocándose en datos reales, escenarios reales y estudios de caso • Privacidad • Equidad • Propiedad intelectual • Explicabilidad	▪ Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar] ▪ 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] ▪ 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]
Lecturas : [**Brookshear2019**]	

Unidad 24: Conceptos Fundamentales de Gráficos y Técnicas Interactivas (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descripción general del pipeline de gráficos por computadora y sistemas interactivos ▪ Conceptos básicos de renderizado: rasterización, trazado de rayos (*ray tracing*), modelos de sombreado ▪ Teoría del color y espacios de color ▪ Sistemas de coordenadas y transformaciones ▪ Técnicas básicas de interacción: dispositivos de entrada, manejo de eventos ▪ Conceptos avanzados de renderizado: iluminación global (*global illumination*), renderizado basado en física ▪ Hardware gráfico y técnicas de aceleración ▪ Percepción humana y cognición visual	▪ Explicar las etapas básicas del pipeline de gráficos por computadora [Explicar] ▪ Describir técnicas básicas de renderizado como rasterización y trazado de rayos [Describir] ▪ Aplicar teoría del color para diseñar gráficos visualmente efectivos [Aplicar] ▪ Implementar transformaciones de coordenadas para gráficos 2D y 3D [Implementar] ▪ Diseñar aplicaciones interactivas básicas utilizando dispositivos de entrada comunes y manejo de eventos [Diseñar] ▪ Comparar técnicas avanzadas de renderizado como iluminación global y renderizado basado en física [Comparar] ▪ Analizar el rol del hardware gráfico en acelerar tareas de renderizado [Analizar] ▪ Explicar cómo la percepción humana influye en el diseño de gráficos e interacción [Explicar]
Lecturas : [**Brookshear2019**]	

Unidad 25: Renderizado Aplicado y Técnicas (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Trazado de rayos (*ray tracing*): estructuras de aceleración, muestreo (*sampling*), sombreado ▪ Mapeo de texturas (*texture mapping*) y filtrado	▪ Implementar algoritmos de rasterización para renderizado de primitivas [Implementar] ▪ Implementar trazado de rayos básico con estructuras de aceleración [Implementar]
Lecturas : [**Brookshear2019**]	

Unidad 26: Clase de cierre: ¿Cómo funciona un motor de búsqueda como Google? (2 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis del problema. ▪ El índice no crece linealmente con el tamaño de la información indexada. ▪ El tiempo de respuesta no depende del tamaño de la "base de datos". ▪ El tiempo de respuesta no depende del número de ocurrencias encontradas. ▪ Combinar varias estructuras de datos para llegar a una solución. ▪ Analizar la escalabilidad de la solución.	▪ Comprender los principios bajo los cuales se crea un motor de búsqueda [Usar] ▪ Aplicar correctamente estructuras de datos para resolver el problema [Usar] ▪ Aplicar conceptos relacionados con la complejidad algorítmica en un motor de búsqueda [Usar].
Lecturas : [**Brookshear2019**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS112. Programación Orientada a Objetos I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS112. Programación Orientada a Objetos I
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BIC101. Introducción a la Computación. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Pragmática del Lenguaje (2 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Dominios de problemas y paradigma de programación ▪ Criterios para un buen diseño de lenguaje de programación: • Principios de diseño de lenguajes como la ortogonalidad • Definir construcciones de control e iteración • Modularización de software grande ▪ Breve revisión de paradigmas de programación. ▪ Comparación entre programación funcional e imperativa. ▪ Historia de los lenguajes de programación (énfasis en C y C++).	▪ Discutir el contexto histórico de los paradigmas de lenguajes de programación [Familiarizarse]. ▪ Evaluar las compensaciones en el diseño de lenguajes para aplicaciones específicas [Evaluar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 2: Traducción y Ejecución de Lenguajes (4 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de ejecución para JIT (Just-In-Time), compilador, intérprete ▪ Uso de código intermedio, por ejemplo, bytecode ▪ Ejecución como código nativo o dentro de una máquina virtual ▪ Máquinas virtuales y lenguajes intermedios. ▪ Modelos de compilación vs. interpretación.	▪ Explicar y comprender las diferencias entre implementaciones de lenguaje compilado, JIT e interpretado, incluyendo los beneficios y limitaciones de cada una [Familiarizarse] ▪ Diferenciar sintaxis y análisis (parsing) de semántica y evaluación [Usar] ▪ Ilustrar ambigüedad en el análisis usando if-then-else anidado/expresión aritmética y mostrar resolución usando orden de precedencia [Evaluar]
Lecturas : [**Stroustrup2013**], [**Deitel2017**]	

Unidad 3: Fundamentos de Sistemas de Tipos (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Un tipo como un conjunto de valores junto con un conjunto de operaciones: • Tipos primitivos (por ejemplo, números, booleanos) • Tipos compuestos construidos a partir de otros tipos (por ejemplo, registros/estructuras, uniones, arreglos, listas, funciones, referencias usando operaciones de conjuntos) ▪ Tipos genéricos (polimorfismo paramétrico): • Definición y ventajas del polimorfismo: paramétrico, de subtipado (subtyping), de sobrecarga (overloading) y de coerción • Comparación de tipos monomórficos y polimórficos • Comparación con polimorfismo ad-hoc (sobrecarga) y polimorfismo de subtipos • Parámetros genéricos y tipado • Uso de bibliotecas genéricas como colecciones • Comparación con polimorfismo ad hoc (sobrecarga) y polimorfismo de subtipos • Polimorfismo prescriptivo vs descriptivo • Modelos de implementación de tipos polimórficos • Subtipado ▪ Modelos de declaración (enlace, visibilidad, alcance, tiempo de vida). ▪ Visión general de la verificación de tipos en C++. ▪ Conversión de tipos y casting en C++.	▪ Definir y usar piezas de programa (como funciones, clases, métodos) que usen tipos genéricos, incluyendo para colecciones [Usar] ▪ Discutir las diferencias entre genéricos, subtipado y sobrecarga [Familiarizarse] ▪ Explicar múltiples beneficios y limitaciones del tipado estático al escribir, mantener y depurar software [Evaluar]
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 4: Fundamentos de Programación Orientada a Objetos (14 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición de clases: campos, métodos y constructores. ▪ Subclases, herencia (incluyendo herencia múltiple) y sobrescritura de métodos. ▪ Despacho dinámico: definición de llamada a método. ▪ Subtipado: • Polimorfismo de subtipos; conversiones ascendentes (*upcasts*) implícitas en lenguajes tipados. • Noción de reemplazo conductual: los subtipos actúan como el supertipo. • Relación entre subtipado y herencia. ▪ Clases y objetos en C++. ▪ Constructores y destructores. ▪ Especificadores de acceso y encapsulamiento. ▪ Jerarquías de herencia. ▪ Polimorfismo y funciones virtuales.	▪ Componer una clase a través del diseño, implementación y prueba para cumplir con los requisitos de comportamiento [Usar] ▪ Construir una jerarquía de clases simple utilizando subclases que permita reutilizar código para subclases distintas [Usar] ▪ Predecir y validar el flujo de control en un programa usando despacho dinámico [Evaluar] ▪ Diseñar clases en C++ con encapsulamiento adecuado [Usar]. ▪ Implementar jerarquías de herencia para reutilización de código [Usar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 5: Modelo de Ejecución y Memoria de Sistemas (12 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Acceso directo, indirecto e indexado a ubicación de memoria ▪ Representación en tiempo de ejecución de abstracciones de datos como variables, arreglos, vectores, registros, elementos de datos basados en punteros como listas enlazadas y árboles, y objetos ▪ Asignación y acceso de bajo nivel a estructuras de datos de alto nivel como tipos de datos básicos, arreglos n-dimensionales, vectores, registros y objetos ▪ Retorno de procedimiento como mecanismo de desasignación automática para elementos de datos locales en la pila ▪ Gestión manual de memoria: asignar, desasignar y reutilizar memoria del montón ▪ Gestión automática de memoria: recolección de basura (garbage collection) como técnica automatizada usando la noción de alcanzabilidad (reachability) ▪ Distribución de memoria (pila, montón, estática). ▪ Aritmética de punteros y operaciones. ▪ Asignación dinámica de memoria (new/delete). ▪ Fugas de memoria y punteros colgantes. ▪ Punteros inteligentes (unique_ptr, shared_ptr).	▪ 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] ▪ 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] ▪ Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Evaluar] ▪ Implementar RAII para la gestión de recursos [Usar]. ▪ Depurar errores relacionados con memoria en programas C++ [Usar].
Lecturas : [**Stroustrup2013**], [**Deitel2017**]	

Unidad 6: Programación Funcional (10 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Expresiones lambda y evaluación: • Enlace (binding) de variables y reglas de ámbito (scope). • Paso de parámetros. • Expresiones lambda anidadas y orden de reducción. ▪ Usar funciones de orden superior (tomar, devolver y almacenar funciones). ▪ Plantillas de funciones y plantillas de clases. ▪ Especialización de plantillas. ▪ Expresiones lambda en C++11/14. ▪ Conceptos de la Biblioteca de Plantillas Estándar (STL).	▪ Desarrollar funciones útiles que tomen y devuelvan otras funciones [Usar] ▪ Interpretar correctamente las variables y el ámbito léxico en un programa que usa clausuras de funciones [Evaluar] ▪ Usar mecanismos de encapsulación funcional como clausuras e interfaces modulares [Usar] ▪ Implementar funciones genéricas usando plantillas [Usar]. ▪ Usar expresiones lambda para programación funcional [Usar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 7: Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Clases de colecciones, iteradores y otros componentes comunes de biblioteca. ▪ Contenedores de secuencia (vector, list, deque). ▪ Contenedores asociativos (set, map, unordered_set). ▪ Adaptadores de contenedor (stack, queue, priority_queue). ▪ Iteradores y algoritmos. ▪ Funtores y predicados.	▪ Usar clases de colecciones e iteradores efectivamente para resolver un problema [Usar] ▪ 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] ▪ Comparar y contrastar los mecanismos para definir y proteger elementos de datos dentro de los enfoques procedural, funcional y orientado a objetos [Evaluar] ▪ Seleccionar contenedores STL apropiados para necesidades específicas [Usar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 8: Construcciones de Programación Avanzadas (6 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) ▪ Metaprogramación: macros, programación generativa, desarrollo basado en modelos (model-based development) ▪ Sobrecarga de operadores. ▪ Semántica de movimiento y referencias a valor-r. ▪ Manejo de excepciones en C++. ▪ Espacios de nombres y programación modular.	▪ Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] ▪ 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] ▪ Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] ▪ Implementar semántica de movimiento para optimización de rendimiento [Usar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**]	

Unidad 9: Scripting de Shell (4 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comandos del sistema: • Interfaz con sistemas operativos ▪ Tuberías (piping) ▪ Abstracción de archivo y operadores ▪ Programas y procesos ▪ Automatización de la compilación y pruebas de programas C++ mediante scripts (Makefiles, scripts de build/test).	▪ Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Usar]
Lecturas : [**Robbins2005**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS1D1. Estructuras Discretas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS1D1. Estructuras Discretas
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Matemáticas Discretas (12 horas)	
Resultados esperados: AG-C07,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones ▪ Conjuntos: operaciones, cardinalidad, conjuntos potencia, productos cartesianos ▪ Relaciones: propiedades, relaciones de equivalencia, órdenes parciales ▪ Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa ▪ Lógica avanzada: sistemas de demostración, completitud, decidibilidad ▪ Teoría avanzada de conjuntos: conjuntos infinitos, argumento diagonal de Cantor	▪ Explicar lógica proposicional y de predicados [Explicar] ▪ Construir demostraciones lógicas simples [Aplicar] ▪ Definir operaciones y relaciones básicas de conjuntos [Explicar] ▪ Definir propiedades de las relaciones [Explicar] ▪ Definir tipos de funciones [Explicar]
Lecturas : [**Rosen2019**], [**Lehman2018**]	

Unidad 2: Matemáticas Discretas (12 horas)	
Resultados esperados: AG-C07,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Lógica: lógica proposicional y de predicados, cuantificadores, demostraciones ▪ Lógica avanzada: sistemas de demostración, completitud, decidibilidad ▪ Teoría avanzada de conjuntos: conjuntos infinitos, argumento diagonal de Cantor ▪ Relaciones avanzadas: clausuras, retículos ▪ Funciones avanzadas: funciones recursivas, computabilidad	▪ Construir demostraciones lógicas simples [Aplicar] ▪ Explicar conceptos de conjuntos infinitos y cardinalidad [Explicar] ▪ Aplicar operaciones de clausura a relaciones [Aplicar] ▪ Determinar si una relación es una relación de equivalencia [Aplicar]
Lecturas : [**Rosen2019**], [**Velleman2019**]	

Unidad 3: Matemáticas Discretas (12 horas)	
Resultados esperados: AG-C07,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Relaciones: propiedades, relaciones de equivalencia, órdenes parciales ▪ Funciones: inyectivas, sobreyectivas, biyectivas, composición, inversa ▪ Grafos: definiciones, caminos, ciclos, árboles, conectividad ▪ Teoría avanzada de grafos: planaridad, coloración, redes	▪ Determinar si una relación es una relación de equivalencia [Aplicar] ▪ Componer funciones y encontrar inversas [Aplicar] ▪ Definir conceptos básicos de grafos [Explicar] ▪ Resolver problemas básicos de grafos [Aplicar] ▪ Analizar propiedades avanzadas de grafos [Evaluar]
Lecturas : [**Rosen2019**], [**Scheinerman2012**]	

Unidad 4: Matemáticas Discretas (12 horas)	
Resultados esperados: AG-C07,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conteo: permutaciones, combinaciones, teorema del binomio ▪ Conteo avanzado: funciones generadoras, principio de inclusión-exclusión ▪ Relaciones de recurrencia: resolver recurrencias simples ▪ Recurrencia avanzada: funciones generadoras, comportamiento asintótico	▪ Explicar permutaciones y combinaciones [Explicar] ▪ Aplicar técnicas de conteo para resolver problemas [Aplicar] ▪ Definir relaciones de recurrencia [Explicar] ▪ Resolver relaciones de recurrencia simples [Aplicar] ▪ Desarrollar funciones generadoras para problemas de conteo [Crear]
Lecturas : [**Rosen2019**], [**Grimaldi2003**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BMA103. Cálculo Integral (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BMA103. Cálculo Integral
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	5
2.4 Horas	:	4 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BMA102. Cálculo Diferencial. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Antiderivadas y Técnicas de Integración (20 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ El Teorema Fundamental del Cálculo y la antiderivada ▪ Técnicas de integración: sustitución, partes, fracciones parciales y sustitución trigonométrica	▪ Enunciar el Teorema Fundamental del Cálculo y explicar la relación entre la diferenciación y la integración [Familiarizarse] ▪ Calcular integrales definidas e indefinidas usando técnicas estándar, incluyendo sustitución e integración por partes [Usar]
Lecturas : [**Stewart2015**], [**LarsonEdwards2014**]	

Unidad 2: Integrales Impropias (8 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integrales impropias: criterios de convergencia y pruebas de comparación	▪ Determinar la convergencia de integrales impropias usando criterios de comparación y otras pruebas de convergencia [Evaluar]
Lecturas : [**Stewart2015**], [**LarsonEdwards2014**]	

Unidad 3: Sucesiones, Series y Series de Potencias (20 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sucesiones y series: pruebas de convergencia (cociente, raíz, comparación, integral) ▪ Series de potencias, radio de convergencia, series de Taylor y Maclaurin	▪ Determinar la convergencia de series de potencias y construir series de Taylor para funciones estándar [Evaluar]
Lecturas : [**Stewart2015**], [**LarsonEdwards2014**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BFI101. Física I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BFI101. Física I
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	5
2.4 Horas	:	4 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Cinemática y Dinámica de Partículas (22 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descripción vectorial de posición, velocidad y aceleración; las tres leyes de Newton y marcos de referencia inerciales ▪ Formulación y solución analítica de ecuaciones de movimiento para fuerzas constantes y variables ▪ Diagramas del espacio de fases (posición vs. velocidad) para visualizar el estado mecánico, la estabilidad y la periodicidad ▪ Integración numérica de ecuaciones de movimiento usando métodos de Euler, Verlet y Runge-Kutta ▪ Movimiento con restricciones y ligaduras holonómicas relevantes para eslabones robóticos y cuerpos articulados	▪ Identificar todas las fuerzas que actúan sobre una partícula en un escenario mecánico dado y trazar el diagrama correcto de cuerpo libre [Familiarizarse] ▪ Implementar un algoritmo de integración numérica como el método de Runge-Kutta para calcular la trayectoria de una partícula bajo fuerzas variables [Usar] ▪ Construir e interpretar diagramas del espacio de fases para caracterizar la estabilidad y periodicidad de un sistema mecánico simulado [Evaluar]
Lecturas : [**Serway2018**], [**Young2015**]	

Unidad 2: Trabajo, Energía y Leyes de Conservación (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Teorema trabajo-energía que relaciona el trabajo neto realizado con el cambio en la energía cinética ▪ Funciones de energía potencial y la relación fuerza-gradiente para campos de fuerza conservativos ▪ Conservación de la energía mecánica total y condiciones para su aplicabilidad ▪ Diagramas de energía para identificar puntos de equilibrio, puntos de retorno y estabilidad ▪ Modelos de colisiones elásticas e inelásticas: intercambio de momento y energía en implementaciones de motores de física	▪ Explicar la distinción entre fuerzas conservativas y no conservativas y las condiciones bajo las cuales se conserva la energía mecánica [Familiarizarse] ▪ Aplicar métodos de conservación de energía para calcular velocidades y posiciones en problemas de movimiento unidimensional comunes en implementaciones de motores de física [Usar] ▪ Analizar eventos de colisión elástica e inelástica, calculando velocidades post-colisión y cuantificando la disipación de energía [Evaluar]
Lecturas : [**Serway2018**], [**Tipler2010**]	

Unidad 3: Sistemas de Partículas y Momento Lineal (8 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición de centro de masa: $\vec{R}_{CM} = \frac{1}{M}\sum m_i \vec{r}_i$ ▪ Momento lineal de una partícula y de un sistema: $\vec{p} = m\vec{v}$ y $\vec{P} = M\vec{V}_{CM}$ ▪ Segunda ley de Newton para sistemas: $\vec{F}_{ext} = \frac{d\vec{P}}{dt}$ ▪ Conservación del momento lineal para sistemas aislados ▪ Distinción entre fuerzas internas y externas en un sistema ▪ Colisiones de dos cuerpos, incluyendo casos elásticos, inelásticos y completamente inelásticos ▪ Coeficiente de restitución y su significado físico en las colisiones ▪ Sistemas con masa variable, como la ecuación del cohete: $M\frac{dV}{dt} = v_{ex}\frac{dM}{dt}$ ▪ Sistemas de muchas partículas y su dinámica del centro de masa ▪ Distribuciones continuas de masa y el cálculo de su centro de masa mediante integración	▪ Localizar el centro de masa para un conjunto discreto de partículas en Una, Dos y Tres dimensiones [Usar] ▪ Determinar el momento lineal total de un sistema a partir del movimiento de su centro de masa [Usar] ▪ Aplicar la conservación del momento lineal para resolver problemas de colisión en Una y Dos dimensiones [Usar] ▪ Clasificar colisiones como elásticas, inelásticas o completamente inelásticas basándose en la conservación de la energía cinética [Familiarizarse] ▪ Calcular el coeficiente de restitución a partir de las velocidades antes y después de la colisión [Usar] ▪ Resolver la ecuación del cohete para determinar la velocidad en función de la masa expulsada [Usar] ▪ Analizar cómo las fuerzas internas afectan a las partículas individuales pero no al momento total de un sistema [Evaluar] ▪ Integrar para hallar el centro de masa de cuerpos continuos simétricos como varillas, discos y esferas [Usar]
Lecturas : [**Young2015**], [**Tipler2010**]	

Unidad 4: Movimiento Rotacional y Momento Angular (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Cinemática rotacional incluyendo desplazamiento angular, velocidad y aceleración (θ, ω, α) ▪ Definición de torque: $\vec{\tau} = \vec{r} \times \vec{F}$ y el cálculo de su magnitud ▪ Momento angular de una partícula: $\vec{L} = \vec{r} \times \vec{p}$ ▪ Forma rotacional de la segunda ley de Newton: $\vec{\tau}_{net} = \frac{d\vec{L}}{dt}$ ▪ Momento de inercia $I = \sum m_i r_i^2$ y el teorema de los ejes paralelos ▪ Energía cinética rotacional: $K_{rot} = \frac{1}{2} I\omega^2$ ▪ Conservación del momento angular para sistemas aislados ▪ Movimiento giroscópico y el fenómeno de la precesión ▪ Dinámica de rodadura sin deslizamiento: $v_{CM} = R\omega$ ▪ El efecto Coriolis como manifestación de la dinámica rotacional en marcos no inerciales	▪ Relacionar las magnitudes cinemáticas lineales y angulares para el movimiento circular [Usar] ▪ Calcular el torque respecto a un eje dado para una fuerza aplicada en un punto específico [Usar] ▪ Determinar el momento de inercia para geometrías simples como una varilla, un aro, un disco o una esfera [Usar] ▪ Aplicar la conservación del momento angular para analizar cambios en la velocidad de rotación cuando cambia el momento de inercia [Evaluar] ▪ Calcular la energía cinética rotacional de un cuerpo rígido en rotación [Usar] ▪ Analizar el movimiento de objetos que ruedan sin deslizar por un plano inclinado [Evaluar] ▪ Predecir la dirección de la precesión para una peonza o un giroscopio en rotación [Evaluar] ▪ Explicar el efecto Coriolis en términos de marcos de referencia rotatorios y su efecto sobre los proyectiles [Familiarizarse]
Lecturas : [**Serway2018**], [**Kleppner2013**]	

Unidad 5: Elasticidad (2 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición de esfuerzo (fuerza/área) y deformación unitaria $(\Delta L/L)$ ▪ Módulo de Young y ley de Hooke para deformación longitudinal: $\sigma = E\epsilon$ ▪ Módulo de corte (rigidez) y deformación por cizalladura ▪ Módulo de volumen (compresibilidad) y deformación volumétrica ▪ Límite elástico, región plástica y punto de ruptura en el diagrama esfuerzo-deformación ▪ Energía elástica almacenada en un material deformado	▪ Aplicar la ley de Hooke escalar para calcular la deformación de una barra bajo carga axial [Usar] ▪ Calcular el módulo de Young, de corte o de volumen a partir de datos experimentales esfuerzo-deformación [Usar] ▪ Identificar la región elástica, el límite elástico y el punto de fractura en una curva esfuerzo-deformación [Familiarizarse] ▪ Calcular la energía potencial elástica almacenada en un resorte o barra deformada [Usar]
Lecturas : [**Serway2018**], [**Young2015**]	

Unidad 6: Fluidos: Hidrostática, Hidrodinámica y Viscosidad (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Densidad, presión y su variación con la profundidad en un fluido en reposo: $p = p_0 + \rho g h$ ▪ Principio de Pascal y sus aplicaciones (prensa hidráulica) ▪ Principio de Arquímedes, empuje y flotación ▪ Medición de presión con manómetros y barómetros ▪ Ecuación de continuidad para flujo incompresible: $A_1 v_1 = A_2 v_2$ ▪ Ecuación de Bernoulli y conservación de energía en flujo ideal ▪ Aplicaciones: tubo de Venturi, teorema de Torricelli, tubo de Pitot ▪ Viscosidad y ley de Poiseuille para flujo laminar en tuberías ▪ Número de Reynolds y transición a flujo turbulento ▪ Ley de Stokes para la fuerza de arrastre sobre una esfera en un fluido viscoso	▪ Calcular la presión en un punto de un fluido en reposo dado la profundidad y densidad [Usar] ▪ Aplicar el principio de Arquímedes para determinar si un objeto flota, se hunde o permanece en equilibrio [Usar] ▪ Aplicar las ecuaciones de continuidad y de Bernoulli para resolver problemas de flujo en tuberías de sección variable [Usar] ▪ Interpretar lecturas de manómetros y barómetros para determinar presiones absolutas y manométricas [Evaluar] ▪ Aplicar la ley de Poiseuille para calcular el caudal de un fluido viscoso en un tubo cilíndrico [Usar] ▪ Estimar el número de Reynolds de un flujo y predecir si es laminar o turbulento [Evaluar]
Lecturas : [**Serway2018**], [**Young2015**]	

Unidad 7: Vibraciones y Ondas (14 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Movimiento armónico simple (MAS): ecuación de movimiento, posición, velocidad y aceleración ▪ Sistemas oscilantes: resorte-masa y péndulo simple/físico ▪ Energía en el movimiento armónico simple ▪ Oscilaciones amortiguadas y forzadas; introducción a la resonancia ▪ Ondas transversales y longitudinales; velocidad de propagación ▪ Ecuación de onda y velocidad de una onda en una cuerda tensa: $v = \sqrt{T/\mu}$ ▪ Principio de superposición y ondas estacionarias ▪ Ondas sonoras, intensidad y nivel de intensidad sonora (decibeles) ▪ Efecto Doppler para ondas sonoras ▪ Pulsaciones (beats) e interferencia de ondas de frecuencias cercanas	▪ Deducir la ecuación de movimiento y el período de un sistema resorte-masa y de un péndulo simple [Usar] ▪ Calcular la energía cinética, potencial y total de un oscilador armónico simple en cualquier instante [Usar] ▪ Analizar la respuesta en amplitud de un oscilador forzado cerca de la frecuencia de resonancia [Evaluar] ▪ Calcular la velocidad de propagación de una onda en una cuerda a partir de la tensión y la densidad lineal [Usar] ▪ Determinar las frecuencias y longitudes de onda de los modos normales de una onda estacionaria en una cuerda fija [Usar] ▪ Calcular el nivel de intensidad sonora en decibeles a partir de la intensidad de una onda sonora [Usar] ▪ Aplicar la fórmula del efecto Doppler para fuente y/o observador en movimiento [Usar]
Lecturas : [**Serway2018**], [**Young2015**]	

Unidad 8: Fenómeno de Transporte (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conducción de calor y la ley de Fourier: $\vec{q} = -k\nabla T$ ▪ Conductividad térmica de materiales y resistencia térmica en paredes compuestas ▪ Difusión molecular y la primera ley de Fick: $\vec{J} = -D\nabla C$ ▪ Analogía entre el transporte de calor, masa y momento (números de Prandtl y Schmidt)	▪ Aplicar la ley de Fourier para calcular el flujo de calor a través de una pared plana o compuesta [Usar] ▪ Aplicar la primera ley de Fick para calcular el flujo difusivo de una especie en un gradiente de concentración [Usar] ▪ Identificar analogías matemáticas entre los procesos de transporte de calor, masa y momento [Familiarizarse]
Lecturas : [**Serway2018**], [**Tipler2010**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BEI102. Inglés II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BEI102. Inglés II
2.2 Semestre	:	2do Semestre.
2.3 Créditos	:	2
2.4 Horas	:	4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BEI101. Inglés I. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

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

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Avances y Descubrimientos (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Repaso de Pasado Simple: Afirmativo y Negativo en contextos científicos. ▪ Expresiones de tiempo con 'ago': Cuándo se inventaron o lanzaron tecnologías. ▪ Conjunciones: and, so, but, because — estructuración de explicaciones técnicas. ▪ Vocabulario: Hitos científicos, inventos y descubrimientos tecnológicos.	▪ 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").
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 2: Datos y Variables (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sustantivos contables e incontables en computación: files, data, code, memory. ▪ Uso de some, any, a lot of para cuantificar recursos técnicos. ▪ Expresiones de cantidad: a little memory, a few errors, too many processes. ▪ Vocabulario: Tipos de datos, variables, constantes y unidades de almacenamiento.	▪ El estudiante utiliza correctamente estructuras de sustantivos contables e incontables para describir datos, recursos y estados del sistema en contextos técnicos.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 3: El Futuro de la Tecnología (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Planes futuros: Verb + to + infinitive en la planificación de proyectos. ▪ Uso de 'Will' vs 'Going to' para predicciones y lanzamientos programados. ▪ Verbos auxiliares para pronosticar: tendencias de IA, computación en la nube, computación cuántica. ▪ Vocabulario: Tecnologías emergentes y trayectorias profesionales en computación.	▪ El estudiante expresa intenciones tecnológicas y planes de proyecto, además de realizar predicciones basadas en evidencia sobre tendencias en computación e ingeniería.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 4: Comparando Tecnologías (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Adjetivos comparativos y superlativos aplicados a hardware y software. ▪ Comparación de lenguajes de programación, sistemas operativos y algoritmos. ▪ Sinónimos y antónimos en vocabulario técnico. ▪ Lectura: Artículos cortos de comparación tecnológica e informes de benchmarks.	▪ El estudiante realiza comparaciones técnicas detalladas entre sistemas, lenguajes y herramientas, utilizando estructuras comparativas y superlativas con precisión.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 5: Procesos en Ejecución (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Continuo para describir procesos técnicos activos y en curso. ▪ Uso actual: "The server is processing requests." ▪ Uso futuro con Presente Continuo: despliegues y experimentos programados. ▪ Redacción: Actualizaciones de estado e informes de avance en proyectos técnicos.	▪ El estudiante describe procesos computacionales activos, experimentos en curso y actividades técnicas programadas utilizando la forma del presente continuo.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 6: Hitos del Software (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción al Presente Perfecto para describir logros con relevancia actual. ▪ Palabras clave: ever, never, yet, just — aplicadas a versiones de software e investigación. ▪ Vocabulario: Historial de versiones, parches, actualizaciones, artículos publicados. ▪ Lectura: Noticias tecnológicas y changelogs de software.	▪ El estudiante comunica logros técnicos e hitos de software con relevancia presente, sin especificar un tiempo exacto, utilizando el presente perfecto.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 7: Desafíos Técnicos (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Adjetivos terminados en -ed e -ing para describir experiencias y procesos técnicos. ▪ Verbos modales de obligación: must, have to — aplicados a requisitos de ingeniería. ▪ Vocabulario: Desafíos de ingeniería, restricciones, trade-offs y estándares. ▪ Redacción: Planteamiento corto de un problema técnico.	▪ El estudiante describe los aspectos técnicos de los desafíos de ingeniería y expresa obligaciones y restricciones utilizando vocabulario especializado.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 8: Inglés para Tecnología II (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Terminología técnica: Dispositivos de entrada/salida, almacenamiento y periféricos. ▪ Lectura y seguimiento de instrucciones de instalación y configuración de software. ▪ Comunicación en soporte técnico: Descripción de síntomas y soluciones. ▪ Presentación: Describir un sistema computacional a una audiencia no especializada.	▪ Al finalizar el curso, el estudiante interpreta instrucciones técnicas, describe componentes de hardware y comunica conceptos básicos de computación a audiencias diversas en inglés.
Lecturas : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS113. Programación Orientada a Objetos II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS113. Programación Orientada a Objetos II
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS112. Programación Orientada a Objetos I. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Assessment)

7. TEMAS

Unidad 1: Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Clases de colecciones, iteradores y otros componentes comunes de biblioteca. ▪ Contenedores asociativos (std::set, std::map, variantes no ordenadas). ▪ Adaptadores de contenedor (stack, queue, priority_queue). ▪ Algoritmos avanzados de STL. ▪ Funtores y predicados. ▪ Categorías de iteradores y rasgos.	▪ Usar clases de colecciones e iteradores efectivamente para resolver un problema [Usar] ▪ 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] ▪ 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] ▪ Implementar asignadores personalizados para contenedores STL [Usar].
Lecturas : [**Stroustrup2013**], [**Josuttis2019**], [**Vandevoorde2017**]	

Unidad 2: Sistemas de Tipos: Polimorfismo y Tipado Complementario (12 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipos genéricos (polimorfismo paramétrico): • Definición y ventajas del polimorfismo: paramétrico, de subtipado (subtyping), de sobrecarga (overloading) y de coerción • Comparación de tipos monomórficos y polimórficos • Comparación con polimorfismo ad-hoc (sobrecarga) y polimorfismo de subtipos • Parámetros genéricos y tipado • Uso de bibliotecas genéricas como colecciones • Comparación con polimorfismo ad hoc (sobrecarga) y polimorfismo de subtipos • Polimorfismo prescriptivo vs descriptivo • Modelos de implementación de tipos polimórficos • Subtipado ▪ SFINAE (Fallo de Sustitución No es un Error): fundamento histórico de la programación genérica restringida en C++, base de bibliotecas anteriores a C++20. ▪ Concepts (C++20): sintaxis declarativa para restringir parámetros de plantilla, sucesor moderno de SFINAE con mejores diagnósticos de error. ▪ if constexpr (C++17): ramificación en tiempo de compilación, reemplazo de gran parte del despacho por etiquetas (tag dispatch) antes resuelto con SFINAE. ▪ Reenvío perfecto y referencias universales. ▪ Plantillas variádicas. ▪ Cómputo en tiempo de compilación con constexpr y consteval.	▪ Definir y usar piezas de programa (como funciones, clases, métodos) que usen tipos genéricos, incluyendo para colecciones [Evaluar] ▪ Discutir las diferencias entre genéricos, subtipado y sobrecarga [Evaluar] ▪ Aplicar Concepts para restringir parámetros de plantilla y mejorar diagnósticos de error [Usar]. ▪ Distinguir cuándo usar Concepts, if constexpr o SFINAE heredado al mantener código C++ existente [Evaluar]. ▪ Implementar reenvío perfecto para paso eficiente de argumentos [Usar].
Lecturas : [**Stroustrup2013**], [**Vandevoorde2017**], [**StroustrupTour2022**]	

Unidad 3: Gestión de Memoria (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Asignación y acceso de bajo nivel a estructuras de datos de alto nivel como tipos de datos básicos, arreglos n-dimensionales, vectores, registros y objetos ▪ Retorno de procedimiento como mecanismo de desasignación automática para elementos de datos locales en la pila ▪ Gestión manual de memoria: asignar, desasignar y reutilizar memoria del montón ▪ Gestión automática de memoria: recolección de basura (garbage collection) como técnica automatizada usando la noción de alcanzabilidad (reachability) ▪ Lvalues y rvalues en C++11/14/17/20. ▪ Constructores de movimiento y operadores de asignación de movimiento. ▪ Referencias a valor-r y colapso de referencias. ▪ Patrón RAII (Adquisición de Recursos es Inicialización). ▪ Personalización de punteros inteligentes. ▪ Detección y prevención de fugas de memoria.	▪ 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] ▪ 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] ▪ Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Evaluar] ▪ Implementar semántica de movimiento para optimización de recursos [Usar]. ▪ Diseñar clases RAII para gestión de recursos segura ante excepciones [Usar].
Lecturas : [**Meyers2014**], [**Stroustrup2013**]	

Unidad 4: Computación Paralela y Distribuida (12 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Seguridad (safety) y vivacidad (liveness): • Condiciones de carrera (race conditions) • Dependencias/precondiciones • Modelos de fallos (fault models) • Terminación ▪ Modelos de programación: • Modelo de actores (*actor models*) • Modelos procedurales y reactivos • Modelos de programación síncrona/asíncrona • Paralelismo de datos (*data parallelism*) ▪ Comunicación y coordinación: • Mutexes • Paso de mensajes (*message-passing*) • Memoria compartida (*shared memory*) • Cobegin-coend • Monitores • Canales (*channels*) • Hilos (*threads*) • Guardas (*guards*) ▪ std::thread y gestión de hilos. ▪ std::jthread (C++20): unión automática (RAII) y cancelación cooperativa vía stop_token. ▪ Mutexes, candados y variables de condición. ▪ Mecanismos de sincronización de C++20: std::latch, std::barrier, std::counting_semaphore. ▪ Operaciones atómicas y programación sin bloqueo. ▪ std::async y futures. ▪ 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). ▪ Patrones de diseño seguros para hilos. ▪ Carreras de datos y prevención de interbloqueos.	▪ Implementar programas concurrentes correctos usando múltiples modelos de programación, como memoria compartida, actores, futuros, construcciones de sincronización y primitivas de paralelismo de datos [Usar] ▪ Usar construcciones de sincronización como métodos monitorizados/sincronizados en un programa simple [Usar] ▪ Usar un modelo de paso de mensajes para analizar un protocolo de comunicación [Evaluar] ▪ Implementar estructuras de datos sin bloqueo usando operaciones atómicas [Usar]. ▪ Diferenciar std::thread de std::jthread y justificar cuándo usar cada uno [Usar]. ▪ Analizar la seguridad en hilos de algoritmos concurrentes [Evaluar].
Lecturas : [**Williams2019**], [**Stroustrup2013**]	

Unidad 5: Construcciones de Programación Avanzadas (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) ▪ Metaprogramación: macros, programación generativa, desarrollo basado en modelos (model-based development) ▪ Herencia múltiple y herencia virtual. ▪ Patrón CRTP (Patrón de Plantilla Curiosamente Recursivo). ▪ deducing this (C++23): simplifica CRTP al eliminar la necesidad de static_cast<Derived&>(*this) para polimorfismo estático. ▪ Plantillas de expresión. ▪ Rasgos de tipos e introspección en tiempo de compilación. ▪ Diseño basado en políticas. ▪ Diseño de interfaz para bibliotecas.	▪ Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] ▪ 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] ▪ Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Evaluar] ▪ Implementar CRTP para polimorfismo estático [Usar]. ▪ Diseñar interfaces de biblioteca extensibles usando diseño basado en políticas [Usar].
Lecturas : [**Stroustrup2013**], [**Gregor2017**]	

Unidad 6: Modelo de Ejecución y Memoria de Sistemas (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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 ▪ Estructuras de datos para traducción, ejecución, traducción y movilidad de código como pila, montón (heap), alias (compartición usando punteros), secuencia indexada y cadena ▪ 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. ▪ Interfaz de C++ con otros lenguajes (FFI). ▪ Llamadas al sistema e interacción con el SO. ▪ E/S mapeada en memoria. ▪ Perfilado de rendimiento y evaluación comparativa. ▪ Técnicas de optimización para HPC.	▪ Explicar cómo se ejecuta una construcción central del lenguaje, como abstracciones de datos y abstracciones de control [Usar] ▪ 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] ▪ 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] ▪ Perfilar y optimizar programas C++ para rendimiento [Usar]. ▪ Explicar cómo el compilador implementa el polimorfismo en tiempo de ejecución mediante vtables [Evaluar]. ▪ Interfazar código C++ con bibliotecas C [Usar].
Lecturas : [**Stroustrup2013**], [**Williams2019**]	

Unidad 7: Construcciones de Programación Avanzadas (6 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Abstracciones orientadas a objetos: herencia múltiple, mixins, rasgos (traits), multimétodos (multimethods) ▪ Patrones creacionales, estructurales y de comportamiento. ▪ Patrones singleton seguros para hilos. ▪ Modismos de diseño modernos en C++. ▪ Inyección de dependencias e inversión de control.	▪ Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] ▪ 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] ▪ Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] ▪ Implementar patrones de diseño seguros para hilos [Usar]. ▪ Evaluar compensaciones arquitectónicas en sistemas C++ [Evaluar].
Lecturas : [**Gamma1994**], [**Stroustrup2013**]	

Unidad 8: Construcciones de Programación Avanzadas (6 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Abstracciones de control: manejo de excepciones, continuaciones (continuations), mónadas (monads). ▪ Optimizaciones del compilador y banderas. ▪ Metodologías de evaluación comparativa. ▪ Estructuras de datos amigables con la caché. ▪ Predicción de bifurcación y alineación. ▪ Reducción del tiempo de compilación. ▪ Integración continua para pruebas de rendimiento.	▪ Usar varias construcciones e idiomas de programación avanzados correctamente [Usar] ▪ 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] ▪ Discutir cómo varias construcciones de programación avanzadas interactúan con la definición e implementación de otras características del lenguaje [Familiarizarse] ▪ Aplicar técnicas de optimización para mejorar el rendimiento del programa [Usar]. ▪ Interpretar resultados de evaluación comparativa para guiar decisiones de optimización [Usar].
Lecturas : [**Meyers2014**], [**Stroustrup2013**]	

Unidad 9: Control de Versiones y CI/CD (4 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Sistemas de construcción para C++: CMake, el estándar de facto de la industria. ▪ Alternativas modernas: xmake, Meson y Bazel – comparación de enfoques declarativos frente a basados en Lua/Python. ▪ Organización de proyectos multi-archivo: unidades de traducción, cabeceras, bibliotecas estáticas y dinámicas. ▪ Gestión de dependencias en C++ (vcpkg, Conan).	▪ 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] ▪ Configurar un proyecto C++ multi-archivo usando un sistema de construcción declarativo [Usar]. ▪ Comparar CMake con al menos una alternativa moderna (xmake, Meson o Bazel) en velocidad de compilación y reproducibilidad [Evaluar].
Lecturas : [**Stroustrup2013**], [**StroustrupTour2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS2B1. Desarrollo Basado en Plataformas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS2B1. Desarrollo Basado en Plataformas
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS112. Programación Orientada a Objetos I. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Aspectos Comunes: Plataformas, APIs y Restricciones (5 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de las plataformas de desarrollo (es decir, web, móvil, juegos, robótica, embebida e interactiva). • Entrada/sensores/dispositivos de control/dispositivos hápticos • Limitaciones de recursos ◦ Computacional ◦ Almacenamiento de datos ◦ Memoria ◦ Comunicación • 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 • Salida/actuadores/dispositivos hápticos ▪ Programación mediante Interfaces de Programación de Aplicaciones (API) específicas de la plataforma vs construcción de aplicaciones tradicional ▪ Programación bajo restricciones y requisitos de la plataforma (por ejemplo, herramientas de desarrollo disponibles, desarrollo, consideraciones de seguridad) Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad	▪ Enumerar las restricciones de la programación móvil [Listar/Enumerar]
Lecturas : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unidad 2: Aspectos Comunes: Lenguajes de Plataforma y Patrones Web (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de los Lenguajes de plataforma (por ejemplo, Python, Swift, Lua, Kotlin) ▪ Técnicas para aprender y dominar un lenguaje de programación específico de la plataforma	▪ Enumerar las características de los lenguajes de scripting [Listar/Enumerar] ▪ Describir el modelo de tres capas (three-tier) de la programación web [Describir] ▪ Describir cómo se mantiene el estado en la programación web [Describir]
Lecturas : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unidad 3: Frontend, Frameworks y Estándares Web (12 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Lenguajes de programación web (por ejemplo, HTML5, JavaScript, Typescript, CSS) ▪ Plataformas, frameworks o meta-frameworks web • Servicios en la nube • API, Componentes Web (Web Components) ▪ Software como Servicio (SaaS) ▪ Estándares web como el modelo de objetos del documento (DOM), accesibilidad Accesibilidad y Diseño Inclusivo ▪ Consideraciones de Seguridad y Privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad	▪ Comparar y contrastar la programación web con la programación de propósito general [Comparar] ▪ Describir las diferencias entre Software-como-Servicio (SaaS) y productos de software tradicionales [Describir] ▪ Discutir cómo los estándares web impactan el desarrollo de software [Debatir] ▪ Evaluar una aplicación web existente contra los estándares web actuales [Evaluar]
Lecturas : [**Flanagan2020**], [**Duckett2014**], [**Fielding:2000:Rest**]	

Unidad 4: Plataformas Móviles: Fundamentos del Desarrollo (6 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollo con • Lenguajes de programación móvil • Entornos de programación móvil ▪ Restricciones de la plataforma móvil • Diseño de interfaz de usuario Comprender al Usuario: Objetivos individuales e interacciones con otros • Seguridad ▪ Acceso • Acceso a datos a través de APIs Construcción de Consultas • Diseño de puntos finales de API (endpoints) para aplicaciones móviles: trampas y consideraciones de diseño • Interfaces de red y web Modelado de Datos, Fundamentos de Redes y Comunicaciones	▪ Comparar la programación móvil con la programación de propósito general [Comparar] ▪ Evaluar los pros y los contras del desarrollo de aplicaciones móviles nativas y multiplataforma [Evaluar]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 5: Plataformas Móviles: Arquitectura y Restricciones (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollo • Desarrollo nativo (native) versus multiplataforma (cross-platform) • Patrones de diseño/arquitectura de software para aplicaciones móviles Diseño de API y Modelado, Principios y Arquitectura de Software, Calidad y Evaluación del Diseño ▪ Restricciones de la plataforma móvil • Diseño de interfaz de usuario responsivo (responsive) Accesibilidad y Diseño Inclusivo • Heterogeneidad y movilidad de los dispositivos • Diferencias en las experiencias de usuario (por ejemplo, entre aplicaciones móviles y basadas en web) • Compensación (tradeoff) entre energía y rendimiento ▪ Especificación y pruebas Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas	▪ Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] ▪ Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] ▪ Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 6: Plataformas Móviles: Capacidades y Computación Asíncrona (3 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Capacidades (*affordances*) de la computación móvil • Aplicaciones conscientes de la ubicación (*location-aware*) • Computación impulsada por sensores (por ejemplo, giroscopio, acelerómetro, datos de salud de un reloj) • Telefonía y mensajería instantánea • Realidad aumentada Inmersión ▪ Computación asíncrona Algoritmos • Diferencia con la programación síncrona tradicional • Manejo del éxito mediante devoluciones de llamada (*callbacks*) • Manejo de errores de forma asíncrona • Pruebas de código asíncrono y problemas típicos en las pruebas	▪ Desarrollar una aplicación móvil consciente de la ubicación con integración de API de datos [Crear] ▪ Construir una aplicación móvil impulsada por sensores capaz de registrar datos en un servidor remoto [Crear] ▪ Crear una aplicación de comunicación que incorpore telefonía y mensajería instantánea [Crear]
Lecturas : [**Deitel2017**], [**Hellman2013**]	

Unidad 7: SPD-SEP/Interactiva (10 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pautas éticas al usar modelos de IA para ayudar en el periodismo y la creación de contenido ▪ Rendición de cuentas (*accountability*) por las salidas generadas por IA ▪ Comportamiento entre programadores de prompts (*prompt programmers*) y desarrolladores de IA ▪ Confianza con el público al usar modelos de IA	▪ Indicar un marco para la rendición de cuentas (*accountability*) en el despliegue de modelos de IA, incluyendo documentación clara y atribución [Evaluar] ▪ Discutir códigos de conducta éticos y estándares profesionales relevantes para la programación de prompts (*prompt programming*) y el desarrollo de IA [Debatir] ▪ Crear planes y materiales de comunicación para educar al público sobre las capacidades, limitaciones y salvaguardas éticas de la IA [Crear]
Lecturas : [**Duckett2014**], [**Deitel2017**]	

Unidad 8: Plataformas Interactivas de Datos y Análisis (6 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Plataformas de Cuadernos Computacionales (por ejemplo, Cuadernos Jupyter; Google Colab; R; SPSS; Observable) ▪ Plataformas de Análisis de Datos y SQL en la Nube (por ejemplo, BigQuery) Construcción de Consultas • Apache Spark • Visualizaciones de Datos Visualización ▪ Presentaciones Interactivas Respaldadas por Datos ▪ Herramientas de Diseño con Bucles de Retroalimentación de Baja Latencia (*Low-Latency Feedback Loops*) • Herramientas de renderizado • Herramientas de diseño gráfico	▪ Analizar grandes conjuntos de datos de forma interactiva [Analizar] ▪ Crear una pista de acompañamiento para una interpretación musical, como la codificación en vivo (*live coding*) [Crear] ▪ Crear cuadernos computacionales convincentes que construyan una narrativa para un objetivo/historia periodística dada [Crear] ▪ Implementar código interactivo que use un conjunto de datos y genere gráficos exploratorios [Implementar] ▪ Contextualizar los atributos de diferentes estilos de análisis de datos, como interactivo versus tubería (*pipeline*) de ingeniería [Contextualizar] ▪ Escribir un programa usando una plataforma de computación de cuadernos (por ejemplo, búsqueda, ordenación o manipulación de grafos) [Escribir]
Lecturas : [**Flanagan2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.

- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI161. IA Aplicada (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI161. IA Aplicada
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BIC101. Introducción a la Computación. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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. (Familiarity)

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

7. TEMAS

Unidad 1: Fundamentos de IA y Panorama Global (16 horas)	
Resultados esperados: AG-C01,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ ¿Qué es la IA? Definiciones, historia y panorama global actual. ▪ Tipos de sistemas de IA: chatbots, generadores de imágenes, asistentes de investigación entre regiones. ▪ Ecosistemas occidentales de IA: OpenAI, Anthropic, Google, Microsoft. ▪ Ecosistemas chinos de IA: DeepSeek, Kimi, ERNIE, Zhipu AI, Baidu. ▪ Capacidades y limitaciones de la IA: análisis comparativo entre plataformas. ▪ Alfabetización digital en la era de la IA: pensamiento crítico sobre diversos resultados de IA.	▪ Aplicar conocimientos fundamentales de IA para identificar tipos de sistemas y sus capacidades [Usar]. ▪ Analizar sistemas inteligentes globales y sus características distintivas [Evaluar]. ▪ Evaluar críticamente los resultados de IA en diferentes contextos culturales [Evaluar].
Lecturas : [**Ng2024**], [**Mollick2024**], [**UNESCO2023**], [**Tencent2024**]	

Unidad 2: Interacción Efectiva con IA e Ingeniería de Prompts Multiplataforma (20 horas)	
Resultados esperados: AG-C01,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de escritura efectiva de prompts: claridad, contexto, restricciones entre plataformas. ▪ Patrones de prompts: persona, plantilla, cadena de pensamiento, few-shot para diferentes sistemas de IA. ▪ Herramientas occidentales: técnicas específicas para ChatGPT, Claude, Gemini, Copilot. ▪ Herramientas chinas: características específicas y mejores prácticas para DeepSeek, Kimi, ERNIE. ▪ Estrategias multiplataforma: aprovechar diferentes fortalezas de IA para tareas complejas. ▪ Refinamiento iterativo: cómo mejorar prompts basándose en resultados de diversos sistemas. ▪ Sesiones prácticas: talleres comparativos de escritura de prompts entre plataformas.	▪ Diseñar soluciones de prompts efectivas para diversos sistemas de IA [Evaluar]. ▪ Desarrollar flujos de trabajo multiplataforma que integren diferentes herramientas de IA [Usar]. ▪ Implementar estrategias de refinamiento iterativo para optimizar resultados de IA [Usar].
Lecturas : [**White2023**], [**PromptEngineering2024**], [**Microsoft2024**], [**DeepSeek2024**]	

Unidad 3: Aplicaciones de IA entre Disciplinas y Plataformas (16 horas)	
Resultados esperados: AG-C01,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ IA para investigación: revisión de literatura, análisis de datos usando múltiples asistentes de IA. ▪ IA para escritura: documentos académicos, informes con verificación multiplataforma. ▪ IA para resolución de problemas: cálculos científicos, diseño de ingeniería con herramientas especializadas. ▪ IA para creatividad: lluvia de ideas, desarrollo de conceptos entre contextos culturales. ▪ Fortalezas específicas por plataforma: cuándo usar herramientas de IA occidentales vs. chinas. ▪ Talleres disciplinares: aplicaciones personalizadas usando diversos ecosistemas de IA. ▪ Estudios de caso: aplicaciones en el mundo real en investigación científica entre regiones.	▪ Aplicar fundamentos de IA para resolver problemas específicos de cada disciplina [Usar]. ▪ Diseñar soluciones que integren múltiples sistemas inteligentes para tareas complejas [Evaluar]. ▪ Desarrollar aplicaciones prácticas usando herramientas de IA occidentales y chinas [Usar]. ▪ Implementar flujos de trabajo que aprovechen fortalezas específicas de cada plataforma [Evaluar].
Lecturas : [**Mollick2023**], [**Google2024**], [**Alibaba2024**], [**OpenAI2024**]	

Unidad 4: Ética Global en IA y Uso Responsable (12 horas)	
Resultados esperados: AG-C01,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comprensión de sesgos en IA: perspectivas culturales occidentales y orientales. ▪ Marcos éticos: comparando la Ley de IA de la UE, regulaciones chinas y estándares globales. ▪ Integridad académica: citación apropiada y uso de IA en trabajos académicos entre plataformas. ▪ Privacidad y seguridad de datos: diferencias regionales en el manejo de datos de IA. ▪ Sensibilidad cultural: navegando resultados de IA en contextos globales. ▪ Impacto ambiental: consideraciones de sostenibilidad de diferentes sistemas de IA. ▪ Desarrollo de pautas personales para el uso ético de IA en entornos internacionales. ▪ Estudios de caso: dilemas éticos en el despliegue de IA entre regiones.	▪ Demostrar responsabilidad profesional en el uso de tecnologías de IA [Evaluar]. ▪ Aplicar principios éticos profesionales en contextos globales de IA [Usar]. ▪ Desarrollar pautas éticas para mitigar sesgos en sistemas de IA [Evaluar]. ▪ Evaluar implicaciones éticas del despliegue de IA en diferentes culturas [Usar].
Lecturas : [**UNESCO2023**], [**EU2024**], [**Bender2021**], [**ChinaAI2024**], [**Tencent2023**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BMA104. Cálculo Diferencial e Integral Avanzado (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BMA104. Cálculo Diferencial e Integral Avanzado
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	5
2.4 Horas	:	4 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BMA102. Cálculo Diferencial. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El cálculo diferencial e integral avanzado es fundamental para la comprensión rigurosa de los fenómenos multidimensionales en ingeniería y ciencias de la computación. Este curso aborda el estudio de funciones vectoriales, el cálculo multivariable, las transformaciones diferenciables, las integrales múltiples y el cálculo integral en variedades, incluyendo los grandes teoremas de Green, Stokes y Gauss. Estas herramientas son esenciales para modelar sistemas complejos, optimizar funciones multivariables y resolver problemas en gráficos por computadora, visión artificial y simulaciones físicas.

5. OBJETIVOS

- Estudiar las funciones vectoriales de variable real y la geometría diferencial de curvas en el espacio.
- Aplicar el cálculo diferencial a funciones de varias variables: derivadas parciales, gradiente, plano tangente, optimización y Teorema de Taylor.
- Analizar transformaciones diferenciables entre espacios vectoriales mediante la matriz Jacobiana y los teoremas de función inversa e implícita.
- Evaluar integrales múltiples en distintos sistemas de coordenadas y calcular áreas, volúmenes y centros de masa.
- Calcular integrales de línea y aplicar el Teorema de Green y el concepto de campos conservativos.
- Calcular integrales de superficie y aplicar los Teoremas de Stokes y de la Divergencia.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Funciones Vectoriales de una Variable Real (14 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición, dominio y rango de funciones vectoriales de una variable real ▪ Límites, continuidad y derivadas de funciones vectoriales ▪ Longitud de arco y parametrización natural de una curva ▪ Vectores tangente, normal y binormal; triedro móvil de Frenet ▪ Curvatura, torsión y fórmulas de Frenet-Serret	▪ Identificar límites, continuidad y derivadas de funciones vectoriales de una variable real [Familiarizarse] ▪ Calcular vectores tangente, normal y binormal y reparametrizar una curva por longitud de arco [Usar] ▪ Calcular curvatura y torsión usando las fórmulas de Frenet-Serret [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

Unidad 2: Cálculo Diferencial Multivariable (18 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Límites y continuidad de funciones de varias variables en $\mathbb{R}^n$ ▪ Derivadas parciales, derivadas direccionales y el vector gradiente ▪ Plano tangente, diferencial total y aproximación lineal ▪ Extremos de funciones multivariables y multiplicadores de Lagrange ▪ Teorema de Taylor para funciones de varias variables	▪ Evaluar límites y determinar la continuidad de funciones de varias variables [Familiarizarse] ▪ Calcular gradientes, derivadas direccionales y planos tangentes para funciones multivariables [Usar] ▪ Aplicar los multiplicadores de Lagrange para resolver problemas de optimización con restricciones en computación y ciencia de datos [Usar] ▪ Expandir funciones de varias variables usando el Teorema de Taylor y estimar cotas de error [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

Unidad 3: Transformaciones Multivariables (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Transformaciones de $\mathbb{R}^n$ a $\mathbb{R}^m$ y su diferenciabilidad ▪ Matriz jacobiana y determinante jacobiano ▪ Cambio de variable en coordenadas polares, cilíndricas y esféricas ▪ Teorema de la función inversa ▪ Teorema de la función implícita en contextos multivariables	▪ Describir transformaciones entre sistemas de coordenadas usando la matriz jacobiana [Familiarizarse] ▪ Calcular el jacobiano y aplicar cambio de variable para evaluar integrales múltiples [Usar] ▪ Aplicar los teoremas de la función inversa y de la función implícita para analizar invertibilidad local [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

Unidad 4: Integrales Múltiples (18 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integrales dobles sobre rectángulos y regiones generales; integrales iteradas (teorema de Fubini) ▪ Integrales dobles en coordenadas polares y fórmula de cambio de variable ▪ Integrales triples en coordenadas cartesianas y aplicaciones a volumen y masa ▪ Integrales triples en coordenadas cilíndricas y esféricas ▪ Aplicaciones: área, volumen, centros de masa y momentos de inercia	▪ Evaluar integrales dobles y triples sobre regiones generales usando integrales iteradas [Usar] ▪ Aplicar cambios de coordenadas en sistemas polares, cilíndricos y esféricos para simplificar integrales múltiples [Usar] ▪ Calcular áreas, volúmenes y centros de masa usando integrales múltiples [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

Unidad 5: Integrales de Línea y Campos Vectoriales (10 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Campos vectoriales: definición, visualización, divergencia y rotacional ▪ Integrales de línea de funciones escalares: longitud de arco y masa ▪ Integrales de línea de campos vectoriales: trabajo y circulación ▪ Campos conservativos, funciones potencial e independencia de la trayectoria ▪ Teorema de Green y sus aplicaciones en el plano	▪ Calcular integrales de línea escalares y vectoriales sobre curvas orientadas [Usar] ▪ Determinar si un campo vectorial es conservativo y construir su función potencial [Usar] ▪ Aplicar el Teorema de Green para evaluar integrales de línea mediante integrales dobles [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

Unidad 6: Integrales de Superficie y Teoremas Fundamentales (12 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Superficies paramétricas, plano tangente y área de superficie ▪ Integrales de superficie de funciones escalares ▪ Integrales de flujo: integrales de superficie de campos vectoriales ▪ Teorema de Stokes: relación entre integrales de línea e integrales de superficie ▪ Teorema de la Divergencia (Gauss): relación entre flujo e integral de volumen	▪ Parametrizar superficies y evaluar integrales de superficie escalares y de flujo [Usar] ▪ Aplicar los Teoremas de Stokes y de la Divergencia para reducir integrales complejas [Evaluar] ▪ Relacionar propiedades diferenciales locales (rotacional, divergencia) con el comportamiento global mediante teoremas integrales [Evaluar]
Lecturas : [**Apostol1997**], [**MarsdenTromba2011**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

STA251. Estadística y Probabilidades (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	STA251. Estadística y Probabilidades
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BMA102. Cálculo Diferencial. (1er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Resúmenes Numéricos: Posición, Dispersión y Forma (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Medidas de posición: media, mediana, moda, media recortada y media ponderada ▪ Medidas de dispersión: varianza, desviación estándar, RIQ, MAD y rango ▪ Medidas de forma: asimetría y curtosis; cuantiles y percentiles ▪ Estadísticas de resumen robustas: resistencia a valores atípicos y distribuciones de colas pesadas	▪ Calcular e interpretar medidas de posición, dispersión y forma para un conjunto de datos univariado [Usar] ▪ Seleccionar estadísticas de resumen apropiadas según las propiedades distribucionales de los datos [Usar] ▪ Comparar estadísticas de resumen clásicas y robustas y justificar la elección para un conjunto de datos dado [Evaluar]
Lecturas : [**Devore2016**]	

Unidad 2: Espacios Muestrales y Eventos (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Experimentos aleatorios, resultados y espacios muestrales: finitos, contables e incontables ▪ Operaciones de teoría de conjuntos sobre eventos: unión, intersección, complemento y leyes de De Morgan ▪ Sigma-álgebras y espacios medibles como fundamento de la probabilidad rigurosa ▪ Conjuntos de Borel y la sigma-álgebra de Borel sobre la recta real	▪ Definir espacios muestrales, eventos y sigma-álgebras para un experimento aleatorio dado [Familiarizarse] ▪ Aplicar operaciones de teoría de conjuntos para calcular probabilidades de eventos compuestos [Usar] ▪ Construir sigma-álgebras apropiadas para espacios medibles que surgen en problemas estadísticos [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 3: Axiomas y Propiedades de la Probabilidad (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Los tres axiomas de probabilidad de Kolmogorov y sus consecuencias ▪ Propiedades derivadas: regla del complemento, regla de la adición, monotonicidad y continuidad de la probabilidad ▪ Interpretaciones clásica, frecuentista y subjetiva (bayesiana) de la probabilidad ▪ Métodos de conteo: permutaciones, combinaciones y el principio de multiplicación	▪ Enunciar los axiomas de Kolmogorov y derivar propiedades fundamentales de la probabilidad [Familiarizarse] ▪ Calcular probabilidades de eventos utilizando la regla de la adición y la regla del complemento [Usar] ▪ Aplicar métodos de conteo para calcular probabilidades en espacios muestrales finitos [Usar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 4: Probabilidad Condicional e Independencia (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición e interpretación de la probabilidad condicional P(A|B) y sus propiedades ▪ Teorema de Bayes y ley de la probabilidad total: derivación y aplicaciones ▪ Independencia de eventos: independencia por pares e independencia mutua ▪ Independencia condicional y su papel en los modelos gráficos probabilísticos	▪ Explicar la probabilidad condicional y la independencia y su relación [Familiarizarse] ▪ Aplicar el teorema de Bayes y la ley de la probabilidad total para resolver problemas de probabilidad [Usar] ▪ Analizar si los eventos son independientes o condicionalmente independientes en un modelo de probabilidad dado [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 5: Variables Aleatorias Discretas y Distribuciones (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Funciones de masa de probabilidad (FMP) y funciones de distribución acumulada (FDA) para variables aleatorias discretas ▪ Distribuciones discretas clave: Bernoulli, Binomial, Geométrica, Binomial Negativa, Poisson, Hipergeométrica ▪ Propiedades de las distribuciones discretas: media, varianza, asimetría y curtosis ▪ Aproximación de Poisson a la Binomial y argumentos de distribución límite	▪ Identificar y describir las principales distribuciones de probabilidad discretas y sus parámetros [Familiarizarse] ▪ Calcular probabilidades, medias y varianzas utilizando las FMP de distribuciones discretas estándar [Usar] ▪ Seleccionar una distribución discreta apropiada para modelar un fenómeno aleatorio dado y justificar la elección [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 6: Variables Aleatorias Continuas y Distribuciones (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Funciones de densidad de probabilidad (FDP) y FDA para variables aleatorias continuas ▪ Distribuciones continuas clave: Uniforme, Normal, Exponencial, Gamma, Beta, Chi-cuadrado, t, F ▪ La distribución Normal: propiedades, normal estándar y su papel central en la inferencia estadística ▪ La familia exponencial de distribuciones: forma canónica y estadísticos suficientes	▪ Describir las propiedades y parámetros de las principales distribuciones de probabilidad continuas [Familiarizarse] ▪ Calcular probabilidades y cuantiles para distribuciones continuas estándar [Usar] ▪ Identificar distribuciones pertenecientes a la familia exponencial y derivar sus estadísticos suficientes [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 7: Distribuciones Conjuntas, Marginales y Condicionales (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ FMP y FDP conjuntas para dos o más variables aleatorias ▪ Distribuciones marginales y condicionales derivadas de distribuciones conjuntas ▪ Covarianza, correlación y sus propiedades; la independencia implica covarianza cero (el recíproco no siempre es cierto) ▪ La distribución normal bivariada y sus distribuciones condicionales	▪ Derivar distribuciones marginales y condicionales a partir de una distribución conjunta dada [Usar] ▪ Calcular la covarianza y correlación entre dos variables aleatorias [Usar] ▪ Analizar la estructura condicional de la distribución normal bivariada [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 8: Teoremas Límite: LGN y TCL (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ley débil de los grandes números: convergencia en probabilidad y sus implicaciones para la estimación ▪ Teorema central del límite: enunciado, condiciones y papel en la justificación de aproximaciones normales ▪ Aplicaciones del TCL: aproximación normal a la binomial y distribuciones muestrales ▪ Ley fuerte de los grandes números: convergencia casi segura ▪ El método delta: distribución asintótica de funciones suaves de estimadores	▪ Enunciar la LGN y el TCL y explicar las condiciones bajo las cuales se cumplen [Familiarizarse] ▪ Aplicar el TCL para derivar aproximaciones normales y distribuciones muestrales [Usar] ▪ Aplicar el método delta para derivar distribuciones asintóticas de estimadores no lineales [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 9: Estimación Puntual (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propiedades deseables: insesgamiento, consistencia, eficiencia y suficiencia ▪ Estimación por método de momentos: derivación y propiedades asintóticas ▪ Estimación de máxima verosimilitud: derivación, propiedades y principio de invarianza ▪ Cota inferior de Cramér-Rao y el concepto de información de Fisher	▪ Describir las propiedades clave de los estimadores y explicar las concesiones entre ellas [Familiarizarse] ▪ Derivar estimadores por método de momentos y de máxima verosimilitud para distribuciones estándar [Usar] ▪ Evaluar la eficiencia de un estimador usando la cota inferior de Cramér-Rao [Evaluar]
Lecturas : [**Devore2016**]	

Unidad 10: Estimación por Intervalos e Intervalos de Confianza (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Construcción de intervalos de confianza mediante cantidades pivote y distribuciones exactas ▪ Interpretación correcta de los intervalos de confianza: probabilidad de cobertura y errores conceptuales comunes ▪ Intervalos de confianza para muestras grandes basados en normalidad asintótica ▪ Dualidad entre intervalos de confianza y pruebas de hipótesis	▪ Construir intervalos de confianza para medias, proporciones y varianzas usando métodos exactos y de gran muestra [Usar] ▪ Interpretar correctamente los intervalos de confianza e identificar interpretaciones erróneas comunes [Usar] ▪ Explicar la dualidad entre intervalos de confianza y pruebas de hipótesis y utilizarla para extraer inferencias [Evaluar]
Lecturas : [**Devore2016**]	

Unidad 11: Fundamentos de las Pruebas de Hipótesis (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Marco de Neyman-Pearson: hipótesis nula y alternativa, regiones de rechazo, errores tipo I y II ▪ Valores p: definición, interpretación correcta y usos incorrectos comunes; potencia y tamaño de muestra ▪ Pruebas paramétricas comunes: prueba z, prueba t, prueba F y prueba ji-cuadrada ▪ Lema de Neyman-Pearson y pruebas uniformemente más poderosas	▪ Explicar la lógica de las pruebas de hipótesis, los errores tipo I y II y el significado de un valor p [Familiarizarse] ▪ Aplicar pruebas paramétricas estándar e interpretar los resultados en contexto [Usar] ▪ Derivar la prueba más poderosa para una hipótesis simple usando el lema de Neyman-Pearson [Evaluar]
Lecturas : [**Devore2016**]	

Unidad 12: Inferencia Bayesiana (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Marco bayesiano: a priori, verosimilitud, a posteriori y el papel del teorema de Bayes ▪ Distribuciones a priori: informativas, no informativas y conjugadas ▪ Inferencia a posteriori: estimaciones puntuales, intervalos de credibilidad y distribuciones predictivas a posteriori ▪ Comparación de modelos bayesianos: factores de Bayes y probabilidades a posteriori de modelos	▪ Explicar el enfoque bayesiano de la inferencia y contrastarlo con el enfoque frecuentista [Familiarizarse] ▪ Derivar la distribución a posteriori para modelos conjugados estándar [Usar] ▪ Aplicar factores de Bayes para comparar modelos estadísticos e interpretar la evidencia [Evaluar]
Lecturas : [**Ross2014**], [**Devore2016**]	

Unidad 13: Regresión Lineal Simple (4 horas)	
Resultados esperados: AG-C07	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estimación por mínimos cuadrados ordinarios (MCO): derivación de coeficientes y sus propiedades ▪ Inferencia sobre coeficientes de regresión: pruebas t, intervalos de confianza y valores p ▪ Coeficiente de determinación R-cuadrado y descomposición de la varianza mediante análisis de varianza ▪ Teorema de Gauss-Markov: propiedad MELI del MCO bajo supuestos clásicos	▪ Ajustar un modelo de regresión lineal simple e interpretar los coeficientes estimados [Usar] ▪ Realizar pruebas de hipótesis y construir intervalos de confianza para parámetros de regresión [Usar] ▪ Verificar los supuestos de Gauss-Markov y evaluar las consecuencias cuando se violan [Evaluar]
Lecturas : [**Devore2016**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BEI201. Inglés III (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BEI201. Inglés III
2.2 Semestre	:	3er Semestre.
2.3 Créditos	:	2
2.4 Horas	:	4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BEI102. Inglés II. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Comunicación Técnica (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Repaso de Tiempos: Presente, Pasado y Futuro en documentación técnica. ▪ Oraciones interrogativas con Wh- questions complejas para el análisis de sistemas. ▪ Palabras con más de un significado en contextos técnicos (Polisemia). ▪ Expresiones para describir actividades profesionales e intereses de investigación.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 2: Comparando Arquitecturas (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Uso de "What's it like?": Descripción de características de sistemas e interfaces. ▪ Adjetivos comparativos y superlativos avanzados en análisis de benchmarks. ▪ Colocaciones (collocations) en computación: high-performance, low-level, fault-tolerant. ▪ Lectura: Informes de comparación de arquitecturas (RISC vs. CISC, SQL vs. NoSQL).	▪ El estudiante analiza y describe diferencias entre arquitecturas, marcos de trabajo y sistemas computacionales utilizando estructuras comparativas variadas con precisión técnica.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 3: Antes y Ahora en la Computación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Perfecto vs. Pasado Simple: Contraste entre eventos históricos y estado actual. ▪ Expresiones temporales: for, since, ever, never aplicadas a la evolución tecnológica. ▪ Adverbios de tiempo y modo en la narrativa científica. ▪ Vocabulario: Generaciones de tecnología, sistemas heredados (legacy), plataformas actuales.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 4: Estándares y Protocolos de Ingeniería (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modales de obligación y permiso: Have to, Must, Can en normas profesionales. ▪ Recomendaciones técnicas: Should, Ought to — revisiones de código y diseño. ▪ Vocabulario: Estándares IEEE/ACM, convenciones de codificación, políticas de seguridad. ▪ Redacción formal: Memorandos técnicos y recomendaciones de ingeniería.	▪ El estudiante expresa normas de ingeniería, estándares y recomendaciones profesionales utilizando verbos modales apropiados en escritura técnica formal y discusión entre pares.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 5: ¿Cómo se construye el Software? (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Voz Pasiva: Presente y Pasado Simple para describir procesos de ingeniería. ▪ Vocabulario: Fabricación, despliegue, integración y pruebas. ▪ Expresiones para dar opiniones técnicas y discutir trade-offs de diseño. ▪ Redacción: Descripción de proceso para un sistema de software o hardware.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 6: Depuración y Análisis (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pasado Continuo vs. Pasado Simple en reportes de incidentes y fallos. ▪ Conectores de narrativa y secuencia: while, when, suddenly, as a result. ▪ Vocabulario: Bugs, excepciones, crashes, logs, análisis de causa raíz. ▪ Redacción: Informe de incidente de un fallo de software o de red.	▪ El estudiante narra sesiones de depuración, fallos de sistemas e incidentes experimentales utilizando tiempos narrativos y conectores de secuencia técnicos con fluidez.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 7: Investigación e Innovación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formas de Futuro: Will, Going to, Presente Continuo para proyección de investigación. ▪ Expresiones de probabilidad y certeza en contextos científicos. ▪ Vocabulario: Ciclos de innovación, propuestas de investigación, prospectiva tecnológica. ▪ Presentaciones orales: .El futuro de la computación en mi campo”.	▪ El estudiante proyecta metas de investigación y discute direcciones futuras en computación e ingeniería utilizando diversas estructuras de futuro con grados apropiados de certeza.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 8: Inglés Técnico III (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Terminología de Programación y Desarrollo Web: funciones, APIs, frameworks. ▪ Descripciones de Interfaz de Usuario (UI/UX) y conceptos de usabilidad. ▪ Vocabulario básico de gestión de proyectos: sprints, milestones, deliverables. ▪ Redacción: Especificación técnica simple o documento README.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS210. Algoritmos y Estructuras de Datos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS210. Algoritmos y Estructuras de Datos
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS113. Programación Orientada a Objetos II. (3er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos avanzados para estructuras de datos (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Revisión de metaprogramación con plantillas y rasgos avanzados. ▪ Rasgos de tipos y SFINAE para compilación condicional. ▪ Plantillas variádicas y paquetes de parámetros. ▪ Reenvío perfecto y referencias universales. ▪ Expresiones lambda con capturas genéricas. ▪ Diseño basado en políticas y CRTP (Patrón de Plantilla Curiosamente Recursivo). ▪ Restricciones basadas en conceptos (conceptos C++20 si son aplicables).	▪ Diseñar rasgos de tipos para personalizar el comportamiento de estructuras de datos [Usar]. ▪ Implementar plantillas variádicas para operaciones genéricas de contenedores [Usar]. ▪ Aplicar diseño basado en políticas para crear estructuras de datos configurables [Evaluar]. ▪ Usar reenvío perfecto para implementar constructores eficientes [Usar].
Lecturas : [**Stroustrup2013**], [**Vandevoorde2017**]	

Unidad 2: Estructuras de Datos Fundamentales (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: • Operaciones de diccionario (insertar, eliminar, encontrar) ▪ Arreglos: • Numéricos vs no numéricos, cadenas de caracteres • Unidimensionales (vector) vs multidimensionales (matriz) ▪ Registros/Estructuras/Tuplas y Objetos ▪ Listas enlazadas (por razones históricas): • Simples vs Dobles y Lineales vs Circulares ▪ Pilas ▪ Colas y deques: • Cola de prioridad basada en montículo ▪ Tablas/mapas hash: • Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) ▪ árboles: • Binarios, n-arios y árboles de búsqueda • Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) ▪ Conjuntos	▪ Para cada ADT/Estructura de Datos en esta unidad: • Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. • Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Usar] ▪ Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Usar] ▪ Implementar vector con política de asignación personalizable usando plantillas [Usar]. ▪ Diseñar nodos de lista enlazada genéricos con gestión de memoria basada en rasgos [Usar]. ▪ Comparar complejidad tiempo/espacio de diferentes estructuras lineales [Evaluar]. ▪ Aplicar plantillas variádicas para inicializar contenedores con múltiples elementos [Usar].
Lecturas : [**Cormen2009**], [**Stroustrup2013**]	

Unidad 3: Estructuras de Datos Fundamentales (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ árboles: • Binarios, n-arios y árboles de búsqueda • Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) ▪ Implementación de plantilla de árbol binario con soporte de iteradores. ▪ Árbol de búsqueda binaria con operadores de comparación basados en rasgos. ▪ Algoritmos de recorrido de árbol (iterativo y recursivo). ▪ Árboles de expresión y sus aplicaciones. ▪ Estructuras de nodo basadas en plantilla con carga útil configurable.	▪ 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] ▪ 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] ▪ Implementar árbol binario genérico con orden de recorrido configurable [Usar]. ▪ Diseñar evaluador de árbol de expresión usando patrón visitante [Usar]. ▪ Analizar complejidades de operaciones de árbol para diferentes estrategias de balanceo [Evaluar].
Lecturas : [**Cormen2009**], [**Knuth1997**]	

Unidad 4: Balanced and Multi-way Trees (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Implementación de árbol AVL con rasgo de balanceo basado en plantillas. ▪ Propiedades y operaciones de árbol rojo-negro. ▪ Implementación de plantilla de B-Tree para almacenamiento basado en disco. ▪ Estructuras de árbol 2-3 y árbol 2-3-4. ▪ Implementación de trie digital (árbol de prefijos). ▪ Árboles de sufijos y sus aplicaciones en procesamiento de cadenas. ▪ Estrategias de división/fusión de nodos para B-trees.	▪ Implementar árboles auto-balanceados usando diseño basado en políticas [Usar]. ▪ Diseñar B-tree para aplicaciones de índice de base de datos [Usar]. ▪ Evaluar compensaciones entre diferentes estructuras de árbol balanceado [Evaluar]. ▪ Implementar trie para coincidencia eficiente de prefijos de cadenas [Usar].
Lecturas : [**Cormen2009**], [**Knuth1997**], [**Sedgewick2011**]	

Unidad 5: Estructuras de Datos Fundamentales (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ árboles: • Binarios, n-arios y árboles de búsqueda • Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) ▪ Implementación de montículo binario usando representación de arreglo. ▪ Variantes de montículo mínimo y montículo máximo. ▪ Algoritmos heapify y análisis de complejidad. ▪ Implementación de ADT de cola de prioridad. ▪ Visión general de montículo binomial y montículo de Fibonacci. ▪ Aplicaciones de montículos: algoritmo de Dijkstra, ordenamiento por montículo.	▪ Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Familiarizarse] ▪ Implementar montículo genérico con rasgo de comparación configurable [Usar]. ▪ Diseñar cola de prioridad que soporte múltiples estrategias de actualización de prioridad [Usar]. ▪ Analizar complejidad de operaciones de montículo para diferentes tipos de montículo [Evaluar]. ▪ Aplicar montículo en implementaciones de algoritmos de grafos [Usar].
Lecturas : [**Cormen2009**], [**Sedgewick2011**]	

Unidad 6: Estructuras de Datos Fundamentales (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tablas/mapas hash: • Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) ▪ Diseño de función hash y propiedades. ▪ Implementación de encadenamiento separado con listas enlazadas. ▪ Direccionamiento abierto: sondeo lineal, sondeo cuadrático, doble hashing. ▪ Hashing perfecto y hashing perfecto mínimo. ▪ Hashing cuco y hashing de rayuela. ▪ Estrategias de redimensionamiento de tabla hash y gestión de factor de carga.	▪ Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Familiarizarse] ▪ Diseñar tabla hash genérica con resolución de colisiones configurable [Usar]. ▪ Implementar funciones hash personalizadas para tipos definidos por el usuario [Usar]. ▪ Evaluar rendimiento de diferentes esquemas de hashing [Evaluar]. ▪ Aplicar tablas hash en la implementación de tabla de símbolos del compilador [Usar].
Lecturas : [**Cormen2009**], [**Knuth1997**]	

Unidad 7: Concurrent Data Structures (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de estructuras de datos seguras para hilos. ▪ Listas enlazadas concurrentes basadas en bloqueos. ▪ Algoritmos sin bloqueo y libres de espera. ▪ Tablas hash concurrentes con bloqueo de grano fino. ▪ Operaciones atómicas y ordenamiento de memoria. ▪ Colas concurrentes (limitadas e ilimitadas). ▪ Pilas concurrentes y técnicas de eliminación.	▪ Diseñar estructuras de datos seguras para hilos usando mutex y guardas de bloqueo [Usar]. ▪ Implementar lista enlazada sin bloqueo usando operaciones atómicas [Usar]. ▪ Analizar rendimiento de estructuras de datos concurrentes vs secuenciales [Evaluar]. ▪ Aplicar colas concurrentes en patrones productor-consumidor [Usar].
Lecturas : [**Herlihy2012**], [**Williams2019**]	

Unidad 8: Specialized Structures and Applications (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructuras de datos persistentes. ▪ Unión-búsqueda de conjuntos disjuntos (union-find) con compresión de trayectoria. ▪ Filtros de Bloom y sus propiedades probabilísticas. ▪ Listas por niveles para búsqueda balanceada probabilística. ▪ Árboles de segmentos y árboles de Fenwick para consultas de rango. ▪ Estructuras de datos espaciales: árboles k-d, quadtrees. ▪ Estructuras de datos inconscientes de la caché.	▪ Implementar estructuras de datos persistentes con compartición estructural [Usar]. ▪ Diseñar union-find con unión por rango y compresión de trayectoria [Usar]. ▪ Aplicar estructuras especializadas para resolver problemas específicos de dominio [Evaluar]. ▪ Evaluar compensaciones espacio-tiempo en estructuras de datos probabilísticas [Evaluar].
Lecturas : [**Okasaki1999**], [**Sedgewick2011**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán

actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS211. Teoría de la Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS211. Teoría de la Computación
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS1D1. Estructuras Discretas. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Lenguajes Formales y Autómatas (18 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Autómatas formales: • Estado Finito • Pila • Linealmente Acotado • Máquina de Turing ▪ Lenguajes formales, gramáticas y Jerarquía de Chomsky: • Regulares (Tipo-3): ◦ Expresiones Regulares • Libres de Contexto (Tipo-2) • Sensibles al Contexto (Tipo-1) • Recursivamente Enumerables (Tipo-0) ▪ Autómatas deterministas y no deterministas ▪ Demostraciones del Lema de Bombeo: • Demostración de la limitación de Estado Finito/Lenguaje Regular • Limitación de Autómatas de Pila/Lenguaje Libre de Contexto	▪ Para cada autómata formal en esta unidad: • Explicar su definición comparando sus características con los otros autómatas de esta unidad • Usando un ejemplo, explicar paso a paso cómo el autómata opera sobre la entrada incluyendo si acepta la entrada asociada • Explicar un ejemplo de entradas que pueden y no pueden ser aceptadas por el autómata. [Explicar] ▪ Dado un problema, desarrollar un autómata apropiado que aborde el problema [Crear] ▪ Desarrollar una expresión regular para un lenguaje regular dado expresado en lenguaje natural [Crear] ▪ Convertir entre notaciones equivalentemente poderosas para un lenguaje, incluyendo entre DFAs, NFAs y expresiones regulares, y entre PDAs y CFGs [Aplicar] ▪ Aplicar lemas de bombeo, o medios alternativos, para demostrar las limitaciones de los autómatas de Estado Finito y Pila [Aplicar]
Lecturas : [**Sipser2012**], [**Hopcroft2013**]	

Unidad 2: Lenguajes Formales y Autómatas (16 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Autómatas formales: • Estado Finito • Pila • Linealmente Acotado • Máquina de Turing ▪ Lenguajes formales, gramáticas y Jerarquía de Chomsky: • Regulares (Tipo-3): ◦ Expresiones Regulares • Libres de Contexto (Tipo-2) • Sensibles al Contexto (Tipo-1) • Recursivamente Enumerables (Tipo-0) ▪ Formas Normales: Chomsky y Greibach ▪ Demostraciones del Lema de Bombeo: • Demostración de la limitación de Estado Finito/Lenguaje Regular • Limitación de Autómatas de Pila/Lenguaje Libre de Contexto ▪ Relaciones entre autómatas formales, lenguajes y gramáticas	▪ Para cada autómata formal en esta unidad: • Explicar su definición comparando sus características con los otros autómatas de esta unidad • Usando un ejemplo, explicar paso a paso cómo el autómata opera sobre la entrada incluyendo si acepta la entrada asociada • Explicar un ejemplo de entradas que pueden y no pueden ser aceptadas por el autómata. [Explicar] ▪ Dado un problema, desarrollar un autómata apropiado que aborde el problema [Crear] ▪ Convertir entre notaciones equivalentemente poderosas para un lenguaje, incluyendo entre DFAs, NFAs y expresiones regulares, y entre PDAs y CFGs [Aplicar] ▪ Aplicar lemas de bombeo, o medios alternativos, para demostrar las limitaciones de los autómatas de Estado Finito y Pila [Aplicar] ▪ Construir GLCs para descripciones de lenguajes formales [Evaluar]
Lecturas : [**Sipser2012**], [**Hopcroft2013**]	

Unidad 3: Lenguajes Formales y Autómatas (16 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Autómatas formales: • Estado Finito • Pila • Linealmente Acotado • Máquina de Turing ▪ Lenguajes formales, gramáticas y Jerarquía de Chomsky: • Regulares (Tipo-3): ◦ Expresiones Regulares • Libres de Contexto (Tipo-2) • Sensibles al Contexto (Tipo-1) • Recursivamente Enumerables (Tipo-0) ▪ Decidabilidad, (in)computabilidad y parada ▪ La tesis de Church-Turing ▪ Reducibilidad y reducciones ▪ Modelos equivalentes de computación algorítmica: • Máquinas de Turing y Variaciones (por ejemplo, multicinta, no deterministas) • Cálculo Lambda • Funciones Mu-Recursivas	▪ Para cada autómata formal en esta unidad: • Explicar su definición comparando sus características con los otros autómatas de esta unidad • Usando un ejemplo, explicar paso a paso cómo el autómata opera sobre la entrada incluyendo si acepta la entrada asociada • Explicar un ejemplo de entradas que pueden y no pueden ser aceptadas por el autómata. [Explicar] ▪ Explicar una Máquina de Turing universal y su operación [Explicar] ▪ 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] ▪ Explicar la Tesis de Church-Turing y su importancia para la computación algorítmica [Explicar] ▪ Aplicar aritmetización y diagonalización para demostrar que el Problema de la Parada para Máquinas de Turing es Indecidible [Aplicar] ▪ Dado un lenguaje indecidible conocido, aplicar una reducción por mapeo o historia computacional para demostrar que otro lenguaje es indecidible [Aplicar]
Lecturas : [**Sipser2012**], [**Kozen2006**]	

Unidad 4: Lenguajes Formales y Autómatas (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Lenguajes formales, gramáticas y Jerarquía de Chomsky: • Regulares (Tipo-3): ◦ Expresiones Regulares • Libres de Contexto (Tipo-2) • Sensibles al Contexto (Tipo-1) • Recursivamente Enumerables (Tipo-0) ▪ Relaciones entre autómatas formales, lenguajes y gramáticas ▪ Modelos equivalentes de computación algorítmica: • Máquinas de Turing y Variaciones (por ejemplo, multicinta, no deterministas) • Cálculo Lambda • Funciones Mu-Recursivas ▪ Lenguajes Sensibles al Contexto (Tipo-1) y aplicaciones	▪ Para cada modelo formal en esta unidad: • Explicar su definición comparando sus características con las otras en esta unidad • Explicar ejemplos de entradas que son y no pueden ser aceptadas por el lenguaje/gramática. [Explicar] ▪ Explicar las funciones Recursivas Primitivas y Generales (cero, sucesor, selección, recursión primitiva, composición y Mu), su importancia e implementaciones en Máquina de Turing [Explicar] ▪ Explicar cómo se realiza la computación en el Cálculo Lambda (por ejemplo, conversión alfa y reducción beta) [Explicar] ▪ Explicar el teorema de Rice y su importancia [Explicar]
Lecturas : [**Sipser2012**], [**Kozen2006**]	

Unidad 5: Marco de Análisis de Complejidad (3 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Marco de Análisis de Complejidad: • Rendimiento de un algoritmo en el mejor caso, caso promedio y peor caso • Mediciones empíricas y relativas (Orden de Crecimiento) • Tamaño de entrada y operaciones primitivas • Eficiencia de tiempo y espacio	▪ 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] ▪ Evaluar informalmente la clase de complejidad fundamental de algoritmos simples [Evaluar]
Lecturas : [**Sipser2012**], [**Cormen2022**]	

Unidad 6: Notación Asintótica y Clases de Complejidad (4 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis de complejidad asintótica (cotas promedio y peor caso): • Notaciones formales Big-O, Big-Omega y Big-Theta • Clases de Complejidad Fundamentales y Ejemplos/Problemas Representativos: ◦ $O(1)$ Constante (por ejemplo, acceso a arreglo) ◦ $O(\log_2 n)$ Logarítmica (por ejemplo, búsqueda binaria) ◦ $O(n)$ Lineal (por ejemplo, búsqueda lineal) ◦ $O(n \log_2 n)$ Log Lineal (por ejemplo, mergesort) ◦ $O(n^2)$ Cuadrática (por ejemplo, ordenamiento por selección) ◦ $O(n^c)$ Polinomial (por ejemplo, $O(n^3)$ eliminación gaussiana) ◦ $O(2^n)$ Exponencial (por ejemplo, Mochila, Satisfactibilidad (SAT), Viajante de Comercio (TSP), todos los subconjuntos) ◦ $O(n!)$ Factorial (por ejemplo, circuito hamiltoniano, todas las permutaciones)	▪ Usando ejemplos, explicar cada una de las clases de complejidad fundamentales en esta unidad [Explicar] ▪ Explicar a una audiencia no técnica la importancia de los algoritmos tratables versus intratables usando una explicación intuitiva de la complejidad Big-O [Explicar]
Lecturas : [**Sipser2012**], [**Cormen2022**]	

Unidad 7: Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (7 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Notaciones Little-o, Little-Omega y Little Theta ▪ Análisis recursivo formal	▪ Usar relaciones de recurrencia para evaluar la complejidad de tiempo de algoritmos definidos recursivamente [Aplicar] ▪ Aplicar relaciones de recurrencia elementales usando una forma del Teorema Maestro [Aplicar]
Lecturas : [**Sipser2012**], [**Cormen2022**]	

Unidad 8: Teoría de la Complejidad Computacional (6 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tractabilidad e intractabilidad: • Clases de Complejidad P, NP y NP-Completo • Problemas NP-Completos (por ejemplo, SAT, Mochila, TSP) • Reducciones ▪ Modelos de complejidad basados en Máquinas de Turing: • Complejidad de tiempo: ◦ Clases P, NP, NP-C y EXP ◦ Teorema de Cook-Levin • Complejidad de espacio: ◦ NSpace y PSpace ◦ Teorema de Savitch	▪ Explicar la importancia de la NP-Completitud [Explicar] ▪ Explicar cómo NP-Duro es una cota inferior y NP es una cota superior para la NP-Completitud [Explicar] ▪ Explicar ejemplos de problemas NP-completos [Explicar] ▪ Explicar el Teorema de Cook-Levin y la NP-Completitud de SAT [Explicar] ▪ Explicar las clases P y NP [Explicar] ▪ Demostrar que un problema es NP-Completo reduciendo un problema NP-C clásico conocido a él (por ejemplo, 3SAT y Clique) [Crear] ▪ Explicar la clase P-espacio y su relación con la clase EXP [Explicar]
Lecturas : [**Sipser2012**], [**Cormen2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.

- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS221. Arquitectura de Computadores (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS221. Arquitectura de Computadores
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS1D1. Estructuras Discretas. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Organización de Máquina a Nivel de Ensamblador (20 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitectura de máquina von Neumann ▪ Unidad de control: captación, decodificación y ejecución de instrucciones ▪ Introducción a SIMD vs MIMD y la taxonomía de Flynn ▪ Organización de multiprocesadores/multinúcleo de memoria compartida ▪ Arquitectura del conjunto de instrucciones (ISA) (por ejemplo, x86, ARM y RISC-V): • Conjuntos de instrucciones de ancho fijo vs variable • Formatos de instrucción • Manipulación de datos, control, E/S • Modos de direccionamiento • Programación en lenguaje de máquina • Programación en lenguaje ensamblador ▪ Mecanismos de llamada y retorno de subrutinas ▪ E/S e interrupciones ▪ Segmentos de montículo, estáticos, pila y código	▪ Discutir cómo las unidades funcionales clásicas de von Neumann se implementan en sistemas embebidos, particularmente en memoria dentro y fuera del chip [Debatir] ▪ Describir cómo se ejecutan las instrucciones en una máquina von Neumann clásica, con extensiones para hilos, sincronización de multiprocesadores y ejecución SIMD [Describir] ▪ 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)] ▪ Discutir cómo se representan las instrucciones a nivel de máquina y en el contexto de un ensamblador simbólico [Debatir] ▪ Mapear un ejemplo de patrones de lenguaje de alto nivel a notaciones de lenguaje ensamblador/máquina [Mapear/Mapa conceptual] ▪ Contrastar diferentes formatos de instrucción considerando aspectos como direcciones por instrucción y formatos de longitud variable vs longitud fija [Contrastar] ▪ Analizar un diagrama de subrutina para comentar cómo se manejan las llamadas de subrutina a nivel de ensamblador [Analizar] ▪ Describir conceptos básicos de interrupciones y operaciones de E/S [Describir] ▪ Escribir un programa simple en lenguaje ensamblador para procesamiento y manipulación de cadenas/arreglos [Escribir]
Lecturas : [**Patterson2020**], [**Harris2012**]	

Unidad 2: Jerarquía de Memoria (18 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Jerarquía de memoria: la importancia de la localidad temporal y espacial ▪ Organización y operaciones de memoria principal ▪ Memoria persistente (por ejemplo, SSD, discos estándar) ▪ Latencia, tiempo de ciclo, ancho de banda e intercalado ▪ Memorias caché: • Mapeo de direcciones • Tamaño de bloque • Política de reemplazo y almacenamiento • Prebúsqueda ▪ Coherencia de caché multiprocesador ▪ Memoria virtual (soporte de hardware) ▪ Manejo de fallas y confiabilidad ▪ Confiabilidad: • Codificación de error • Compresión de datos • Integridad de datos ▪ Procesamiento en Memoria (PIM)	▪ 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] ▪ Medir el efecto de la latencia de memoria en el tiempo de ejecución [Evaluar] ▪ Enumerar las funciones de un sistema con gestión de memoria virtual [Enumerar] ▪ 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]
Lecturas : [**Patterson2020**], [**Stallings2015**]	

Unidad 3: Interconexión y Comunicación (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Fundamentos de E/S: • Protocolo de enlace y almacenamiento en búfer • E/S programada • E/S basada en interrupciones ▪ Estructuras de interrupción: vectorizadas y priorizadas, acuse de recibo de interrupción ▪ Dispositivos de E/S (por ejemplo, ratón, teclado, pantalla, cámara, sensores, actuadores) ▪ Almacenamiento externo, organización física y unidades ▪ Fundamentos de buses: • Protocolos de bus • Arbitraje • Acceso directo a memoria (DMA)	▪ 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] ▪ Enumerar varios tipos de buses en un sistema informático [Enumerar] ▪ Enumerar las ventajas de los discos magnéticos y contrastarlas con las de los discos de estado sólido [Listar/Enumerar]
Lecturas : [**Patterson2020**], [**Tanenbaum2012**]	

Unidad 4: Organización Funcional (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Implementación de rutas de datos simples, incluyendo segmentación de instrucciones, detección de riesgos y resolución (por ejemplo, paradas, reenvío) ▪ Unidad de control: • Implementación cableada • Realización microprogramada ▪ Segmentación de instrucciones ▪ Introducción al paralelismo a nivel de instrucción (ILP)	▪ Comparar implementaciones alternativas de rutas de datos en arquitecturas de computadoras modernas [Comparar] ▪ Producir un conjunto de señales de control para sumar dos enteros usando implementaciones cableadas y microprogramadas [Producir] ▪ Discutir el paralelismo a nivel de instrucción usando segmentación y los riesgos significativos que pueden ocurrir [Debatir] ▪ Diseñar un procesador completo, incluyendo ruta de datos y control [Diseñar] ▪ Calcular los ciclos promedio por instrucción para una implementación dada de procesador y sistema de memoria [Computar/Calcular]
Lecturas : [**Harris2012**], [**Patterson2020**]	

Unidad 5: Arquitecturas Heterogéneas (10 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitecturas SIMD y MIMD (por ejemplo, GPUs de Propósito General, TPUs y NPUs) ▪ Sistemas de memoria heterogéneos: • Memoria compartida versus memoria distribuida • Memoria volátil vs no volátil • Protocolos de coherencia ▪ Arquitecturas Específicas de Dominio (DSAs): • Acelerador de Aprendizaje Automático • Computación en red • Sistemas embebidos para aplicaciones emergentes • Computación neuromórfica • Dispositivos de computación perimetral ▪ Soluciones de empaquetado e integración como 3DIC y chiplets ▪ Aprendizaje automático en diseño de arquitectura: • Algoritmos de IA para análisis de carga de trabajo • Optimización de configuraciones de arquitectura para rendimiento y eficiencia energética ▪ 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.	▪ Analizar un diagrama de sistema con arquitecturas paralelas alternativas, por ejemplo, SIMD y MIMD, e identificar las diferencias clave [Evaluar] ▪ Discutir qué problemas de gestión de memoria se encuentran en multiprocesadores que no están presentes en uniprocesadores y cómo estos problemas podrían resolverse [Evaluar] ▪ 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] ▪ Discutir cómo determinarías cuándo usar un acelerador específico de dominio en lugar de una CPU de propósito general [Evaluar] ▪ Enumerar diferencias clave en los principios de diseño arquitectónico entre una unidad de procesamiento basada en vector y basada en escalar [Familiarizarse] ▪ Enumerar las ventajas y desventajas de una arquitectura PIM [Familiarizarse] ▪ Reestructurar una colección de objetos (AoS) a un diseño DOD (SoA) y medir el impacto en rendimiento [Usar].
Lecturas : [**Patterson2020**], [**Stallings2015**]	

Unidad 6: Arquitecturas de Procesador Seguras (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de Hardware Seguro: • Análisis de Riesgos de Seguridad, Protección de Activos y Modelo de Amenazas • Aceleración Criptográfica con Hardware • Soporte para virtualización (por ejemplo, aislamiento de SO) ▪ Raíces de confianza en hardware, Funciones Físicamente No Clonables (PUF) ▪ Generadores de Números Aleatorios por Hardware ▪ Extensiones de protección de memoria: • Verificación de límites de puntero en tiempo de ejecución (por ejemplo, desbordamiento de búfer) • Protección a nivel microarquitectónico • Protección a nivel de ISA ▪ Entorno de Ejecución Confiable (TEE): • Protecciones de Base de Computadora Confiable • Protección de máquinas virtuales • Protección de contenedores • Módulos de software confiables (Enclaves) ▪ Cifrado homomórfico para procesamiento de datos que preserva la privacidad	▪ Discutir principios de hardware seguro, explorando un marco para análisis de riesgos y protección de activos [Debatir] ▪ Resumir cómo las Funciones Físicamente No Clonables (PUF) pueden ser un identificador único de dispositivo en aplicaciones de seguridad [Resumir] ▪ Distinguir un generador de números aleatorios con soporte de hardware dedicado de generadores sin hardware dedicado a generar entropía [Distinguir] ▪ Enumerar las ventajas y desventajas de la protección de memoria a nivel de ISA [Listar/Enumerar] ▪ Describir problemas de diseño clave de un entorno de ejecución confiable (TEE) para soportar máquinas virtuales [Describir]
Lecturas : [**Patterson2020**], [**Anderson2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS271. Bases de Datos I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS271. Bases de Datos I
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	▪ CS112. Programación Orientada a Objetos I. (2do Sem) ▪ CS1D1. Estructuras Discretas. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Conceptos Fundamentales de Sistemas de Bases de Datos (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propósito y ventajas de los sistemas de bases de datos ▪ Componentes de los sistemas de bases de datos ▪ Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) ▪ Arquitectura de base de datos, independencia de datos y abstracción de datos ▪ Gestión de transacciones ▪ Normalización ▪ Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) ▪ Cómo soportar aplicaciones solo de CRUD ▪ Bases de datos distribuidas/sistemas basados en la nube ▪ Datos estructurados, semiestructurados y no estructurados ▪ Uso de un lenguaje de consulta declarativo ▪ Sistemas que soportan contenido estructurado y/o de flujo	▪ Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Familiarizarse] ▪ Enumerar los componentes de un sistema de base de datos (relacional) [Familiarizarse] ▪ Explicar las ventajas de los sistemas de bases de datos sobre los sistemas de archivos [Usar]. ▪ Describir la arquitectura de tres esquemas y la independencia de datos [Familiarizarse]. ▪ Identificar diferentes tipos de usuarios de bases de datos y sus roles [Familiarizarse].
Lecturas : [**Ramakrishnan2002**], [**Silberschatz2019**]	

Unidad 2: Modelado de Datos (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos conceptuales (por ejemplo, entidad-relación, diagramas UML) ▪ Conjuntos de entidades, conjuntos de relaciones y atributos. ▪ Restricciones de cardinalidad: uno-a-uno, uno-a-muchos, muchos-a-muchos. ▪ Conjuntos de entidades débiles y relaciones identificadoras. ▪ Características extendidas E-R: especialización, generalización, agregación. ▪ Diagramas de clases UML para modelado de datos.	▪ Describir los componentes del modelo E-R (o algún otro modelo no relacional) [Familiarizarse] ▪ Modelar un entorno dado usando un modelo de datos conceptual [Usar] ▪ Diseñar diagramas E-R para escenarios del mundo real [Usar]. ▪ Convertir diagramas E-R a esquemas relacionales [Usar]. ▪ Aplicar características E-R extendidas para modelado complejo [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unidad 3: Bases de Datos Relacionales (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integridad de entidad y referencial: Clave candidata, superclaves ▪ Diseño de base de datos relacional ▪ Operaciones de álgebra relacional: selección, proyección, join, unión, etc. ▪ Restricciones de integridad: dominio, clave, entidad, referencial. ▪ Cálculo relacional: cálculo de tuplas y de dominio. ▪ Mapeo de diagramas E-R a esquemas relacionales.	▪ Describir las características definitorias detrás del modelo de datos relacional [Familiarizarse] ▪ Comentar sobre la diferencia entre una clave foránea y una superclave [Usar] ▪ Expresar consultas usando álgebra relacional [Usar]. ▪ Diseñar esquemas relacionales con restricciones apropiadas [Usar]. ▪ Mapear diagramas E-R complejos a relaciones normalizadas [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unidad 4: Construcción de Consultas (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formación de Consultas SQL: • Ejecución interactiva de SQL • Ejecución programática de una consulta SQL ▪ Lenguaje de definición de datos: CREATE, ALTER, DROP. ▪ Tipos de datos y restricciones de dominio. ▪ Creación y gestión de índices. ▪ Definición de vistas y vistas materializadas. ▪ Diseño e implementación de esquemas de base de datos.	▪ Componer consultas SQL que incorporen selección, proyección, unión, intersección, diferencia de conjuntos y división de conjuntos [Usar] ▪ Definir, en SQL, un esquema de relación, incluyendo todas las restricciones de integridad y activadores de eliminación/actualización [Usar] ▪ Crear y modificar esquemas de base de datos usando SQL DDL [Usar]. ▪ Implementar restricciones de integridad a nivel de esquema [Usar]. ▪ Diseñar y crear índices apropiados para rendimiento [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Beaulieu2009**]	

Unidad 5: Construcción de Consultas (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Consultas SQL complejas: subconsultas anidadas, subconsultas correlacionadas. ▪ Funciones de agregación y operaciones GROUP BY. ▪ Operaciones de conjunto: UNION, INTERSECT, EXCEPT. ▪ Operaciones JOIN: inner, outer, natural, cross joins. ▪ Manipulación de datos: INSERT, UPDATE, DELETE. ▪ Control de transacciones: COMMIT, ROLLBACK, SAVEPOINT.	▪ Determinar cuándo una consulta SQL anidada está correlacionada o no [Usar] ▪ Componer una consulta SQL para actualizar una tupla en una relación [Usar] ▪ Formular consultas complejas con múltiples joins y subconsultas [Usar]. ▪ Aplicar agregación para análisis de datos y generación de informes [Usar]. ▪ Implementar operaciones de modificación de datos con control de transacciones [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Beaulieu2009**]	

Unidad 6: Bases de Datos Relacionales (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción a la teoría de dependencias funcionales ▪ Teoría de Normalización: • Descomposición de un esquema; propiedades de unión sin pérdida y preservación de dependencias de una descomposición • Formas normales (BCNF) • Desnormalización (para eficiencia) ▪ Dependencias funcionales y reglas de inferencia. ▪ Formas normales: 1FN, 2FN, 3FN, FNBC. ▪ Algoritmos de descomposición: sin pérdida de join y que preserva dependencias. ▪ Consideraciones prácticas en normalización.	▪ Describir cómo la teoría de dependencias funcionales generaliza la noción de clave [Familiarizarse] ▪ Detectar qué forma normal produce una descomposición dada [Usar] ▪ Identificar dependencias funcionales en esquemas relacionales [Usar]. ▪ Normalizar relaciones a FNBC [Evaluar]. ▪ Evaluar descomposiciones para sin pérdida de join y preservación de dependencias [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unidad 7: Internos de DBMS (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de Transacciones: • Niveles de Aislamiento • ACID • Serializabilidad • Transacciones Distribuidas ▪ Control de Concurrencia: • Bloqueo en 2 Fases • Estrategias de manejo de bloqueos mutuos • Modelos de consistencia basados en quórum ▪ Propiedades ACID: Atomicidad, Consistencia, Aislamiento, Durabilidad. ▪ Serializabilidad y serializabilidad por conflicto. ▪ Control de concurrencia basado en bloqueos: bloqueo en dos fases. ▪ Detección y prevención de interbloqueos. ▪ Niveles de aislamiento y sus compensaciones.	▪ Describir las cuatro propiedades para un gestor de transacciones correcto [Familiarizarse] ▪ Esbozar el principio de serializabilidad [Familiarizarse] ▪ Explicar las propiedades ACID y su importancia [Usar]. ▪ Analizar programaciones de transacciones para serializabilidad [Evaluar]. ▪ Aplicar protocolos de bloqueo para control de concurrencia [Usar].
Lecturas : [**Ramakrishnan2002**], [**Bernstein1987**]	

Unidad 8: Seguridad y Privacidad de Datos (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diferencias entre seguridad de datos y privacidad de datos ▪ Información de identificación personal (PII) y su protección ▪ Consideraciones éticas para asegurar la seguridad y privacidad de los datos ▪ Seguridad de bases de datos: autenticación, autorización, encriptación. ▪ Ataques de inyección SQL y prevención. ▪ Introducción a bases de datos NoSQL y big data. ▪ Gobernanza de datos y cumplimiento normativo.	▪ Describir las diferencias en los objetivos para seguridad de datos y privacidad de datos [Familiarizarse] ▪ Identificar y mitigar riesgos asociados con diferentes enfoques para proteger datos [Usar] ▪ Implementar medidas básicas de seguridad de bases de datos [Usar]. ▪ Prevenir vulnerabilidades de inyección SQL [Evaluar]. ▪ Discutir consideraciones éticas en la recolección y uso de datos [Evaluar].
Lecturas : [**Ramakrishnan2002**], [**Elmasri2015**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS401. Metodología de la Investigación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS401. Metodología de la Investigación
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	2
2.4 Horas	:	1 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS100. Introducción a la Ciencia de la Computación. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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).

6. RESULTADOS DEL ESTUDIANTE

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

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

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

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

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

7. TEMAS

Unidad 1: Fundamentos de Investigación Científica en Computación (8 horas)	
Resultados esperados: AG-C05,AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipos de investigación en computación: empírica, teórica, constructiva y evaluativa. ▪ El método científico aplicado a la investigación en computación. ▪ Principales bases de datos académicas: IEEE Xplore, ACM Digital Library, Google Scholar, Semantic Scholar y arXiv. ▪ Estrategias de búsqueda bibliográfica: operadores booleanos, seguimiento de citas y búsqueda sistemática. ▪ Gestores de referencias: Zotero, Mendeley e integración con BibTeX.	▪ Identificar los tipos de investigación en computación y sus diferencias metodológicas. [Familiarizarse] ▪ Utilizar las principales bases de datos académicas para encontrar literatura relevante en un tópico dado. [Usar] ▪ Organizar referencias bibliográficas usando gestores de referencias y formato BibTeX. [Usar]
Lecturas : [**ieee**], [**acm**]	

Unidad 2: Lectura Crítica y Análisis de Literatura Científica (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructura de un artículo científico: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. ▪ Técnicas para la lectura eficiente de artículos científicos (lectura en tres pasadas). ▪ Evaluación crítica: identificación de contribuciones, limitaciones y problemas abiertos. ▪ Construcción de una matriz de lectura y tablas comparativas de trabajos relacionados. ▪ Clasificación y diseño de taxonomías de enfoques existentes sobre un tópico.	▪ Leer y evaluar críticamente artículos científicos identificando sus contribuciones y limitaciones. [Usar] ▪ Construir una matriz de lectura organizada que sintetice múltiples trabajos sobre un tópico dado. [Usar] ▪ Diseñar una taxonomía preliminar de enfoques para un área de investigación asignada. [Evaluar]
Lecturas : [**ieee**], [**acm**], [**citeseer**]	

Unidad 3: Ética en Investigación e Integridad Académica (6 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integridad académica: plagio, autoplagio, fabricación y falsificación de datos. ▪ Ética en investigación en computación: privacidad, sesgo algorítmico y estudios con sujetos humanos. ▪ Acceso abierto, licencias (Creative Commons, GPL) y propiedad intelectual en software. ▪ Buenas prácticas de citación y atribución de autoría en trabajos colaborativos. ▪ El proceso de revisión por pares: rol del revisor y responsabilidades del autor.	▪ Distinguir las formas de mala conducta académica y aplicar prácticas de citación correctas. [Usar] ▪ Analizar dilemas éticos en investigación computacional aplicando principios de integridad. [Evaluar] ▪ Explicar los derechos de propiedad intelectual y las licencias relevantes en computación. [Familiarizarse]
Lecturas : [**ieee**], [**acm**]	

Unidad 4: Redacción Técnica y Herramientas de Escritura Científica (10 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructura de LaTeX: preámbulo, secciones, ecuaciones, figuras, tablas y referencias cruzadas. ▪ Plantillas de conferencias y revistas: IEEE y ACM. ▪ Estilo de escritura académica en computación: claridad, precisión y voz activa. ▪ Redacción de cada sección del artículo: resumen, introducción, trabajos relacionados, metodología, resultados y conclusiones. ▪ Integración Continua de documentos: control de versiones de documentos LaTeXcon Git. ▪ Gestión de bibliografía con BibTeX y JabRef.	▪ Redactar un artículo técnico completo en LaTeXusando plantillas IEEE o ACM. [Usar] ▪ Aplicar las convenciones de escritura académica propias de la comunidad de computación. [Usar] ▪ Evaluar la calidad de redacción de un artículo propio o ajeno según criterios de claridad y estructura. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 5: Revisión Sistemática de Literatura y Artículos de Survey (10 horas)	
Resultados esperados: AG-C04,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipos de revisiones: narrativa, sistemática, estudio de mapeo y meta-análisis. ▪ Protocolo de revisión sistemática de literatura (SLR): preguntas de investigación, estrategia de búsqueda y criterios de selección. ▪ Pautas PRISMA: diagrama de flujo de selección de estudios primarios. ▪ Técnicas de síntesis: narrativa, votación y análisis temático. ▪ Diseño de taxonomías y esquemas de clasificación para un survey. ▪ Estructura y redacción de un artículo del tipo survey.	▪ Diseñar y ejecutar un protocolo de revisión sistemática de literatura para un tópico de computación. [Usar] ▪ Aplicar las pautas PRISMA para documentar y reportar el proceso de selección de estudios. [Usar] ▪ Redactar un artículo de tipo survey con taxonomía, síntesis crítica y comparación de enfoques. [Evaluar]
Lecturas : [**ieee**], [**acm**], [**citeseer**]	

Unidad 6: Presentación y Difusión de Resultados de Investigación (6 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Técnicas de presentación oral para audiencias científicas: estructura, manejo del tiempo y sesión de preguntas. ▪ Diseño de pósters para conferencias y simposios de investigación. ▪ Respuesta a revisores: cómo abordar comentarios de forma constructiva y argumentada. ▪ Planificación de un proyecto de investigación: cronograma, hitos y entregables. ▪ Prácticas de colaboración y coautoría en investigación científica.	▪ Preparar y exponer una presentación oral de resultados de investigación ante una audiencia académica. [Usar] ▪ Elaborar un plan de trabajo para un proyecto de investigación con cronograma e hitos. [Usar] ▪ Responder constructivamente a comentarios de revisores en el proceso de publicación. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

MA202. Métodos Numéricos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	MA202. Métodos Numéricos
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BMA103. Cálculo Integral. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

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

7. TEMAS

Unidad 1: Análisis de Errores y Aritmética de Punto Flotante (6 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Representación en punto flotante IEEE 754: épsilon de máquina, desbordamiento, subdesbordamiento y modos de redondeo ▪ Errores de redondeo, cancelación catastrófica y pérdida de significado ▪ Análisis de error directo e inverso; número de condición de un problema ▪ Estabilidad numérica: algoritmos estables vs. inestables; análisis de error inverso de Wilkinson ▪ Propagación de errores en operaciones aritméticas y evaluación de funciones	▪ Explicar el estándar IEEE 754 de punto flotante e identificar las fuentes de errores de redondeo y cancelación [Familiarizarse] ▪ Estimar el número de condición de un problema y predecir la precisión de un resultado numérico [Usar] ▪ Analizar un algoritmo para determinar su estabilidad numérica usando cotas de error directo e inverso [Evaluar]
Lecturas : [**ChapraCanale2015**], [**BurdenFaires2010**]	

Unidad 2: Búsqueda de Raíces de Ecuaciones No Lineales (10 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Método de bisección: acotamiento del intervalo, criterio de parada y cota del error ▪ Método de Newton-Raphson: derivación, convergencia cuadrática y casos de falla ▪ Método de la secante como alternativa al método de Newton sin derivada explícita ▪ Orden de convergencia y comparación de la eficiencia entre métodos iterativos ▪ Extensión del método de Newton a sistemas de ecuaciones no lineales mediante la matriz Jacobiana, base de solucionadores en aprendizaje automático	▪ Comparar las garantías de convergencia de los métodos de bisección, Newton-Raphson y secante [Familiarizarse] ▪ Implementar el método de Newton-Raphson para aproximar raíces de una función no lineal [Usar] ▪ Analizar el orden de convergencia de un método iterativo y justificar su elección según el costo computacional [Evaluar]
Lecturas : [**ChapraCanale2015**], [**BurdenFaires2010**]	

Unidad 3: Teoría de Aproximación e Interpolación (10 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Interpolación polinomial: Lagrange, diferencias divididas de Newton y fenómeno de Runge ▪ Interpolación por splines: splines cúbicos, B-splines y métodos de funciones polinómicas a trozos ▪ Mejor aproximación en espacios normados: Chebyshev (minimax) y mínimos cuadrados ▪ Polinomios de Chebyshev: propiedades, ortogonalidad y convergencia espectral ▪ Aproximación trigonométrica y transformada rápida de Fourier (FFT)	▪ Explicar el fenómeno de Runge y justificar la elección de nodos de Chebyshev para mitigarlo [Familiarizarse] ▪ Construir interpolantes por splines cúbicos y de Chebyshev para datos dados y estimar el error de interpolación [Usar] ▪ Aplicar la FFT para calcular eficientemente aproximaciones trigonométricas de una función muestreada [Evaluar]
Lecturas : [**ChapraCanale2015**], [**BurdenFaires2010**]	

Unidad 4: Integración Numérica y Cuadratura (10 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Reglas de Newton-Cotes: trapezoidal, Simpson y reglas compuestas; análisis de error ▪ Cuadratura adaptativa y control automático del error ▪ Cuadratura gaussiana: nodos y pesos óptimos, conexión con polinomios ortogonales ▪ Tratamiento numérico de integrales impropias y singulares ▪ Integración de Monte Carlo y métodos cuasi-Monte Carlo para integrales de alta dimensión	▪ Derivar la fórmula de error para la regla compuesta de Simpson e identificar su orden de precisión [Familiarizarse] ▪ Seleccionar y aplicar una regla de cuadratura apropiada (Gaussiana, adaptativa) según la regularidad del integrando [Usar] ▪ Aplicar integración de Monte Carlo para estimar integrales de alta dimensión y cuantificar el error estadístico [Evaluar]
Lecturas : [**ChapraCanale2015**], [**BurdenFaires2010**]	

Unidad 5: Métodos Numéricos para Ecuaciones Diferenciales (12 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Métodos de Runge-Kutta: Euler, RK4 y métodos embebidos para control de errores ▪ Métodos de pasos múltiples: Adams-Bashforth, Adams-Moulton y métodos BDF ▪ Análisis de estabilidad: cero-estabilidad, regiones de estabilidad absoluta y EDOs rígidas ▪ Métodos de diferencias finitas para EDPs parabólicas y elípticas: estabilidad y convergencia	▪ Comparar solucionadores de EDOs explícitos e implícitos y explicar su estabilidad relativa para problemas rígidos [Familiarizarse] ▪ Implementar un método de Runge-Kutta con control adaptativo del paso y aplicarlo a un sistema de EDOs [Usar] ▪ Analizar la estabilidad y convergencia de un esquema de diferencias finitas para una EDP parabólica [Evaluar]
Lecturas : [**ChapraCanale2015**], [**BurdenFaires2010**]	

Unidad 6: Álgebra Lineal Numérica (12 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Eliminación gaussiana con pivoteo parcial, factorización LU y análisis de complejidad ▪ Factorización QR vía Householder y Gram-Schmidt; problemas de mínimos cuadrados ▪ Algoritmos de valores propios: iteración de la potencia, algoritmo QR y el método de Lanczos ▪ Descomposición en valores singulares: cálculo, truncamiento y aplicaciones (ACP, pseudoinversa) ▪ Métodos iterativos para sistemas dispersos grandes: GC, GMRES y precondicionamiento	▪ Comparar solucionadores directos e iterativos y seleccionar el método adecuado para una estructura matricial dada [Familiarizarse] ▪ Aplicar la factorización QR y la SVD para resolver problemas de mínimos cuadrados y calcular aproximaciones de bajo rango [Usar] ▪ Analizar la estabilidad numérica y el costo computacional de los algoritmos de valores propios [Evaluar]
Lecturas : [**BurdenFaires2010**], [**TrefethenBau1997**]	

Unidad 7: Método de Elementos Finitos (8 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formulación débil (variacional) de problemas de valor en la frontera y espacios de Sobolev ▪ Método de Galerkin: espacios de aproximación de dimensión finita y matriz de rigidez ▪ Espacios de elementos finitos triangulares y cuadriláteros: elementos de Lagrange y conformidad ▪ Estimaciones de error a priori: lema de Céa y cotas de error de interpolación ▪ Estimaciones de error a posteriori y refinamiento adaptativo de malla	▪ Derivar la formulación débil de un PVE elíptico y mostrar su equivalencia con la forma fuerte [Familiarizarse] ▪ Ensamblar la matriz de rigidez global y el vector de carga para una discretización lineal por elementos finitos [Usar] ▪ Estimar el error en H^1 de una solución por elementos finitos usando el lema de Céa y la teoría de interpolación [Evaluar]
Lecturas : [**ReddyFEM2019**], [**BurdenFaires2010**]	

Unidad 8: Algoritmos de Optimización (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descenso de gradiente y métodos de búsqueda lineal: condiciones de Armijo-Wolfe y tasas de convergencia ▪ Método de Newton y métodos cuasi-Newton (BFGS, L-BFGS) ▪ Optimización con restricciones: condiciones KKT, métodos de penalización y programación cuadrática secuencial ▪ Algoritmos de optimización convexa: métodos de punto interior y ADMM ▪ Descenso de gradiente estocástico (SGD), reducción de varianza y el optimizador Adam	▪ Explicar las garantías de convergencia del descenso de gradiente para funciones convexas suaves [Familiarizarse] ▪ Aplicar los métodos de Newton y cuasi-Newton a problemas de optimización sin restricciones [Usar] ▪ Formular un problema de optimización con restricciones como un sistema KKT y aplicar un método de punto interior [Evaluar]
Lecturas : [**NocedalWright2006**]	

Unidad 9: Computación Paralela y de Alto Rendimiento (4 horas)	
Resultados esperados: AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitecturas paralelas: memoria compartida (OpenMP), memoria distribuida (MPI) y GPU (CUDA) ▪ Modelos de rendimiento: ley de Amdahl, modelo de línea techada y complejidad de comunicación ▪ Algoritmos paralelos para álgebra lineal densa y dispersa (ScaLAPACK, PETSc) ▪ Descomposición de dominio y solucionadores paralelos de EDPs ▪ Diferenciación automática (modo directo y reverso) y su rol en ML científico	▪ Explicar la ley de Amdahl e identificar los cuellos de botella que limitan la aceleración paralela en un algoritmo dado [Familiarizarse] ▪ Implementar una rutina paralela de álgebra lineal usando MPI o OpenMP y medir la eficiencia paralela [Usar] ▪ Aplicar diferenciación automática para calcular gradientes de funciones científicas complejas con fines de optimización [Evaluar]
Lecturas : [**PachecoMalensek2022**]	

Unidad 10: Teoría de Grafos (4 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Grafos: definiciones, isomorfismo, sucesiones de grados, árboles y árboles de expansión ▪ Conectividad, Teorema de Menger y flujos en redes (flujo máximo-corte mínimo) ▪ Emparejamientos (Teorema de Hall), coloraciones de grafos y el polinomio cromático ▪ Grafos planos, fórmula de Euler, Teorema de Kuratowski y el teorema de los cuatro colores ▪ Matrices de adyacencia y laplaciana, valores propios y grafos expansores	▪ Identificar propiedades estructurales de grafos (conectividad, planaridad, bipartición) y aplicar la fórmula de Euler [Familiarizarse] ▪ Aplicar el Teorema de Hall y algoritmos de flujo en redes a problemas de emparejamiento y enrutamiento [Usar] ▪ Analizar el espectro de un grafo para acotar su número cromático y sus propiedades de conectividad [Evaluar]
Lecturas : [**West2001**]	

Unidad 11: Análisis de Series Temporales (2 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estacionariedad, función de autocovarianza y función de autocorrelación (ACF) ▪ Modelos ARMA: identificación, estimación (Yule-Walker, MLE) y diagnóstico ▪ Densidad espectral, el periodograma y el teorema de Wiener-Khinchin ▪ Modelos de espacio de estados y el filtro de Kalman ▪ Modelos ARIMA, ajuste estacional y predicción a múltiples pasos	▪ Identificar la estacionariedad y determinar el orden de un modelo ARMA a partir de gráficos ACF y PACF [Familiarizarse] ▪ Ajustar modelos ARIMA a datos de series temporales, validar residuos y producir predicciones [Usar] ▪ Aplicar el filtro de Kalman para estimar estados ocultos en un modelo lineal gaussiano de espacio de estados [Evaluar]
Lecturas : [**ShumwayStoffer2017**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BEI202. Inglés IV (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BEI202. Inglés IV
2.2 Semestre	:	4to Semestre.
2.3 Créditos	:	2
2.4 Horas	:	4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BEI201. Inglés III. (3er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Buenas Prácticas y Ética (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modales de obligación y consejo: Should, Must, Have to en códigos de ingeniería. ▪ Prohibición y ausencia de obligación en licencias de software y privacidad de datos. ▪ Vocabulario: Ética profesional, propiedad intelectual, IA responsable. ▪ Redacción: Documentos formales de recomendación técnica y política.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 2: Fallos y Causas Raíz (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pasado Perfecto: Estructura y uso en análisis de causa raíz y post-mortems. ▪ Contraste entre Pasado Simple, Continuo y Perfecto en cronologías de fallos. ▪ Vocabulario: Interrupciones de sistema, brechas de seguridad, fallos en cascada. ▪ Expresiones para reportar información: estilo indirecto en informes de incidentes.	▪ El estudiante reconstruye cronologías complejas de fallos y análisis de causas raíz estableciendo secuencias cronológicas claras mediante el contraste de tiempos de pasado.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 3: Lógica Si-Entonces (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Condicionales Tipo 0, 1 y 2 aplicados a la lógica computacional y decisiones de ingeniería. ▪ Oraciones con Ïfz Ünless"para restricciones y requisitos de sistemas. ▪ Vocabulario: Escenarios hipotéticos, análisis de riesgo y trade-offs de optimización. ▪ Pensamiento de diseño hipotético: Ïf we used X, the system would..."	▪ El estudiante plantea escenarios técnicos reales e hipotéticos y sus consecuencias, utilizando estructuras condicionales para razonar sobre decisiones de diseño e ingeniería.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 4: Investigación en Curso (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Perfecto Continuo para describir investigación en progreso. ▪ Contraste entre Presente Perfecto Simple y Continuo en escritura científica. ▪ Vocabulario: Integración continua, desarrollo ágil, investigación iterativa. ▪ Formación de palabras: Prefijos y sufijos en terminología científica y técnica.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 5: Fabricación y Despliegue (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Voz Pasiva en todos los tiempos básicos para descripciones de procesos técnicos. ▪ Verbos con dos objetos en pasiva: data is sent to the server by the client. ▪ Vocabulario: Fabricación de hardware, despliegue en la nube, pipelines CI/CD. ▪ Redacción: Descripción técnica de un proceso de despliegue o producción.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 6: Hojas de Ruta Tecnológicas (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Futuro Continuo y Futuro Perfecto para cronogramas e hitos de proyectos. ▪ Adverbios de probabilidad: perhaps, probably, definitely — en prospectiva tecnológica. ▪ Vocabulario: Hojas de ruta de producto, planificación de versiones, áreas de investigación emergente. ▪ Presentaciones orales: .[El] estado de la IA / computación cuántica en 5 años”.	▪ El estudiante proyecta estados técnicos, hitos futuros completados y resultados de investigación, discutiendo hojas de ruta de computación e ingeniería con vocabulario avanzado.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 7: Equipos Técnicos (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Preguntas indirectas y etiquetas (Tag questions) en entrevistas técnicas y revisiones de pares. ▪ Phrasal Verbs comunes en entornos de ingeniería e investigación. ▪ Vocabulario: Herramientas de colaboración, prácticas de revisión de código, roles de equipo. ▪ Redacción: Publicaciones cortas de blog técnico y actualizaciones de equipo de ingeniería.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unidad 8: Inglés Técnico IV: Redes y Seguridad (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Terminología de redes: protocolos, topologías, modelo OSI, enrutamiento. ▪ Vocabulario de ciberseguridad: vulnerabilidades, amenazas, cifrado, autenticación. ▪ Lectura y redacción de informes de incidentes de red y auditorías de seguridad. ▪ Proyecto: Evaluación de seguridad para una infraestructura computacional.	▪ 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.
Lecturas : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS212. Análisis y Diseño de Algoritmos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS212. Análisis y Diseño de Algoritmos
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	▪ CS210. Algoritmos y Estructuras de Datos. (4to Sem) ▪ CS211. Teoría de la Computación. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Estructuras de Datos Fundamentales (6 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipo de Dato Abstracto (ADT) y operaciones sobre un ADT: • Operaciones de diccionario (insertar, eliminar, encontrar) ▪ Arreglos: • Numéricos vs no numéricos, cadenas de caracteres • Unidimensionales (vector) vs multidimensionales (matriz) ▪ Registros/Estructuras/Tuplas y Objetos ▪ Listas enlazadas (por razones históricas): • Simples vs Dobles y Lineales vs Circulares ▪ Pilas ▪ Colas y deques: • Cola de prioridad basada en montículo ▪ Tablas/mapas hash: • Resolución de colisiones y complejidad (por ejemplo, sondeo, encadenamiento, rehash) ▪ árboles: • Binarios, n-arios y árboles de búsqueda • Balanceados (por ejemplo, AVL, Rojo-Negro, Montículo) ▪ Conjuntos	▪ Para cada ADT/Estructura de Datos en esta unidad: • Explicar su definición, propiedades, representación(es) y operaciones de ADT asociadas. • Explicar paso a paso cómo las operaciones de ADT asociadas con la estructura de datos la transforman. [Explicar] ▪ Explicar cómo se maneja la evitación de colisiones y la resolución de colisiones en tablas hash [Explicar] ▪ Explicar la propiedad de montículo y el uso de montículos como una implementación de una cola de prioridad [Explicar]
Lecturas : [**Cormen2022**], [**Kleinberg2005**]	

Unidad 2: Algoritmos Fundamentales (6 horas)

Resultados esperados: AG-C08,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Grafos (por ejemplo, [no]dirigidos, [a]cíclicos, [no]conexos y [no]ponderados): • Representación de grafos: lista de adyacencia vs matriz ▪ Algoritmos de búsqueda: • Complejidad $O(n)$ (por ejemplo, búsqueda lineal/secuencial en arreglo/lista) • Complejidad $O(\log_2 n)$ (por ejemplo, búsqueda binaria) • Complejidad $O(\log_b n)$ (por ejemplo, búsqueda en árbol no informada en profundidad/amplitud) ▪ Algoritmos de ordenamiento (por ejemplo, estables, inestables): • Complejidad $O(n^2)$ (por ejemplo, inserción, selección) • Complejidad $O(n \log n)$ (por ejemplo, quicksort, merge, timesort) ▪ Algoritmos de grafos: • Camino más corto (por ejemplo, Dijkstra, Floyd) • árbol de expansión mínima (por ejemplo, Prim, Kruskal) ▪ Algoritmos de ordenamiento: • Complejidad $O(n \log n)$ heapsort • Pseudo $O(n)$ complejidad (por ejemplo, bucket, counting, radix) ▪ Algoritmos de grafos: • Clausura transitiva (por ejemplo, Warshall) • Ordenamiento topológico ▪ Emparejamiento: • Emparejamiento eficiente de cadenas (por ejemplo, Boyer-Moore, Knuth-Morris-Pratt) • Emparejamiento de subsecuencia común más larga • Emparejamiento de expresiones regulares	▪ Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar] ▪ Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar] ▪ 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] ▪ 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] ▪ Para cada uno de los algoritmos y enfoques algorítmicos en los temas del Núcleo KA: • Explicar un ejemplo prototípico del algoritmo, y • Explicar paso a paso cómo opera el algoritmo. [Explicar]

Lecturas : [**Cormen2022**], [**Kleinberg2005**]

Unidad 3: Algoritmos Avanzados (4 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Algoritmos de criptografía (por ejemplo, SHA-256) ▪ Algoritmos paralelos ▪ Algoritmos de consenso (por ejemplo, Blockchain): • Prueba de trabajo vs prueba de participación ▪ Algoritmos de computación cuántica: • Basados en oráculo (por ejemplo, Deutsch-Jozsa, Bernstein-Vazirani, Simon) • Aceleración superpolinomial mediante QFT (por ejemplo, Shor) • Aceleración polinomial mediante amplificación de amplitud (por ejemplo, Grover) ▪ Algoritmo de Transformada Rápida de Fourier (FFT) ▪ Algoritmo de evolución diferencial	▪ Una apreciación de la computación cuántica y su aplicación a ciertos problemas [Explicar]
Lecturas : [**Cormen2022**], [**Kleinberg2005**]	

Unidad 4: Estrategias Algorítmicas (24 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas: • Fuerza Bruta (por ejemplo, búsqueda lineal, ordenamiento por selección, viajante de comercio, mochila) • Disminuir y Vencer: ◦ Por una Constante (por ejemplo, ordenamiento por inserción, ordenamiento topológico) ◦ Por un Factor Constante (por ejemplo, búsqueda binaria) ◦ Por un Tamaño Variable (por ejemplo, Euclides) • Dividir y Vencer (por ejemplo, búsqueda binaria, quicksort, mergesort, Strassen) • Voraz (por ejemplo, Dijkstra, Kruskal, Mochila) • Transformar y Vencer: ◦ Simplificación de instancia (por ejemplo, encontrar duplicados mediante preordenamiento de lista) ◦ Cambio de representación (por ejemplo, heapsort) ◦ Reducción de problema (por ejemplo, mínimo común múltiplo, programación lineal) ◦ Programación dinámica (por ejemplo, Floyd, Marshall, Bellman-Ford) • Compromisos espacio vs tiempo (por ejemplo, hash) ▪ Manejo del crecimiento exponencial (por ejemplo, heurística A*, ramificación y poda, retroceso) ▪ Iteración vs recursión (por ejemplo, factorial, búsqueda en árbol) ▪ Paradigmas: • Algoritmos de aproximación • Mejora iterativa (por ejemplo, Ford-Fulkerson, simplex) • Algoritmos aleatorizados/estocásticos (por ejemplo, corte máximo, bolas y cubos) ▪ Computación cuántica	▪ Para cada uno de los paradigmas en esta unidad: • Explicar sus características definitorias • Explicar un ejemplo que demuestre el paradigma incluyendo cómo este ejemplo satisface las características del paradigma. [Explicar] ▪ 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] ▪ Dado un algoritmo, explicar el paradigma utilizado por el algoritmo y cómo ejemplifica este paradigma [Explicar] ▪ 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] ▪ Dar ejemplos de algoritmos iterativos y recursivos que resuelvan el mismo problema, explicar los beneficios y desventajas de cada enfoque [Explicar] ▪ Evaluar si un enfoque voraz conduce a una solución óptima [Evaluar] ▪ Explicar varios enfoques para abordar problemas computacionales cuyas soluciones algorítmicas son exponenciales [Explicar]
Lecturas : [**Cormen2022**], [**Dasgupta2006**]	

Unidad 5: Marco de Análisis de Complejidad (6 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Marco de Análisis de Complejidad: • Rendimiento de un algoritmo en el mejor caso, caso promedio y peor caso • Mediciones empíricas y relativas (Orden de Crecimiento) • Tamaño de entrada y operaciones primitivas • Eficiencia de tiempo y espacio ▪ Mediciones empíricas de rendimiento ▪ Compromisos tiempo-espacio en algoritmos	▪ 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] ▪ 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] ▪ Explicar ejemplos que ilustren los compromisos tiempo-espacio de algoritmos [Explicar] ▪ Explicar cómo el balance del árbol afecta la eficiencia de las operaciones de árbol de búsqueda binaria [Explicar]
Lecturas : [**Cormen2022**], [**Sipser2012**]	

Unidad 6: Notación Asintótica y Clases de Complejidad (6 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis de complejidad asintótica (cotas promedio y peor caso): • Notaciones formales Big-O, Big-Omega y Big-Theta • Clases de Complejidad Fundamentales y Ejemplos/Problemas Representativos: ◦ $O(1)$ Constante (por ejemplo, acceso a arreglo) ◦ $O(\log_2 n)$ Logarítmica (por ejemplo, búsqueda binaria) ◦ $O(n)$ Lineal (por ejemplo, búsqueda lineal) ◦ $O(n \log_2 n)$ Log Lineal (por ejemplo, mergesort) ◦ $O(n^2)$ Cuadrática (por ejemplo, ordenamiento por selección) ◦ $O(n^c)$ Polinomial (por ejemplo, $O(n^3)$ eliminación gaussiana) ◦ $O(2^n)$ Exponencial (por ejemplo, Mochila, Satisfactibilidad (SAT), Viajante de Comercio (TSP), todos los subconjuntos) ◦ $O(n!)$ Factorial (por ejemplo, circuito hamiltoniano, todas las permutaciones)	▪ Para cada clase de complejidad fundamental en esta unidad, explicar un algoritmo que demuestre la complejidad de tiempo de ejecución asociada [Explicar] ▪ Aplicar la notación Big-O para dar cotas superiores en la complejidad tiempo/espacio de algoritmos [Aplicar]
Lecturas : [**Cormen2022**], [**Sipser2012**]	

Unidad 7: Análisis de Complejidad: Recursión, Amortización y Cotas Ajustadas (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Notaciones Little-o, Little-Omega y Little Theta ▪ Análisis recursivo formal ▪ Análisis amortizado	▪ 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] ▪ Usar relaciones de recurrencia para evaluar la complejidad de tiempo de algoritmos definidos recursivamente [Aplicar] ▪ Aplicar relaciones de recurrencia elementales usando una forma del Teorema Maestro [Aplicar]
Lecturas : [**Cormen2022**], [**Sipser2012**]	

Unidad 8: Teoría de la Complejidad Computacional (6 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tractabilidad e intractabilidad: • Clases de Complejidad P, NP y NP-Completo • Problemas NP-Completos (por ejemplo, SAT, Mochila, TSP) • Reducciones	▪ Explicar la importancia de la NP-Completitud [Explicar] ▪ Explicar ejemplos de problemas NP-completos [Explicar] ▪ Explicar las clases P y NP [Explicar]
Lecturas : [**Sipser2012**], [**Cormen2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS261. Inteligencia Artificial (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS261. Inteligencia Artificial
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	STA251. Estadística y Probabilidades. (3er Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Introducción a la IA: Conceptos y Comportamiento Inteligente (4 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de los problemas de IA, Ejemplos de aplicaciones recientes exitosas de IA ▪ Definiciones de agentes con ejemplos (por ejemplo, reactivos, deliberativos) ▪ ¿Qué es el comportamiento inteligente?: • La prueba de Turing y sus defectos • Entrada y salida multimodal • Simulación del comportamiento inteligente • Razonamiento racional versus no racional ▪ Visión general de las Aplicaciones de IA, crecimiento e impacto (económico, social, ético) ▪ Problemas filosóficos ▪ Historia de la IA	▪ Describir la prueba de Turing y el experimento mental de la ”Habitación China” [Explicar] ▪ Diferenciar entre razonamiento/comportamiento óptimo y razonamiento/comportamiento similar al humano [Evaluar] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Explicar]
Lecturas : [**Russell2020**], [**Luger2008**]	

Unidad 2: Características del Problema y Naturaleza de los Agentes (4 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Características del problema: • Totalmente observable versus parcialmente observable • Agente único versus multiagente • Determinista versus estocástico • Estático versus dinámico • Discreto versus continuo ▪ Naturaleza de los agentes: • Autonomía autónoma, semiautónoma, de iniciativa mixta • Reflexivos, basados en metas y basados en utilidad • Toma de decisiones bajo incertidumbre y con información incompleta • La importancia de la percepción y las interacciones ambientales • Agentes basados en aprendizaje • Agentes corporeizados: ◦ sensores, dinámicas, efectores ▪ Practicar la identificación de características del problema en entornos de ejemplo ▪ Profundidad adicional sobre la naturaleza de los agentes con ejemplos ▪ Profundidad adicional sobre Aplicaciones de IA, Crecimiento e Impacto (económico, social, ético, seguridad)	▪ Enumerar las características de un problema específico [Aplicar]
Lecturas : [**Russell2020**], [**Luger2008**]	

Unidad 3: Estado y Búsqueda No Informada (5 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Representación del espacio de estados de un problema: • Especificación de estados, metas y operadores • Factorización de estados en representaciones (espacios de hipótesis) • Resolución de problemas mediante búsqueda en grafos: ◦ por ejemplo, Grafos como un espacio y recorridos de árboles como exploración de ese espacio ◦ Construcción dinámica del grafo (no dada de antemano) ▪ Búsqueda en grafo no informada para resolución de problemas: • Búsqueda en amplitud • Búsqueda en profundidad: ◦ Con profundización iterativa • Búsqueda de costo uniforme ▪ Complejidades de espacio y tiempo de los algoritmos de búsqueda en grafos ▪ Búsqueda bidireccional ▪ Comprensión del espacio de búsqueda: • Construcción de árboles de búsqueda • Espacios de búsqueda dinámicos • Explosión combinatoria del espacio de búsqueda • Topología del espacio de búsqueda (por ejemplo, crestas, puntos de silla, mínimos locales)	▪ Diseñar la representación del espacio de estados para un rompecabezas (por ejemplo, N-reinas o problema de los 3 jarrones) [Aplicar] ▪ 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]
Lecturas : [**Russell2020**], [**Luger2008**]	

Unidad 4: Búsqueda Heurística y CSP (6 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Búsqueda en grafo heurística para resolución de problemas: • Construcción heurística y admisibilidad • Ascenso de colinas • Mínimos locales y el paisaje de búsqueda: ◦ Soluciones locales versus globales • Búsqueda primero el mejor voraz • Búsqueda A* ▪ Búsqueda en haz ▪ Implementación de la búsqueda A* ▪ Satisfacción de restricciones ▪ Variaciones de A* (IDA*, SMA*, RBFS)	▪ 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] ▪ Evaluar si una heurística para un problema dado es admisible/puede garantizar una solución óptima [Evaluar] ▪ 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] ▪ Comparar y contrastar varias búsquedas heurísticas con respecto a la aplicabilidad a un problema dado [Evaluar] ▪ 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]
Lecturas : [**Russell2020**], [**Luger2008**]	

Unidad 5: Búsqueda Adversarial y Optimización Estocástica (5 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Juegos adversariales de dos jugadores: • Búsqueda minimax • Poda alfa-beta: ◦ Corte por profundidad ▪ Búsqueda local ▪ Búsqueda tabú ▪ Juegos adversariales de dos jugadores: • El efecto horizonte • Libros de apertura/soluciones de finales • Qué significa ”resolver” un juego (por ejemplo, damas) ▪ Implementación de búsqueda minimax, búsqueda en haz ▪ Búsqueda expectimax (resolución MDP) y nodos de azar ▪ Búsqueda estocástica: • Recocido simulado • Algoritmos genéticos • Búsqueda en árbol Monte Carlo	▪ 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] ▪ 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] ▪ 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] ▪ Diseñar e implementar una solución de algoritmo genético para un problema [Aplicar] ▪ Diseñar e implementar un programa de recocido simulado para evitar mínimos locales en un problema [Aplicar]
Lecturas : [**Russell2020**], [**Luger2008**]	

Unidad 6: Fundamentos de Aprendizaje Automático (7 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: • Aprendizaje supervisado: ◦ Clasificación ◦ Regresión • Aprendizaje por refuerzo • Aprendizaje no supervisado: ◦ Agrupamiento ▪ Ideas fundamentales: • Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. • Fuentes de error e indecidibilidad en el aprendizaje automático ▪ Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: • Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente ▪ El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): • El compromiso entre sesgo (subajuste) - varianza (sobreajuste)	▪ 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] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] ▪ 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] ▪ Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] ▪ Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]
Lecturas : [**Russell2020**], [**Alpaydin2020**]	

Unidad 7: Aprendizaje Automático: Datos, Representaciones y Evaluación (7 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Trabajar con Datos: • Preprocesamiento de datos: ◦ Importancia y peligros de las elecciones de preprocesamiento • Manejo de valores faltantes (imputación, marcar como faltante): ◦ Implicaciones de imputar versus marcar como faltante • Codificación de variables categóricas, codificación de datos de valor real • Normalización/estandarización • énfasis en datos reales, no ejemplos de libros de texto ▪ Representaciones: • Espacios de hipótesis y complejidad • Expansión de características de base simple, como elevar al cuadrado características univariadas • Representaciones de características aprendidas ▪ Evaluación del aprendizaje automático: • Separación de conjuntos de entrenamiento, validación y prueba • Métricas de rendimiento para clasificadores • Estimación del rendimiento en prueba en datos retenidos • Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación • Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones	▪ Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] ▪ Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] ▪ 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] ▪ 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]
Lecturas : [**Russell2020**], [**Alpaydin2020**]	

Unidad 8: Redes Neuronales y Ética (6 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redes neuronales básicas: • Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos • Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) ▪ ética para el Aprendizaje Automático: • Enfocarse en datos reales, escenarios reales y estudios de caso • Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas	▪ Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]
Lecturas : [**Russell2020**], [**Alpaydin2020**]	

Unidad 9: Formulación del Aprendizaje Automático y Evaluación (3 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: • Función objetivo • Descenso de gradiente • Regularización para evitar sobreajuste (formulación matemática) ▪ Conjuntos de modelos: • Combinación de mayoría ponderada simple ▪ Evaluación de rendimiento: • Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) • Métricas de rendimiento para regresores • Matriz de confusión • Validación cruzada: ◦ Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)	▪ 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] ▪ Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] ▪ Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] ▪ Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] ▪ Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]
Lecturas : [**Goodfellow2016**]	

Unidad 10: Aprendizaje Profundo, RL y Ética (3 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aprendizaje profundo: • Redes feed-forward profundas (solo intuición, sin matemáticas) • Redes neuronales convolucionales (solo intuición, sin matemáticas) • Visualización de representaciones de características aprendidas de redes profundas • Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) ▪ Visión general de los métodos de aprendizaje por refuerzo ▪ Dos o más aplicaciones de algoritmos de aprendizaje automático: • Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego ▪ ética para el Aprendizaje Automático: • Continuar enfocándose en datos reales, escenarios reales y estudios de caso • Privacidad • Equidad • Propiedad intelectual • Explicabilidad	▪ 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] ▪ 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]
Lecturas : [**Goodfellow2016**]	

Unidad 11: Representación del Conocimiento y Razonamiento Probabilístico Básico (6 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipos de representaciones: • Simbólicas, lógicas: ◦ Crear una representación a partir de un enunciado de problema en lenguaje natural • Representaciones subsimbólicas aprendidas • Modelos gráficos (por ejemplo, Bayes ingenuo, red bayesiana) ▪ Repaso del razonamiento probabilístico, teorema de Bayes ▪ Razonamiento bayesiano: • Inferencia bayesiana ▪ Variables aleatorias y distribuciones de probabilidad: • Axiomas de probabilidad • Inferencia probabilística • Regla de Bayes (derivación) • Inferencia bayesiana (ejemplos más complejos)	▪ Dado un enunciado de problema en lenguaje natural, codificarlo como una representación simbólica o lógica [Aplicar] ▪ 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] ▪ Aplicar la regla de Bayes para determinar la probabilidad de una hipótesis dada la evidencia [Aplicar] ▪ Calcular la probabilidad de resultados y probar si los resultados son independientes [Aplicar]
Lecturas : [**Russell2020**]	

Unidad 12: Independencia, Modelos de Markov y Toma de Decisiones (6 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Independencia ▪ Independencia condicional ▪ Cadenas de Markov y modelos de Markov ▪ Utilidad y toma de decisiones	▪ Explicar cómo podemos tomar decisiones bajo incertidumbre, usando conceptos como el teorema de Bayes y la utilidad [Explicar]
Lecturas : [**Russell2020**]	

Unidad 13: Aplicaciones e Impacto Social de la IA (4 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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): • Formular y evaluar una aplicación específica como un problema de IA: ◦ Cómo lidiar con problemas mal especificados o mal planteados • Disponibilidad/escasez y limpieza de datos: ◦ Limpieza y preprocesamiento básico de datos ◦ Sesgo del conjunto de datos • Sesgo algorítmico • Sesgo de evaluación • Evaluación de las implicaciones sociales de la aplicación ▪ Modelos generativos profundos desplegados: • 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. • Visión general de alto nivel de modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard, etc.), sus usos y sus limitaciones/peligros. ▪ Visión general del impacto social de la IA: • ética • Equidad • Confianza/explicabilidad • Privacidad y uso de datos de entrenamiento • Autonomía humana y supervisión/regulaciones/requisitos legales • Sostenibilidad	▪ 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]
Lecturas : [**Russell2020**], [**Alpaydin2020**]	

Unidad 14: Aplicaciones Adicionales e Impacto Social de la IA (4 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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): • Formular y evaluar una aplicación específica como un problema de IA: ◦ Cómo lidiar con problemas mal especificados o mal planteados • Disponibilidad/escasez y limpieza de datos: ◦ Limpieza y preprocesamiento básico de datos ◦ Sesgo del conjunto de datos • Sesgo algorítmico • Sesgo de evaluación • Evaluación de las implicaciones sociales de la aplicación ▪ Profundidad adicional sobre modelos generativos profundos desplegados: • Introducción a cómo funcionan los modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion) incluyendo discusión de atención • Introducción a cómo funcionan los modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard) incluyendo discusión de atención • Idea de modelos fundamentales, cómo usarlos, y los beneficios/problemas con entrenarlos a partir de grandes datos ▪ Análisis y discusión del impacto social de la IA: • ética • Equidad • Confianza/explicabilidad • Privacidad y uso de datos de entrenamiento • Autonomía humana y supervisión/regulaciones/requisitos legales • Sostenibilidad	▪ 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] ▪ 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]
Lecturas : [**Russell2020**], [**Alpaydin2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS272. Bases de Datos II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS272. Bases de Datos II
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS271. Bases de Datos I. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso es una continuación de Gestión de Datos I, enfocándose en los mecanismos internos de los sistemas de gestión de bases de datos (DBMS) y tecnologías de bases de datos modernas. Cubre en profundidad la optimización de consultas, estructuras de indexación e integridad transaccional. Además, introduce a los estudiantes a modelos no relacionales (NoSQL) y arquitecturas distribuidas, que son esenciales para manejar datos a gran escala en entornos de nube contemporáneos.

5. OBJETIVOS

- Comprender las estructuras de almacenamiento interno y técnicas de indexación.
- Analizar y optimizar planes de ejecución de consultas.
- Comprender protocolos de transacción avanzados y mecanismos de recuperación.
- Evaluar e implementar soluciones usando modelos de bases de datos NoSQL.
- Comprender los principios de bases de datos distribuidas y paralelas.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Procesamiento de Consultas (12 horas)

Resultados esperados: AG-C08,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructuras de página ▪ Estructuras de índice: • árboles B+ • Índices hash: estáticos y dinámicos • Creación de índices en SQL ▪ Estructuras de archivos: • Archivos de montón (*heap*) • Archivos hash ▪ Algoritmos para operadores de consulta: • Ordenamiento externo • Selección • Proyección; con y sin eliminación de duplicados • Uniones naturales: Bucle anidado, Mezcla-ordenada, Unión hash • Análisis de eficiencia de algoritmos ▪ Transformaciones de consultas ▪ Optimización de consultas: • Rutas de acceso • Construcción de plan de consulta • Estimación de selectividad • Planes solo de índice ▪ Procesamiento de Consultas en Paralelo (por ejemplo, exploración paralela, unión paralela, agregación paralela) ▪ Afinamiento/rendimiento de base de datos: • Selección de índices • Impacto de índices en el rendimiento de consultas • Desnormalización	▪ Describir el propósito y organización de las estructuras de índice tanto de árbol B+ como hash [Describir] ▪ Componer un comando SQL para crear un índice (de cualquier tipo) [Componer] ▪ Especificar los pasos para los diversos algoritmos de operadores de consulta: ordenamiento externo, proyección con eliminación de duplicados, unión por mezcla-ordenada, unión hash, unión por bucle anidado por bloques [Analizar] ▪ Derivar el tiempo de ejecución (en solicitudes de E/S) para cada uno de los algoritmos anteriores [Analizar] ▪ Transformar una consulta en álgebra relacional a su equivalente apropiado para una ejecución canalizada de profundidad izquierda [Rediseñar] ▪ Calcular estimaciones de selectividad para una operación de selección y/o unión dada [Computar/Calcular] ▪ Describir cómo modificar una estructura de índice para facilitar una operación solo de índice para una relación dada [Describir] ▪ Para un escenario dado decidir qué índices soportar para la ejecución eficiente de un conjunto de consultas [Evaluar (valorar)] ▪ Describir cómo los DBMS aprovechan el paralelismo para acelerar el procesamiento de consultas dividiendo el trabajo entre múltiples procesadores o nodos [Describir]
Lecturas : [**Silberschatz2019**], [**Ramakrishnan2003**]	

Unidad 2: Internos de DBMS (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de Búfer de DB ▪ Gestión de Transacciones: • Niveles de Aislamiento • ACID • Serializabilidad • Transacciones Distribuidas ▪ Control de Concurrencia: • Bloqueo en 2 Fases • Estrategias de manejo de bloqueos mutuos • Modelos de consistencia basados en quórum ▪ Gestor de Recuperación: • Relación con el Gestor de Búfer ▪ Control de Concurrencia: • Control de concurrencia optimista • Control de concurrencia por marca de tiempo ▪ Gestor de Recuperación: • Registro por escritura anticipada • Sistema de recuperación ARIES (Análisis, REDO, UNDO)	▪ Describir cómo un DBMS gestiona su Grupo de Búferes [Describir] ▪ Describir las cuatro propiedades para un gestor de transacciones correcto [Describir] ▪ Esbozar el principio de serializabilidad [Esquematizar/Esbozar]
Lecturas : [**Silberschatz2019**], [**GarciaMolina2008**]	

Unidad 3: Bases de Datos Distribuidas/Computación en la Nube (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ DBMS Distribuido: • Almacenamiento de datos distribuido • Procesamiento de consultas distribuido • Modelo de transacción distribuida • Soluciones homogéneas y heterogéneas • Bases de datos distribuidas cliente-servidor ▪ DBMS Paralelo: • Arquitecturas de DBMS paralelo: memoria compartida, disco compartido, nada compartido • Aceleración y escalamiento, por ejemplo, uso del modelo de procesamiento MapReduce • Replicación de datos y modelos de consistencia débil	▪ Describir los componentes clave de un DBMS distribuido, incluyendo almacenamiento de datos distribuido, procesamiento de consultas y gestión de transacciones [Describir] ▪ Analizar las ventajas y desventajas entre arquitecturas de DBMS paralelo: memoria compartida, disco compartido y nada compartido [Analizar] ▪ Describir estrategias de replicación de datos y modelos de consistencia débil en sistemas de bases de datos distribuidas [Describir]
Lecturas : [**Silberschatz2019**], [**Ozsu2020**]	

Unidad 4: Sistemas NoSQL (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ ¿Por qué NoSQL? (por ejemplo, Desajuste de impedancia entre Aplicación [CRUD] y RDBMS) ▪ Modelo de datos Clave-Valor y Documento ▪ Sistemas de almacenamiento (por ejemplo, sistemas Clave-Valor, Lagos de Datos (*Data Lakes*)) ▪ Modelos de Distribución (Fragmentación y Replicación) ▪ Bases de Datos de Grafos ▪ Modelos de Consistencia (Actualización y Lectura, consistencia de quórum, teorema CAP) ▪ Modelo de procesamiento (por ejemplo, MapReduce, map-reduce multi-etapa, map-reduce incremental) ▪ Estudios de Caso: Sistema de almacenamiento en la nube (por ejemplo, S3); Bases de datos de grafos; "Cuándo no usar NoSQL"	▪ Desarrollar un caso de uso para el uso de NoSQL sobre RDBMS [Crear] ▪ Describir las características definitorias detrás de los modelos de datos basados en Clave-Valor y Documentos [Describir]
Lecturas : [**Sadalage2012**], [**kleppmann2017designing**]	

Unidad 5: Bases de Datos Semiestructuradas y No Estructuradas (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Datos no estructurados vectorizados (texto, video, audio, etc.) y almacenamiento vectorial: • Vectorizador TF-IDF con n-grama • Word2Vec • Base de datos de arreglos o manejo de tipo de datos arreglo ▪ Bases de datos semiestructuradas (por ejemplo, JSON): • Almacenamiento: ◦ Codificación y compresión de tipos de datos anidados • Indexación: ◦ árbol B, índice de salto, filtro Bloom ◦ Índice invertido y compresión de mapa de bits ◦ Indexación por curva de llenado de espacio para datos geo-semiestructurados • Procesamiento de consultas para casos de uso OLTP y OLAP: ◦ Compromisos de inserción, selección, actualización/eliminación ◦ Estudios de caso sobre Postgres/JSON, MongoDB y Snowflake/JSON	▪ Describir representaciones vectorizadas de datos no estructurados (por ejemplo, TF-IDF, Word2Vec) y sus modelos de almacenamiento [Describir] ▪ Aplicar técnicas de indexación (por ejemplo, árbol B, índice invertido, filtro Bloom) a datos semiestructurados almacenados en formato JSON [Aplicar] ▪ Analizar las ventajas y desventajas del procesamiento de consultas en bases de datos semiestructuradas para casos de uso OLTP y OLAP [Analizar]
Lecturas : [**kleppmann2017designing**], [**manning2008introduction**], [**mikolov2013efficient**], [**agrawal2023toward**], [**mongodb_documentation**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las

diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS291. Ingeniería de Software I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS291. Ingeniería de Software I
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	▪ CS113. Programación Orientada a Objetos II. (3er Sem) ▪ CS271. Bases de Datos I. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Control de Versiones y CI/CD (12 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Herramientas de prueba, incluyendo herramientas de análisis estático y dinámico. Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Flujo de revisión colaborativa como puerta de calidad • Revisión obligatoria de *pull requests* antes de fusionar • Propiedad del código mediante reglas de tipo CODEOWNERS • Convenciones de mensajes de *commit* (*conventional commits*) • Reglas de protección de ramas que exigen verificaciones antes de fusionar ▪ Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] ▪ 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] ▪ 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] ▪ Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 2: Ingeniería de Requisitos (14 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Describir requisitos funcionales usando, por ejemplo, casos de uso o historias de usuario. • Usar al menos un método para documentar y estructurar requisitos funcionales. • Comprender cómo el método apoya el diseño y la implementación. • Fortalezas y debilidades de usar un enfoque específico. ▪ Propiedades de los requisitos, incluyendo consistencia, validez, completitud y viabilidad. ▪ Obtención (*elicitation*) de requisitos. • Fuentes de requisitos, por ejemplo, usuarios, administradores o personal de soporte. • Métodos de recopilación de requisitos, por ejemplo, encuestas, entrevistas o análisis de comportamiento. ▪ Requisitos no funcionales, por ejemplo, seguridad, usabilidad o rendimiento, también llamados Atributos de Calidad. Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social ▪ Identificación y gestión de riesgos, incluyendo consideraciones éticas en torno al producto propuesto. Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo ▪ Comunicar y/o formalizar especificaciones de requisitos.	▪ Comparar diferentes métodos de obtención de requisitos a lo largo de múltiples ejes [Comparar] ▪ 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] ▪ Identificar qué comportamientos son requeridos, permitidos o prohibidos a partir de un conjunto dado de requisitos y una lista de comportamientos candidatos [Analizar] ▪ Recopilar un conjunto de requisitos para un sistema de software simple [Analizar] ▪ Identificar áreas de un sistema de software que deben cambiarse, dada una descripción del sistema y un conjunto de nuevos requisitos a implementar [Analizar] ▪ Identificar los requisitos funcionales y no funcionales en un conjunto de requisitos [Analizar]
Lecturas : [**Sommerville2015**], [**Wiegers2013**]	

Unidad 3: Colaboración y Dinámica de Equipo (14 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comunicación efectiva, incluida la oral y escrita, así como formal (correo electrónico, documentos, comentarios, presentaciones) e informal (chat de equipo, reuniones). Comunicación Técnica y Profesional, Comunicación en Equipo y Cultura ▪ Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos. ▪ Programación cooperativa: • Programación por parejas (pair programming) o Enjambre (swarming) • Revisión de código • Colaboración mediante control de versiones ▪ Roles y responsabilidades en un equipo de software: Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo • Ventajas del trabajo en equipo • Riesgos y complejidad de dicha colaboración ▪ Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo	▪ Seguir prácticas efectivas de comunicación en equipo [Analizar] ▪ 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] ▪ Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar] ▪ Colaborar efectivamente en desarrollo/programación cooperativa [Analizar] ▪ Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer] ▪ Redactar y seguir una agenda para una reunión de equipo [Componer] ▪ 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]
Lecturas : [**Wigdor2011**], [**Johnson2013**]	

Unidad 4: Principios y Arquitectura de Software (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de sistemas. Confiabilidad del Sistema • Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) • Separación de preocupaciones (separation of concerns) • Ocultación de información (information hiding) • Acoplamiento y cohesión (coupling and cohesion) ▪ Arquitectura de software. Confiabilidad del Sistema • Paradigmas de diseño ◦ Descomposición funcional de arriba hacia abajo/diseño por capas ◦ Arquitectura orientada a datos ◦ Análisis y diseño orientado a objetos ◦ Diseño dirigido por eventos • Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) • Identificar límites y dependencias de componentes ▪ Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema ▪ Malos olores del código (code smells) y otras indicaciones de la calidad del código, distintas de la corrección. Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad	▪ Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] ▪ 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] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] ▪ Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]
Lecturas : [**Gamma1994**], [**Pressman2019**]	

Unidad 5: Conceptos de verificación y validación (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conceptos de verificación y validación • Verificación: ¿Estamos construyendo la cosa correctamente? • Validación: ¿Construimos la cosa correcta? ▪ Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? ▪ Objetivos de las pruebas • Usabilidad • Confiabilidad • Conformidad con la especificación • Rendimiento • Seguridad	▪ Explicar por qué las pruebas son importantes [Explicar] ▪ Distinguir entre validación y verificación de programas [Distinguir] ▪ Describir diferentes objetivos de las pruebas [Describir]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 6: Herramientas y Análisis de Pruebas (4 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas y automatización de verificación y validación • Análisis estático • Cobertura de código • Fuzzing • Análisis dinámico y contención de fallos (sanitizadores, etc.) • Registro y seguimiento de fallos ▪ Planificación y generación de pruebas • Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos • Uso de números aleatorios y pseudoaleatorios en las pruebas ▪ Pruebas de sistemas asíncronos, paralelos y concurrentes ▪ Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) ▪ Desarrollo guiado por comportamiento (BDD) • Expresar criterios de aceptación como escenarios Gherkin Given/When/Then • Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo ▪ Pruebas de mutación y análisis estático de calidad de código • Pruebas de mutación para evaluar la efectividad de una suite de pruebas • Detección estática de *code smells* y complejidad ciclomática excesiva	▪ Describir y comparar diferentes herramientas para verificación y validación [Describir] ▪ Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] ▪ Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] ▪ Describir enfoques para la estimación de fallos [Describir] ▪ Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] ▪ Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] ▪ 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] ▪ Describir técnicas para la verificación y validación de artefactos que no son código [Describir] ▪ Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] ▪ Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 7: Pruebas de Rendimiento y Evaluación Comparativa (4 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas de rendimiento y evaluación comparativa • Rendimiento (*throughput*) y latencia • Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) • Aceleración (*speedup*) y escalado (*scaling*) ◦ Ley de Amdahl ◦ Ley de Gustafson ◦ Escalado suave (*soft*) y débil (*weak*) • Identificar y medir figuras de mérito (*figures of merit*) • Cuellos de botella de rendimiento comunes ◦ Limitado por cómputo (*compute-bound*) ◦ Limitado por ancho de banda de memoria ◦ Limitado por latencia • Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) ◦ Estimación de la incertidumbre ◦ Intervalos de confianza • Análisis y presentación (gráficos, etc.) • Técnicas de medición de tiempo (*timing*)	▪ Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] ▪ Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] ▪ Describir cuellos de botella de rendimiento comunes [Describir] ▪ Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] ▪ Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] ▪ 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]
Lecturas : [**Bondi2015**], [**Gregg2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS2S1. Sistemas Operativos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS2S1. Sistemas Operativos
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS221. Arquitectura de Computadores. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Rol y Propósito de los Sistemas Operativos (8 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Los sistemas operativos median entre el hardware de propósito general y el software específico de la aplicación. ▪ Funciones universales del sistema operativo (por ejemplo, proceso, interfaces de usuario y dispositivos, persistencia de datos) ▪ Funciones extendidas y/o especializadas del sistema operativo (por ejemplo, sistemas embebidos, tipos de servidor como archivos, web, multimedia, gestores de arranque y seguridad de arranque) ▪ Problemas de diseño (por ejemplo, eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad, energía, seguridad, compensaciones entre verificación de errores y rendimiento, flexibilidad y rendimiento, y seguridad y rendimiento). Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad ▪ Influencias de seguridad, redes, multimedia, computación paralela y distribuida ▪ Preocupación general de seguridad/protección: Omitir considerar la seguridad en cada capa crea una oportunidad para acceder de manera inapropiada a los recursos. • Se puede lograr acceso no autorizado a archivos en una unidad no cifrada moviendo el medio a otra computadora. • La seguridad impuesta por los sistemas operativos puede ser derrotada infiltrando la capa de arranque antes de que se cargue el sistema operativo. • El aislamiento de procesos puede ser subvertido por una verificación de autorización inadecuada en los límites de la API. • Las vulnerabilidades en el firmware del sistema pueden proporcionar vectores de ataque que eviten por completo el sistema operativo. • El aislamiento inadecuado de la memoria, cómputo y hardware de la máquina virtual puede exponer el sistema anfitrión a ataques desde sistemas invitados. • El sistema operativo puede necesitar mitigar la explotación de vulnerabilidades de hardware y firmware, lo que lleva a posibles reducciones de rendimiento (por ejemplo, mitigaciones de Spectre y Meltdown). ▪ Exposición de funciones del sistema operativo en shells y programación de sistemas. Ver también: Scripting de Shell	▪ Comprender los objetivos y funciones de los sistemas operativos modernos [Explicar] ▪ Evaluar los problemas de diseño en diferentes escenarios de uso (por ejemplo, sistema operativo en tiempo real, móvil, servidor) [Evaluar] ▪ Comprender las funciones de un sistema operativo contemporáneo con respecto a conveniencia, eficiencia y capacidad de evolucionar [Explicar] ▪ Comprender cómo la evolución y la estabilidad son deseables y mutuamente antagónicas en las funciones del sistema operativo [Explicar]

Lecturas : [**Silberschatz2018**], [**Tanenbaum2014**]

Unidad 2: Scripting de Shell (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Manejo de errores/excepciones ▪ Tuberías (piping) ▪ Comandos del sistema: • Interfaz con sistemas operativos ▪ Variables de entorno ▪ Abstracción de archivo y operadores ▪ Estructuras de datos, como arreglos y listas. ▪ Expresiones regulares ▪ Programas y procesos ▪ Flujo de trabajo	▪ Crear y ejecutar scripts automatizados para gestionar varias tareas del sistema [Crear] ▪ Resolver varios problemas de procesamiento de texto mediante scripting [Resolver]
Lecturas : [**Robbins2005**], [**Silberschatz2018**]	

Unidad 3: Planificación (16 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Planificación con expropiación y sin expropiación ▪ Planificadores y políticas (por ejemplo, primero en llegar primero en ser servido, trabajo más corto primero, prioridad, round robin, multinivel). Ver también: Gestión de Recursos ▪ Conceptos de planificación de multiprocesador simétrico (SMP) y coherencia de caché Ver también: Jerarquía de Memoria ▪ Temporizadores (por ejemplo, construir muchos temporizadores a partir de temporizadores de hardware finitos). Ver también: Organización de Máquina a Nivel de Ensamblador ▪ Justicia e inanición ▪ Subtemas de sistemas operativos como planificación consciente de energía y planificación en tiempo real. Ver también: Rendimiento y Eficiencia Energética, Características y Arquitecturas Embebidas, Tiempo Real, Memoria y Verificación, Sistemas, Programación y Sensores, Plataformas Móviles: Capacidades y Computación Asíncrona, Plataformas Móviles: Arquitectura y Restricciones, Plataformas Móviles: Fundamentos del Desarrollo ▪ Planificación cooperativa, como futexes de Linux y planificación en espacio de usuario.	▪ 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] ▪ Explicar las relaciones entre algoritmos de planificación y dominios de aplicación [Explicar] ▪ Explicar las distinciones entre tipos de planificadores de procesador como corto plazo, mediano plazo, largo plazo y E/S [Explicar] ▪ Evaluar un problema o solución para determinar la idoneidad para multiprocesamiento asimétrico y/o simétrico [Evaluar] ▪ Evaluar un problema o solución para determinar la idoneidad como proceso vs hilos [Evaluar] ▪ Listar algunos contextos que se benefician de la planificación con expropiación y con plazo límite [Listar/Enumerar] ▪ 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]
Lecturas : [**Silberschatz2018**], [**Stallings2017**]	

Unidad 4: Gestión de Memoria (16 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Revisión de memoria física, traducción de direcciones y hardware de gestión de memoria. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Impacto de la jerarquía de memoria incluyendo concepto de caché, búsqueda en caché y almacenamiento en caché por CPU en mecanismos y políticas del sistema operativo Ver también: Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento ▪ Direccionamiento lógico y físico, virtualización de espacio de direcciones. Ver también: Jerarquía de Memoria, Matemáticas Discretas ▪ Conceptos de paginación, reemplazo de páginas, trashing y asignación de páginas y marcos ▪ Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) rendimiento y flexibilidad • Arenas, asignadores de losas (slab allocators), listas libres, clases de tamaño, páginas de tamaño heterogéneo (páginas enormes) ▪ Almacenamiento en caché de memoria y coherencia de caché y el efecto de vaciar la caché para evitar vulnerabilidades de ejecución especulativaVer también: Organización Funcional, Jerarquía de Memoria, Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento) ▪ Mecanismos y conceptos de seguridad en gestión de memoria incluyendo sandboxing, protección, aislamiento y vectores relevantes de ataque. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Memoria virtual: aprovechando el hardware de memoria virtual para servicios del SO y eficiencia	▪ Explicar jerarquía de memoria y compensaciones costo-rendimiento [Explicar] ▪ Resumir los principios de memoria virtual aplicados a almacenamiento en caché y paginación [Resumir] ▪ Evaluar las compensaciones en términos de tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y velocidad del procesador [Evaluar] ▪ 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] ▪ 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] ▪ Explicar cómo se utiliza el hardware para una virtualización eficiente [Explicar]
Lecturas : [**Silberschatz2018**], [**ArpaciDusseau2018**]	

Unidad 5: API de Sistemas de Archivos e Implementación (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Concepto de archivo incluyendo datos, metadatos, operaciones y modo de acceso ▪ Montaje de sistema de archivos ▪ Control de acceso a archivos ▪ Compartición de archivos ▪ Métodos básicos de asignación de archivos, incluida la tabla de asignación enlazada ▪ Estructuras del sistema de archivos que comprenden asignación de archivos incluyendo varias estructuras de directorio y métodos para identificar archivos de manera única (por ejemplo, nombre, identificador o ubicación de almacenamiento de metadatos) ▪ Técnicas de asignación/desasignación/almacenamiento (algoritmos y estructura de datos) impacto en rendimiento y flexibilidad (es decir, fragmentación interna y externa y compactación) ▪ Gestión de espacio libre como usar tablas de bits vs enlazado ▪ Implementación de directorios para segmentar y rastrear ubicación de archivos	▪ Explicar las elecciones que deben tomarse en el diseño de sistemas de archivos [Explicar] ▪ Evaluar diferentes enfoques para organización de archivos, reconociendo las fortalezas y debilidades de cada uno [Evaluar] ▪ Aplicar constructores de software apropiadamente dado el conocimiento de la implementación del sistema de archivos [Aplicar]
Lecturas : [**Silberschatz2018**], [**Stallings2017**]	

Unidad 6: Protección y Seguridad (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descripción general de mecanismos de seguridad del sistema operativo. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Ataques y antagonismo (planificación, etc.). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Revisión de vulnerabilidades principales en sistemas operativos reales. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad ▪ Estrategias de mitigación del sistema operativo como copias de seguridad. Ver también: Confiabilidad del Sistema ▪ Separación política/mecanismo. Ver también: Gestión de Riesgos y Datos, Gobierno, Gestión y Política de Seguridad ▪ Métodos y dispositivos de seguridad. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad • Anillos de protección (historia desde Multics hasta x86 virtualizado) • Anillos -1 y -2 de x86_64 (hipervisor y ME/PSP) ▪ Protección, control de acceso y autenticación. Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Protocolos, Aplicaciones y Criptografía Avanzada, Algoritmos Criptográficos, Fundamentos de Criptografía	▪ Comprender el requisito de mecanismos de protección y seguridad en sistemas operativos [Explicar] ▪ Listar y describir los vectores de ataque que aprovechan vulnerabilidades del SO [Listar/Enumerar] ▪ Comprender los mecanismos disponibles en un SO para controlar el acceso a recursos [Explicar] ▪ Resumir las características y limitaciones de un sistema operativo que impactan la protección y seguridad [Resumir]
Lecturas : [**Silberschatz2018**], [**Tanenbaum2014**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

FG106. Teatro (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	FG106. Teatro
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	2
2.4 Horas	:	1 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: El Arte, la Creatividad y el Teatro (6 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ ¿Qué es el Arte? Una experiencia vivencial y personal. ▪ La llave maestra: la creatividad. ▪ La importancia del teatro en la formación personal y profesional. ▪ Utilidad y enfoque del arte teatral. ▪ El teatro como herramienta pedagógica y de desarrollo personal.	▪ Reconocer la vigencia del arte y la creatividad en el desarrollo personal y social [Usar]. ▪ Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usar]. ▪ Identificar nociones básicas del teatro y su función en la vida universitaria [Usar].
Lecturas : [**Majorana**], [**PAVIS**], [**Miravalles1998**]	

Unidad 2: El Juego: el quehacer del actor (6 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Juego, luego existo. El juego como lenguaje universal. ▪ El juego del niño y el juego dramático. ▪ Juegos de integración grupal y juegos de creatividad. ▪ La improvisación espontánea. ▪ La secuencia teatral y el calentamiento corporal.	▪ Reconocer el juego como herramienta fundamental del teatro [Usar]. ▪ Interiorizar y revalorar el juego como aprendizaje creativo [Usar]. ▪ Acercarse de manera espontánea y natural a la vivencia teatral [Usar].
Lecturas : [**PAVIS**], [**Miravalles1998**]	

Unidad 3: La expresión corporal y el uso dramático del objeto (9 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Toma de conciencia del cuerpo: el cuerpo como instrumento. ▪ Toma de conciencia del espacio escénico. ▪ Toma de conciencia del tiempo y el ritmo. ▪ Creación de secuencias individuales y colectivas: cuerpo, espacio y tiempo. ▪ El uso dramático del objeto: el juego teatral. ▪ Presentaciones teatrales con el uso del objeto.	▪ Experimentar nuevas formas de expresión y comunicación corporal [Usar]. ▪ Desarrollar mecanismos de control y manejo del cuerpo en el espacio escénico [Usar]. ▪ Cultivar la imaginación y la capacidad de respuesta a estímulos auditivos, rítmicos y visuales [Usar]. ▪ Explorar estados emocionales y climas colectivos a través del movimiento [Usar].
Lecturas : [**PAVIS**], [**Grotowski1970**]	

Unidad 4: Comunicación No Verbal en el Teatro (9 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Relajación, concentración y respiración. ▪ Desinhibición e interacción con el grupo. ▪ La improvisación: técnicas y ejercicios. ▪ Equilibrio, peso, tiempo y ritmo corporal. ▪ Análisis del movimiento: tipos y calidades. ▪ La presencia escénica y la energía del actor. ▪ La danza y la coreografía teatral. ▪ El gesto expresivo y el uso de la máscara.	▪ Ejercitarse en el manejo de destrezas comunicativas no verbales [Usar]. ▪ Practicar juegos y ejercicios de lenguaje corporal individual y grupal [Usar]. ▪ Expresar libre y creativamente emociones y sentimientos a través de representaciones originales [Usar]. ▪ Distinguir los diferentes tipos de actuación y sus fundamentos técnicos [Usar].
Lecturas : [**PAVIS**], [**Stanislavski1983**], [**Grotowski1970**]	

Unidad 5: Huellas del Teatro en el Tiempo (6 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ El origen del teatro: rito y representación en Grecia y Roma. ▪ El teatro medieval y la Commedia dell'Arte. ▪ Teatro del Siglo de Oro español y teatro isabelino inglés. ▪ Romanticismo, Naturalismo y Realismo: Ibsen y Chéjov. ▪ El teatro épico de Brecht y el teatro psicológico de Stanislavski. ▪ El teatro del absurdo, el teatro contemporáneo y el teatro total. ▪ Teatro en el Perú: Yuyachkani, La Tarumba, Pataclaun y otros.	▪ Conocer la influencia que la sociedad ha ejercido en el teatro a lo largo de la historia [Usar]. ▪ Apreciar el valor y el aporte de dramaturgos y directores teatrales fundamentales [Usar]. ▪ Analizar el contexto social e histórico del arte teatral en distintas épocas [Usar]. ▪ Reflexionar sobre el teatro peruano y sus expresiones contemporáneas [Usar].
Lecturas : [**PAVIS**], [**Balta2001**]	

Unidad 6: El Montaje Teatral (12 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Apreciación teatral: asistencia y análisis crítico de obras. ▪ El espacio escénico: tipos de escenario y su uso dramático. ▪ El texto dramático: lectura, análisis y adaptación. ▪ Construcción del personaje: técnica y psicología. ▪ Dirección y puesta en escena: fundamentos básicos. ▪ Elementos de producción: vestuario, maquillaje, escenografía y utilería. ▪ Presentación pública de ejercicios teatrales en grupo.	▪ Emplear la creación teatral como manifestación de ideas y sentimientos ante la sociedad [Usar]. ▪ Aplicar las técnicas practicadas en una propuesta escénica concreta y significativa [Usar]. ▪ Intercambiar experiencias y realizar presentaciones teatrales breves en grupo frente al público [Usar].
Lecturas : [**PAVIS**], [**Miravalles1998**], [**Brook1969**], [**Balta2001**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

BEI203. Inglés V (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	BEI203. Inglés V
2.2 Semestre	:	5to Semestre.
2.3 Créditos	:	2
2.4 Horas	:	4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BEI202. Inglés IV. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Argumentación Científica (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Repaso avanzado de verbos auxiliares y tiempos verbales en escritura académica. ▪ Preguntas negativas y respuestas enfáticas en el debate científico. ▪ Formación de palabras: Prefijos y sufijos comunes en computación y ciencias. ▪ Corrección de afirmaciones erróneas: Refutación de argumentos en discusiones técnicas.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 2: Metodología de Investigación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Simple vs. Presente Continuo: Distinción entre hechos generales y experimentos en curso. ▪ Verbos de estado (State verbs) y verbos de acción en descripciones experimentales. ▪ Vocabulario: Diseño de investigación, variables experimentales, métodos de recolección de datos. ▪ Redacción: Descripción de un método de investigación o procedimiento experimental.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 3: Estudios de Caso (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tiempos narrativos: Pasado Simple, Pasado Continuo y Pasado Perfecto en análisis de casos. ▪ Estructura de Üsed toz "Would"para describir sistemas heredados y prácticas pasadas. ▪ Vocabulario: Casos históricos de computación, cambios de paradigma, lecciones aprendidas. ▪ Redacción: Informe estructurado de estudio de caso sobre un incidente o innovación computacional.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 4: Propuestas de Investigación (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Predicciones y planes: Repaso del futuro para la redacción de propuestas académicas. ▪ Futuro Continuo y Futuro Perfecto en el alcance del proyecto y los resultados esperados. ▪ Vocabulario: Objetivos de investigación, contribuciones esperadas, alcance y limitaciones. ▪ Redacción: Propuesta de investigación corta o abstract de proyecto técnico.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 5: Documentación Técnica (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Patrones verbales (Verbos seguidos de -ing o infinitivo) en escritura instruccional. ▪ Expresiones con "Like"para analogías y comparaciones técnicas. ▪ Vocabulario: Documentación de APIs, manuales de usuario, especificaciones técnicas. ▪ Debates: Estándares de documentación de código abierto y reproducibilidad.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 6: Proyectos a Largo Plazo (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presente Perfecto Continuo vs. Presente Perfecto Simple en informes de avance de investigación. ▪ Verbos modales de deducción (Presente y Pasado) para interpretar resultados experimentales. ▪ Vocabulario: Sistemas de largo plazo, programas de investigación sostenidos, desarrollo de startups. ▪ Presentaciones orales: Ïnforme de avance de un proyecto de investigación en computación".	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 7: Redacción Técnica Avanzada (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Oraciones de relativo (Defining vs Non-defining) en la redacción de artículos técnicos. ▪ Voz pasiva avanzada en prosa científica y de ingeniería formal. ▪ Vocabulario: Revisión de pares, factor de impacto, prácticas de citación, acceso abierto. ▪ Redacción: Abstract e introducción de un artículo de investigación en computación.	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unidad 8: Inglés para Ingeniería de Software (12 horas)	
Resultados esperados: AG-C03,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Metodologías Ágiles: Scrum, Kanban y XP — terminología en inglés. ▪ QA (Quality Assurance) y Testing: Planes de prueba, reportes de bugs, criterios de aceptación. ▪ Documentación de requerimientos de software: historias de usuario, casos de uso, especificaciones. ▪ Simulación: Reunión técnica de equipo (Sprint planning y retrospectiva).	▪ 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.
Lecturas : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS231. Redes y Comunicación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS231. Redes y Comunicación
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS2S1. Sistemas Operativos. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos de Redes y Comunicaciones (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Importancia de las redes en la informática contemporánea, y desafíos asociados. Ver también: Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad ▪ Organización de Internet (por ejemplo, usuarios, Proveedores de Servicios de Internet, sistemas autónomos, proveedores de contenido, redes de entrega de contenido) ▪ Técnicas de conmutación (por ejemplo, circuito y paquete) ▪ Capas y sus roles (aplicación, transporte, red, enlace de datos y física) ▪ Principios de estratificación (por ejemplo, encapsulación y modelo de reloj de arena). Ver también: Sistemas Digitales y Lógica, Procesamiento Secuencial y Paralelo, Máquinas de Estado ▪ Elementos de red (por ejemplo, routers, switches, hubs, puntos de acceso y hosts) ▪ Conceptos básicos de colas (por ejemplo, relación con latencia, congestión, niveles de servicio, etc.)	▪ Articular la organización de Internet [Articular] ▪ Listar y definir la terminología de red apropiada [Listar/Enumerar] ▪ Describir la estructura en capas de una arquitectura de red típica [Describir] ▪ Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Analizar]
Lecturas : [**Kurose2021**], [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unidad 2: Comunicación de un solo salto (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción a modulación, ancho de banda y medios de comunicación ▪ Codificación y Entramado ▪ Control de Acceso al Medio (MAC) (por ejemplo, acceso aleatorio y acceso programado) ▪ Ethernet y WiFi ▪ Conmutación (por ejemplo, árboles de expansión, VLANs). ▪ Topologías de Red de área Local (por ejemplo, centro de datos, redes de campus).	▪ Describir algunos aspectos básicos de modulación, ancho de banda y medios de comunicación [Describir] ▪ Describir en detalle un protocolo MAC [Describir] ▪ Demostrar comprensión de las compensaciones (trade-offs) de soluciones de codificación y entramado [Demostrar] ▪ Describir detalles de la implementación de Ethernet [Describir] ▪ Describir cómo funciona la conmutación [Describir] ▪ Describir un tipo de topología de LAN [Describir]
Lecturas : [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unidad 3: Aplicaciones en Red (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Esquemas de denominación y direccionamiento (por ejemplo, DNS e Identificadores Uniformes de Recursos) ▪ Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación ▪ Diversidad de demandas de aplicaciones en red (por ejemplo, latencia, ancho de banda y tolerancia a pérdidas). Ver también: Comunicación, Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social, Equidad Digital y Accesibilidad, Impacto Social e IA ▪ Cobertura de protocolos de capa de aplicación (por ejemplo, HTTP) ▪ Interacciones con APIs de TCP, UDP y Socket. Ver también: Programas: Distribución, Programas: Propiedades de Ejecución, Programación GPU, Programas: Mapeos de Implementación, Programas: Paralelismo Declarativo, Programas: Inicio de Actividades	▪ Definir los principios de denominación, direccionamiento y localización de recursos [Definir] ▪ Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar] ▪ Describir los detalles de un protocolo de capa de aplicación [Describir] ▪ Implementar una aplicación simple cliente-servidor basada en sockets [Implementar]
Lecturas : [**Kurose2021**], [**PetersonDavie2022**]	

Unidad 4: Soporte de Fiabilidad (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Entrega no fiable (por ejemplo, UDP) ▪ Principios de fiabilidad (por ejemplo, entrega sin pérdidas, duplicación o fuera de orden). Ver también: Confiabilidad del Sistema ▪ Control de errores (por ejemplo, retransmisión, corrección de errores) ▪ Control de flujo (por ejemplo, parar y esperar, basado en ventana) ▪ Control de congestión (por ejemplo, notificación implícita y explícita de congestión) ▪ TCP y problemas de rendimiento (por ejemplo, Tahoe, Reno, Vegas, Cubic)	▪ Describir el funcionamiento de protocolos de entrega fiable [Describir] ▪ Listar los factores que afectan el rendimiento de los protocolos de entrega fiable [Listar/Enumerar] ▪ Describir algunos problemas de diseño de fiabilidad de TCP [Describir] ▪ Diseñar e implementar un protocolo fiable simple [Diseñar]
Lecturas : [**Kurose2021**], [**Tanenbaum2021**]	

Unidad 5: Enrutamiento y Reenvío (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas y jerarquía de enrutamiento (por ejemplo, intra/inter dominio, centralizado y descentralizado, enrutamiento de origen, circuitos virtuales, QoS) ▪ Métodos de reenvío (por ejemplo, tablas de reenvío y algoritmos de coincidencia) ▪ IP y problemas de escalabilidad (por ejemplo, NAT, CIDR, BGP, diferentes versiones de IP)	▪ Describir varios paradigmas y jerarquías de enrutamiento [Describir] ▪ Describir cómo se reenvían los paquetes en una red IP [Describir] ▪ Describir cómo Internet aborda los desafíos de escalabilidad [Describir]
Lecturas : [**Kurose2021**], [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unidad 6: Seguridad de Red (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción general sobre seguridad (Amenazas, vulnerabilidades y contramedidas). Ver también: Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad, Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad ▪ Amenazas específicas de red y tipos de ataque (por ejemplo, denegación de servicio, suplantación, rastreo y redireccionamiento de tráfico, atacante en el medio, ataques de integridad de mensajes, ataques de enrutamiento, ransomware y análisis de tráfico). Ver también: Seguridad en IA y Hardware, Control de Acceso y Aplicaciones, Principios y Mentalidad de Seguridad, Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad ▪ Contramedidas: • Criptografía (por ejemplo, SSL, TLS, simétrica/asimétrica) • 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) • Monitoreo de red, detección de intrusiones, firewalls, protección contra suplantación y DoS, honeypots, trazabilidad, BGP Sec, RPKI	▪ Describir algunos de los modelos de amenaza de seguridad de red [Describir] ▪ Describir contramedidas específicas basadas en red [Describir] ▪ Analizar varios aspectos de seguridad de red a partir de un estudio de caso [Analizar]
Lecturas : [**Stallings2022**], [**Kurose2021**]	

Unidad 7: Soporte de Movilidad (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de comunicación celular (por ejemplo, 4G, 5G) ▪ Principios de LANs inalámbricas (principalmente 802.11) ▪ Comunicación dispositivo a dispositivo (por ejemplo, comunicación IoT) ▪ Redes inalámbricas multiescalón (por ejemplo, redes ad hoc, oportunistas, tolerantes a retrasos)	▪ Describir algunos aspectos de comunicación celular como el registro [Describir] ▪ Describir cómo 802.11 soporta usuarios móviles [Describir] ▪ Describir usos prácticos de comunicación dispositivo a dispositivo, así como multiescalón [Describir] ▪ Describir un tipo de red móvil como ad hoc [Describir]
Lecturas : [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unidad 8: Temas Emergentes (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Middleboxes (por ejemplo, avances en el uso de IA, redes basadas en intención, filtrado, inspección profunda de paquetes, balanceo de carga, NAT, CDN) ▪ Virtualización de Red (por ejemplo, SDN, Redes de Centros de Datos) ▪ Redes Cuánticas (por ejemplo, Introducción al dominio, teleportación, seguridad, Internet Cuántica) ▪ Satélite, mmWave, Luz Visible	▪ Describir el valor de los avances en middleboxes en redes [Describir] ▪ Describir la importancia de las Redes Definidas por Software [Describir] ▪ Describir algunos de los valores añadidos logrados al usar Redes Cuánticas [Describir]
Lecturas : [**PetersonDavie2022**], [**Tanenbaum2021**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS311. Algoritmos para Problemas Complejos (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS311. Algoritmos para Problemas Complejos
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS212. Análisis y Diseño de Algoritmos. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Estrategias Algorítmicas (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Búsqueda Completa: Retroceso Recursivo y Máscaras de bits. ▪ Estrategias Voraces: Propiedad de elección y subestructura óptima. ▪ Divide y Vencerás: Búsqueda Binaria sobre la respuesta y Búsqueda Ternaria. ▪ Programación Dinámica: Reducción de estado, DP de dígitos y DP en árboles.	▪ Aplicar técnicas de poda de búsqueda para resolver problemas NP-difíciles con restricciones pequeñas [Usar] ▪ Identificar el paradigma más eficiente para un enunciado de problema dado [Evaluar]
Lecturas : [**Halim2020**], [**Laaksonen2020**]	

Unidad 2: Algoritmos Fundamentales (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Árboles de Fenwick (Árboles Binarios Indexados). ▪ Árboles de Segmentos: Propagación Diferida y Árboles de Segmentos Persistentes. ▪ Descomposición de Raíz Cuadrada y Algoritmo de Mo. ▪ Tablas Dispersas para RMQ.	▪ Implementar estructuras de consulta de rango para actualizaciones de datos en tiempo real [Evaluar] ▪ Analizar las compensaciones entre diferentes estructuras de datos de consulta de rango [Usar]
Lecturas : [**Halim2020**], [**Skiena2020**]	

Unidad 3: Algoritmos Fundamentales (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Flujo en Redes: Flujo Máximo Corte Mínimo (Dinic y Edmonds-Karp). ▪ Componentes Fuertemente Conexas (Tarjan y Kosaraju). ▪ Emparejamiento Bipartito y Descomposición Pesado-Ligero. ▪ Descomposición por Centroide en árboles.	▪ Para cada algoritmo en esta unidad explicar paso a paso cómo opera el algoritmo [Explicar] ▪ Para cada enfoque algorítmico (por ejemplo, ordenamiento) en esta unidad aplicar un ejemplo prototípico del enfoque (por ejemplo, ordenamiento por mezcla) [Aplicar] ▪ 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] ▪ 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] ▪ Para cada uno de los algoritmos y enfoques algorítmicos en los temas del Núcleo KA: • Explicar un ejemplo prototípico del algoritmo, y • Explicar paso a paso cómo opera el algoritmo. [Explicar]
Lecturas : [**Laaksonen2020**], [**Halim2020**]	

Unidad 4: Algoritmos Fundamentales (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Teoría de Números: Criba de Eratóstenes, Euclides Extendido e Inverso Modular. ▪ Combinatoria: Inclusión-Exclusión y Lema de Burnside. ▪ Geometría Computacional: Producto Cruz, Envolvente Convexa (Cadena Monótona) e Intersección de líneas. ▪ Transformada Rápida de Fourier (FFT) para multiplicación de polinomios.	▪ Resolver problemas combinatorios usando aritmética modular [Usar] ▪ Implementar primitivas geométricas para análisis de relaciones espaciales [Evaluar]
Lecturas : [**Skiena2020**], [**Laaksonen2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS312. Estructuras de Datos Avanzadas (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS312. Estructuras de Datos Avanzadas
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS212. Análisis y Diseño de Algoritmos. (5to Sem)

3. PROFESORES

Titular

- Ernesto Cuadros-Vargas <ecuadrosv@uni.edu.pe>
  - Doctor en Ciencia de la Computación, ICMC-USP, Brasil, 2004.
  - Master en Ciencia de la Computación, ICMC-USP, Brasil, 1998.

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Los algoritmos y estructuras de datos son una parte fundamental de la Ciencia de la Computación que nos permite organizar información de manera eficiente. Para cualquier profesional en el campo, una base sólida en esta área es crucial. Este curso se enfoca en estructuras avanzadas como Métodos de Acceso Multidimensional, Métodos de Acceso Espacio-Temporales y Métodos de Acceso Métrico, que son esenciales para aplicaciones de alto rendimiento en motores de búsqueda, sistemas de información geográfica (SIG) y procesamiento de big data.

5. OBJETIVOS

- Comprender, diseñar e implementar estructuras de datos innovadoras para datos multidimensionales.
- Aplicar técnicas de búsqueda por similitud e indexación métrica a tipos de datos complejos.
- Analizar la ”Maldición de la Dimensionalidadz su impacto en la eficiencia de búsqueda.
- Investigar y presentar métodos de indexación contemporáneos para Big Data en inglés.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Programación Orientada a Objetos: Encapsulación, Subtipado y Reflexión (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Técnicas de implementación estáticas y dinámicas. ▪ Abstracción y encapsulación de datos. ▪ Tipos Abstractos de Datos (TAD). ▪ Polimorfismo en tiempo de compilación vs. en tiempo de ejecución. ▪ Estructuras de datos genéricas y Plantillas de C++. ▪ Componentes reutilizables y patrones de diseño para estructuras de datos.	▪ Evaluar el impacto de las técnicas estáticas y dinámicas en el diseño de estructuras de datos [Evaluar]. ▪ Implementar estructuras de datos genéricas y reutilizables usando paradigmas de programación avanzados [Usar]. ▪ Analizar las compensaciones de rendimiento entre diferentes estrategias de implementación [Evaluar].
Lecturas : [**Cuadros2004Implementing**], [**Knuth2007TAOCP-V-I**], [**Gamma94**], [**VanJosGreg:Templates:2018**]	

Unidad 2: Métodos de acceso multidimensional (Parte I) (8 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción a datos multidimensionales. ▪ Representación en espacios vectoriales. ▪ La Maldición de la Dimensionalidad: fundamentos teóricos. ▪ Impacto de la alta dimensionalidad en la indexación tradicional. ▪ Aplicaciones en el mundo real en motores de búsqueda y reconocimiento de patrones.	▪ Explicar la importancia de la representación de datos multidimensionales [Familiarizarse]. ▪ Analizar la complejidad y degradación del rendimiento en espacios de alta dimensión [Usar]. ▪ Discutir escenarios del mundo real donde el impacto de la dimensionalidad es crítico [Usar].
Lecturas : [**Gaede98multidimensional**], [**Samet2006SAM-MAM**]	

Unidad 3: Métodos de acceso multidimensional (Parte II) (20 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructuras de datos espaciales: Quadtree y Octree. ▪ Métodos de Acceso a Puntos: Kd-Tree. ▪ Métodos de Acceso a Regiones: R-Tree (Guttman), R+ tree, R* tree. ▪ Variaciones de R-trees y su relación con la paginación y el tamaño de bloque. ▪ Estructuras avanzadas: X-tree y SS-tree.	▪ Implementar estructuras de datos espaciales para indexación de datos de alto volumen [Usar]. ▪ Evaluar las limitaciones de las estructuras espaciales basadas en árboles en dominios específicos [Evaluar]. ▪ Implementar búsqueda por rango y estrategias de Vecino Más Cercano (k-NN) [Usar].
Lecturas : [**Samet2006SAM-MAM**]	

Unidad 4: Métodos de acceso métrico (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Espacios Métricos: definiciones y propiedades. ▪ Búsqueda por similitud vs. Búsqueda exacta. ▪ Árbol de Punto de Vista (VP-Tree). ▪ Slim-Tree y M-Tree. ▪ Indexación basada en pivotes.	▪ Modelar problemas de búsqueda por similitud usando propiedades de espacios métricos [Usar]. ▪ Comparar métodos de acceso métrico contra métodos multidimensionales para datos no vectoriales [Evaluar].
Lecturas : [**Samet2006SAM-MAM**], [**Zezula07**]	

Unidad 5: Comunicación Técnica y Profesional (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Seminario sobre artículos de investigación recientes (SIGMOD, VLDB, ICDE). ▪ Indexación para Big Data y datos en flujo. ▪ Estructuras de datos compactas y representaciones sucintas. ▪ Tendencias futuras en búsqueda por similitud.	▪ Investigar nuevos métodos para indexar grandes volúmenes de datos complejos [Usar]. ▪ Presentar y dirigir discusiones técnicas en inglés sobre métodos de vanguardia [Evaluar]. ▪ Identificar posibles temas de tesis dentro del dominio de estructuras de datos avanzadas [Familiarizarse].
Lecturas : [**Navarro:CompactDS:2016**], [**Traina00SlimTree**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS342. Compiladores (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS342. Compiladores
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS211. Teoría de la Computación. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso proporciona una comprensión profunda de cómo los lenguajes de alto nivel se transforman en código máquina ejecutable. El estudio de los compiladores permite a los estudiantes comprender la semántica de los lenguajes, la gestión de memoria y la optimización de código, que son habilidades fundamentales para diseñar lenguajes específicos de dominio y realizar ingeniería de software de alto rendimiento.

5. OBJETIVOS

- Comprender las fases de un compilador y las herramientas de construcción.
- Implementar analizadores léxicos y sintácticos basados en gramáticas.
- Generar y optimizar código intermedio para una arquitectura específica.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Análisis de Programas y Analizadores (20 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Representaciones de programa relevantes, como bloques básicos (basic blocks), grafos de flujo de control (control-flow graphs), cadenas definición-uso (def-use chains) y asignación única estática (static single assignment) ▪ Indecidibilidad y consecuencias para el análisis de programas ▪ Análisis insensible al flujo (flow-insensitive), como verificación de tipos y análisis de punteros y alias escalables ▪ Análisis sensible al flujo (flow-sensitive), como análisis de flujo de datos (dataflow) hacia adelante y hacia atrás ▪ Análisis sensible a la ruta (path-sensitive), como verificación de modelos de software (software model checking) y verificación de software ▪ Herramientas y frameworks para implementar analizadores ▪ Papel del análisis estático en la optimización de programas y el análisis de dependencia de datos durante la explotación de concurrencia ▪ Papel del análisis de programas en la verificación (parcial) y la búsqueda de errores (bug-finding) ▪ Paralelización: • Análisis para auto-paralelización • Análisis para detectar errores de concurrencia	▪ Explicar la diferencia entre grafo de flujo de datos (dataflow graph) y grafo de flujo de control [Explicar] ▪ Explicar por qué los análisis de programa no triviales y sólidos (sound) deben ser aproximados [Explicar] ▪ Argumentar por qué un análisis es correcto (sound y terminante) [Argumentar] ▪ Explicar por qué el alias potencial limita el análisis de programa sólido y cómo el análisis de alias puede ayudar [Explicar] ▪ Usar los resultados de un análisis de programa para optimización de programas y/o corrección parcial del programa [Usar]
Lecturas : [**Aho2006**], [**Appel2004**]	

Unidad 2: Traducción y Ejecución de Lenguajes (24 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de ejecución para JIT (Just-In-Time), compilador, intérprete ▪ Uso de código intermedio, por ejemplo, bytecode ▪ Limitaciones y beneficios de JIT, compilador e intérprete ▪ Compiladores/transpiladores cruzados (cross compilers/transpilers) ▪ Representación BNF y BNF extendida de gramática libre de contexto ▪ árbol de análisis (parse tree) usando una oración simple como expresión aritmética o sentencia if-then-else ▪ Ejecución como código nativo o dentro de una máquina virtual ▪ Canalización de traducción de lenguaje: análisis sintáctico, análisis (parsing), verificación de tipos opcional, generación de código/traducción y optimización, enlace (linking), carga (loading), ejecución ▪ Representación en tiempo de ejecución de construcciones centrales del lenguaje como objetos (tablas de métodos) y funciones que pueden pasarse como parámetros a y devolverse desde funciones (clausuras) ▪ Desarrollo seguro de compiladores	▪ Explicar y comprender las diferencias entre implementaciones de lenguaje compilado, JIT e interpretado, incluyendo los beneficios y limitaciones de cada una [Explicar] ▪ Diferenciar sintaxis y análisis (parsing) de semántica y evaluación [Diferenciar] ▪ Usar BNF y BNF extendida para especificar la sintaxis de construcciones simples como if-then-else, declaración de tipo y construcciones iterativas para lenguajes conocidos como C++ o Python [Usar] ▪ Ilustrar el árbol de análisis usando una oración/expresión aritmética simple [Aplicar] ▪ Ilustrar traducción de diagramas de sintaxis a BNF/BNF extendida para construcciones simples como if-then-else, declaración de tipo, construcciones iterativas, etc [Aplicar] ▪ Ilustrar ambigüedad en el análisis usando if-then-else anidado/expresión aritmética y mostrar resolución usando orden de precedencia [Aplicar] ▪ Discutir los beneficios y limitaciones de la recolección de basura, incluyendo la noción de alcanzabilidad [Debatir]
Lecturas : [**Aho2006**], [**Appel2004**]	

Unidad 3: Comportamiento en Tiempo de Ejecución y Sistemas (20 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de proceso usando pilas y montones para asignar y desasignar registros de activación y recuperar entornos usando punteros de marco (frame pointers) y direcciones de retorno durante una llamada a procedimiento, incluyendo ejemplos de paso de parámetros ▪ Esquemas de búsqueda de código usando tablas hash para métodos en implementaciones de programas orientados a objetos ▪ Distribución de datos para objetos y registros de activación ▪ Asignación de objetos en el montón (heap) ▪ Implementar entidades virtuales y métodos virtuales; tablas de métodos virtuales (virtual method tables) y su aplicación ▪ Comportamiento en tiempo de ejecución de programas orientados a objetos ▪ Comparar y contrastar la asignación de memoria durante el intercambio de información usando paso de parámetros y variables no locales (usando cadena de enlaces estáticos). ▪ Enfoques y técnicas de gestión dinámica de memoria: malloc/free, recolección de basura (mark-sweep, copying, reference counting), regiones (también conocidas como arenas o zonas) ▪ Compilación justo a tiempo (just-in-time) y recompilación dinámica ▪ Interfaz con el sistema operativo (por ejemplo, para inicialización del programa) ▪ Interoperabilidad entre lenguajes de programación incluyendo mecanismos de paso de parámetros y representación de datos: • *Big endian*, *little endian* • Distribución de datos de tipos de datos compuestos como arreglos ▪ Otras características comunes de las máquinas virtuales, como carga de clases (class loading), hilos (threads) y verificación de seguridad ▪ Sandboxing	▪ Discutir beneficios y limitaciones de la gestión automática de memoria [Debatir] ▪ Explicar el uso de metadatos en representaciones en tiempo de ejecución de objetos y registros de activación, como punteros de clase, longitudes de arreglo, direcciones de retorno y punteros de marco (frame pointers) [Explicar] ▪ Comparar y contrastar asignación estática vs asignación basada en pila vs asignación basada en montón de elementos de datos [Comparar] ▪ Explicar por qué algunos elementos de datos no pueden desasignarse automáticamente al final de una llamada a procedimiento/método (necesidad de recolección de basura) [Explicar] ▪ Discutir ventajas, desventajas y dificultades de la compilación justo a tiempo y la recompilación dinámica [Debatir] ▪ Discutir el uso del sandboxing en código móvil [Debatir] ▪ Identificar los servicios proporcionados por los sistemas en tiempo de ejecución de lenguajes modernos [Analizar]
Lecturas : [**Aho2006**], [**Cooper2011**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI263. Introducción al Aprendizaje de Máquina (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI263. Introducción al Aprendizaje de Máquina
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS261. Inteligencia Artificial. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- Comprender los principios matemáticos detrás de algoritmos de ML.
- Implementar pipelines completos de ML con Python.
- Evaluar y optimizar modelos usando métricas estándar.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos de ML (15 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aprendizaje supervisado vs no supervisado ▪ Overfitting y regularización ▪ Validación cruzada y curvas de aprendizaje	▪ Explicar el bias-variance tradeoff [Familiarizarse] ▪ Implementar k-fold cross-validation [Usar]
Lecturas : [**Bishop2006**], [**Goodfellow2016**]	

Unidad 2: Modelos Lineales (15 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Regresión lineal y logística ▪ Máquinas de vectores de soporte (SVM) ▪ Análisis discriminante lineal	▪ Programar modelos lineales con scikit-learn [Usar] ▪ Interpretar coeficientes de regresión [Evaluar]
Lecturas : [**Hastie2009**], [**Murphy2012**]	

Unidad 3: Ensambles y Árboles (15 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Árboles de decisión y random forests ▪ Gradient Boosting (XGBoost, LightGBM) ▪ Stacking y votación	▪ Optimizar hiperparámetros con GridSearch [Usar] ▪ Visualizar árboles de decisión [Evaluar]
Lecturas : [**Hastie2009**], [**Geron2022**]	

Unidad 4: Redes Neuronales Básicas (15 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Perceptrones multicapa (MLP) ▪ Backpropagation y optimizadores ▪ Introducción a Keras/TensorFlow	▪ Construir redes neuronales simples [Usar] ▪ Monitorizar entrenamiento con TensorBoard [Evaluar]
Lecturas : [**Goodfellow2016**], [**Chollet2021**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online

permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

FI201. Física Computacional (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	FI201. Física Computacional
2.2 Semestre	:	6to Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	BFI101. Física I. (2do Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso aplica los principios de la física a problemas computacionales, con énfasis en la luz, la propagación de ondas, colisiones y la transferencia de energía. Estos conceptos son esenciales en áreas como gráficos por computadora, simulaciones físicas y desarrollo de videojuegos.

5. OBJETIVOS

- Comprender los principios físicos relevantes para la computación.
- Aplicar estos principios para resolver problemas computacionales específicos.
- Implementar algoritmos basados en la física para simulaciones y gráficos por computadora.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Óptica y Propagación de la Luz (10 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Naturaleza de la luz. ▪ Reflexión y refracción. ▪ Lentes y espejos. ▪ Interferencia y difracción. ▪ Modelos de iluminación (ej. Phong, Blinn-Phong).	▪ Describir las propiedades de la luz y su propagación. [Familiarizarse] ▪ Aplicar las leyes de la reflexión y refracción. [Usar] ▪ Implementar modelos de iluminación en gráficos por computadora. [Evaluar]
Lecturas : [**YoungFreedman2018**], [**Hecht2017**]	

Unidad 2: Colisiones y Transferencia de Energía (8 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Impulso y momento lineal. ▪ Colisiones elásticas e inelásticas. ▪ Conservación de la energía en colisiones. ▪ Deformación de mallas elásticas (ej. modelo de masa-resorte).	▪ Aplicar los principios de conservación del momento lineal y la energía en colisiones. [Familiarizarse] ▪ Modelar la deformación de mallas elásticas producto de un impacto. [Usar] ▪ Implementar simulaciones de colisiones en un entorno computacional. [Evaluar]
Lecturas : [**YoungFreedman2018**], [**Taylor2005**]	

Unidad 3: Mecánica de Sólidos Rígidos (8 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Rotación de cuerpos rígidos. ▪ Momento de inercia. ▪ Torque y energía cinética rotacional.	▪ Describir la rotación de cuerpos rígidos. [Familiarizarse] ▪ Calcular el momento de inercia. [Usar] ▪ Aplicar las ecuaciones de la dinámica rotacional. [Evaluar]
Lecturas : [**YoungFreedman2018**], [**Taylor2005**]	

Unidad 4: Dinámica de Fluidos (6 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios básicos de la dinámica de fluidos. ▪ Viscosidad. ▪ Flujo laminar y turbulento.	▪ Describir las propiedades de los fluidos. [Familiarizarse] ▪ Explicar los conceptos de viscosidad y flujo laminar/turbulento. [Usar] ▪ Resolver problemas sencillos de dinámica de fluidos. [Evaluar]
Lecturas : [**YoungFreedman2018**]	

Unidad 5: Termodinámica (6 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Leyes de la termodinámica. ▪ Transferencia de calor.	▪ Enunciar las leyes de la termodinámica. [Familiarizarse] ▪ Describir los mecanismos de transferencia de calor. [Usar]
Lecturas : [**YoungFreedman2018**]	

Unidad 6: Simulación Física (10 horas)	
Resultados esperados: AG-C08,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Métodos numéricos para simulación física. ▪ Integración de Verlet. ▪ Detección de colisiones. ▪ Sistemas de partículas. ▪ Restricciones y solvers.	▪ Implementar métodos numéricos básicos para simulación física. [Familiarizarse] ▪ Utilizar la integración de Verlet para simular el movimiento. [Usar] ▪ Implementar algoritmos de detección de colisiones. [Evaluar]
Lecturas : [**Taylor2005**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS251. Computación Gráfica (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS251. Computación Gráfica
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	MA202. Métodos Numéricos. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

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

7. TEMAS

Unidad 1: Conceptos Fundamentales de Gráficos y Técnicas Interactivas (16 horas)	
Resultados esperados: AG-C07,AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Descripción general del pipeline de gráficos por computadora y sistemas interactivos ▪ Conceptos básicos de renderizado: rasterización, trazado de rayos (*ray tracing*), modelos de sombreado ▪ Teoría del color y espacios de color ▪ Sistemas de coordenadas y transformaciones ▪ Técnicas básicas de interacción: dispositivos de entrada, manejo de eventos ▪ Conceptos avanzados de renderizado: iluminación global (*global illumination*), renderizado basado en física ▪ Hardware gráfico y técnicas de aceleración ▪ Percepción humana y cognición visual	▪ Explicar las etapas básicas del pipeline de gráficos por computadora [Explicar] ▪ Describir técnicas básicas de renderizado como rasterización y trazado de rayos [Describir] ▪ Aplicar teoría del color para diseñar gráficos visualmente efectivos [Aplicar] ▪ Implementar transformaciones de coordenadas para gráficos 2D y 3D [Implementar] ▪ Diseñar aplicaciones interactivas básicas utilizando dispositivos de entrada comunes y manejo de eventos [Diseñar] ▪ Comparar técnicas avanzadas de renderizado como iluminación global y renderizado basado en física [Comparar] ▪ Analizar el rol del hardware gráfico en acelerar tareas de renderizado [Analizar] ▪ Explicar cómo la percepción humana influye en el diseño de gráficos e interacción [Explicar]
Lecturas : [**Marschner2021**], [**Hearn2014**]	

Unidad 2: Renderizado Aplicado y Técnicas (16 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Algoritmos de rasterización: conversión por barrido (*scan conversion*), antialiasing ▪ Trazado de rayos (*ray tracing*): estructuras de aceleración, muestreo (*sampling*), sombreado ▪ Mapeo de texturas (*texture mapping*) y filtrado ▪ Modelos de sombreado: Lambertiano, Phong, BRDF ▪ Técnicas avanzadas: mapeo de sombras (*shadow mapping*), mapeo ambiental (*environment mapping*), renderizado diferido (*deferred rendering*)	▪ Implementar algoritmos de rasterización para renderizado de primitivas [Implementar] ▪ Implementar trazado de rayos básico con estructuras de aceleración [Implementar] ▪ Aplicar técnicas de mapeo de texturas y filtrado para mejorar el realismo visual [Aplicar] ▪ Comparar diferentes modelos de sombreado y sus aplicaciones [Comparar] ▪ Implementar técnicas avanzadas de renderizado como mapeo de sombras o renderizado diferido [Implementar]
Lecturas : [**Marschner2021**], [**Hearn2014**], [**Angel2014**]	

Unidad 3: Modelado Geométrico (16 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Representaciones geométricas: polígonos, curvas, superficies, formas implícitas ▪ Técnicas de modelado: curvas spline, superficies de subdivisión, geometría sólida constructiva ▪ Modelado procedural y generación ▪ Algoritmos geométricos: intersección, distancia, descomposición	▪ Describir diferentes representaciones geométricas y sus compensaciones (trade-offs) [Describir] ▪ Implementar técnicas de modelado como curvas spline o superficies de subdivisión [Implementar] ▪ Crear modelos procedurales usando algoritmos o reglas [Crear] ▪ Aplicar algoritmos geométricos para resolver problemas como pruebas de intersección o procesamiento de mallas (mesh) [Aplicar]
Lecturas : [**Marschner2021**], [**Angel2014**]	

Unidad 4: Visualización (16 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de visualización de datos: percepción, cognición y codificación efectiva ▪ Tipos de visualizaciones: científicas, de información, geográficas ▪ Técnicas de visualización para datos multivariados, temporales y jerárquicos ▪ Visualización interactiva y exploración ▪ Evaluación de la efectividad de la visualización	▪ Aplicar principios de percepción y cognición para diseñar visualizaciones efectivas [Aplicar] ▪ Seleccionar técnicas de visualización apropiadas para diferentes tipos de datos y tareas [Evaluar] ▪ Implementar visualizaciones interactivas para exploración de datos [Implementar] ▪ Evaluar la efectividad de una visualización usando métodos empíricos [Evaluar]
Lecturas : [**Marschner2021**]	

Unidad 5: Sombreado y Renderizado Avanzado (12 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sombreado avanzado: sombreado basado en física, dispersión subsuperficial (subsurface scattering) ▪ Iluminación global (global illumination): radiosidad, mapeo de fotones (photon mapping), trazado de caminos (path tracing) ▪ Técnicas de renderizado en tiempo real: nivel de detalle (level of detail), culling, oclusión ▪ Programación GPU y sombreadores (shaders)	▪ Implementar modelos de sombreado avanzados para materiales realistas [Implementar] ▪ Comparar algoritmos de iluminación global y sus compensaciones [Comparar] ▪ Optimizar el renderizado en tiempo real usando técnicas como nivel de detalle y *culling* [Crear] ▪ Desarrollar sombreadores GPU para efectos de renderizado personalizados [Crear]
Lecturas : [**Marschner2021**], [**Angel2014**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS292. Ingeniería de Software II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS292. Ingeniería de Software II
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS291. Ingeniería de Software I. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Ingeniería de Software II se basa en los principios fundamentales del primer curso, cambiando el enfoque hacia patrones de diseño avanzados, evolución de software y la gestión de proyectos a gran escala. Explora las complejidades de mantener sistemas heredados, asegurar la calidad del software mediante validación rigurosa, y la aplicación de técnicas profesionales de gestión de proyectos en entornos ágiles.

5. OBJETIVOS

- Aplicar patrones de diseño orientados a objetos avanzados para resolver problemas arquitectónicos complejos.
- Dominar la gestión de configuración de software y control de versiones en entornos colaborativos.
- Comprender los principios de evolución y mantenimiento de software.
- Gestionar proyectos de software usando técnicas de estimación y análisis de riesgos.
- Implementar estrategias de prueba avanzadas y métricas de calidad.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Colaboración y Dinámica de Equipo (14 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comunicación efectiva, incluida la oral y escrita, así como formal (correo electrónico, documentos, comentarios, presentaciones) e informal (chat de equipo, reuniones). Comunicación Técnica y Profesional, Comunicación en Equipo y Cultura ▪ Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos. ▪ Programación cooperativa: • Programación por parejas (pair programming) o Enjambre (swarming) • Revisión de código • Colaboración mediante control de versiones ▪ Roles y responsabilidades en un equipo de software: Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo • Ventajas del trabajo en equipo • Riesgos y complejidad de dicha colaboración ▪ Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo	▪ Seguir prácticas efectivas de comunicación en equipo [Analizar] ▪ 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] ▪ Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar] ▪ Colaborar efectivamente en desarrollo/programación cooperativa [Analizar] ▪ Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer] ▪ Redactar y seguir una agenda para una reunión de equipo [Componer] ▪ 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]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 2: Principios y Arquitectura de Software (16 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de sistemas. Confiabilidad del Sistema • Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) • Separación de preocupaciones (separation of concerns) • Ocultación de información (information hiding) • Acoplamiento y cohesión (coupling and cohesion) ▪ Arquitectura de software. Confiabilidad del Sistema • Paradigmas de diseño ◦ Descomposición funcional de arriba hacia abajo/diseño por capas ◦ Arquitectura orientada a datos ◦ Análisis y diseño orientado a objetos ◦ Diseño dirigido por eventos • Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) • Identificar límites y dependencias de componentes ▪ Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema ▪ Malos olores del código (code smells) y otras indicaciones de la calidad del código, distintas de la corrección. Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad	▪ Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] ▪ 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] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] ▪ Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]
Lecturas : [**Gamma1994**], [**Pressman2019**]	

Unidad 3: Prácticas de Codificación (14 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas prácticas a pequeña escala Prácticas de Desarrollo de Software • Pruebas unitarias • Desarrollo guiado por pruebas (Test-driven development): esto es particularmente valioso para los estudiantes psicológicamente, ya que es mucho más fácil involucrarse constructivamente con el desafío de identificar entradas desafiantes para una API dada (casos límite, casos extremos) a priori. Si implementan primero, el instinto es a menudo evitar intentar hacer fallar su nueva creación, mientras que un enfoque de pruebas primero les da la satisfacción intelectual de detectar los casos problemáticos y luego ver cómo pasan más pruebas durante el proceso de desarrollo. ▪ Documentación Prácticas de Desarrollo de Software • Documentación de la interfaz: describir los requisitos de la interfaz, potencialmente incluyendo contratos (formales o informales), pre y post condiciones, invariantes. • 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.) ◦ Aclarar el flujo de datos, cómputo, etc., centrándose en lo que el código es. ◦ Identificar partes sutiles/complicadas del código y refactorizar para que sean autoexplicativas si es posible o proporcionar comentarios apropiados para aclarar. ▪ Estilo de codificación Prácticas de Desarrollo de Software • Guías de estilo • Comentarios • Nomenclatura (*naming*) ▪ "Mejores Prácticas" para la codificación: técnicas, patrones/idiomas, mecanismos para construir programas de calidad Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada • Prácticas de codificación defensiva • Prácticas y principios de codificación segura • Usar mecanismos de manejo de excepciones para hacer los programas más robustos, tolerantes a fallos	▪ Escribir pruebas unitarias apropiadas para un componente pequeño (varias funciones, un solo tipo, etc.) [Escribir] ▪ Escribir comentarios apropiados de interfaz y (si es necesario) de implementación para un componente pequeño [Escribir] ▪ Describir técnicas, patrones de codificación y mecanismos para implementar diseños para lograr propiedades deseadas como confiabilidad, eficiencia y robustez [Describir] ▪ Escribir código robusto utilizando mecanismos de manejo de excepciones [Escribir] ▪ Describir prácticas de codificación segura y defensiva [Describir] ▪ Seleccionar y usar un estándar de codificación definido en un proyecto de software pequeño [Usar] ▪ Comparar y contrastar estrategias de integración, incluyendo integración de arriba hacia abajo (top-down), de abajo hacia arriba (bottom-up) y sándwich [Comparar] ▪ Describir el proceso de analizar e implementar cambios en la base de código desarrollada para un proyecto específico [Describir] ▪ Describir el proceso de analizar e implementar cambios en una base de código existente grande [Describir]

Unidad 4: Conceptos de verificación y validación (10 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conceptos de verificación y validación • Verificación: ¿Estamos construyendo la cosa correctamente? • Validación: ¿Construimos la cosa correcta? ▪ Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? ▪ Objetivos de las pruebas • Usabilidad • Confiabilidad • Conformidad con la especificación • Rendimiento • Seguridad	▪ Explicar por qué las pruebas son importantes [Explicar] ▪ Distinguir entre validación y verificación de programas [Distinguir] ▪ Describir diferentes objetivos de las pruebas [Describir]
Lecturas : [**Pressman2019**]	

Unidad 5: Herramientas y Análisis de Pruebas (4 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas y automatización de verificación y validación • Análisis estático • Cobertura de código • Fuzzing • Análisis dinámico y contención de fallos (sanitizadores, etc.) • Registro y seguimiento de fallos ▪ Planificación y generación de pruebas • Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos • Uso de números aleatorios y pseudoaleatorios en las pruebas ▪ Pruebas de sistemas asíncronos, paralelos y concurrentes ▪ Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) ▪ Desarrollo guiado por comportamiento (BDD) • Expresar criterios de aceptación como escenarios Gherkin Given/When/Then • Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo ▪ Pruebas de mutación y análisis estático de calidad de código • Pruebas de mutación para evaluar la efectividad de una suite de pruebas • Detección estática de *code smells* y complejidad ciclomática excesiva	▪ Describir y comparar diferentes herramientas para verificación y validación [Describir] ▪ Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] ▪ Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] ▪ Describir enfoques para la estimación de fallos [Describir] ▪ Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] ▪ Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] ▪ 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] ▪ Describir técnicas para la verificación y validación de artefactos que no son código [Describir] ▪ Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] ▪ Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 6: Pruebas de Rendimiento y Evaluación Comparativa (4 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas de rendimiento y evaluación comparativa • Rendimiento (*throughput*) y latencia • Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) • Aceleración (*speedup*) y escalado (*scaling*) ◦ Ley de Amdahl ◦ Ley de Gustafson ◦ Escalado suave (*soft*) y débil (*weak*) • Identificar y medir figuras de mérito (*figures of merit*) • Cuellos de botella de rendimiento comunes ◦ Limitado por cómputo (*compute-bound*) ◦ Limitado por ancho de banda de memoria ◦ Limitado por latencia • Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) ◦ Estimación de la incertidumbre ◦ Intervalos de confianza • Análisis y presentación (gráficos, etc.) • Técnicas de medición de tiempo (*timing*)	▪ Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] ▪ Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] ▪ Describir cuellos de botella de rendimiento comunes [Describir] ▪ Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] ▪ Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] ▪ 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]
Lecturas : [**Bondi2015**], [**Gregg2020**]	

Unidad 7: Conceptos de Confiabilidad (10 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Concepto de confiabilidad como probabilidad de fallo o tiempo medio entre fallos, y fallas (faults) como causa de fallos (failures) ▪ Identificar requisitos de confiabilidad para diferentes tipos de software ▪ Fallos de software causados por defectos/errores (bugs), por lo que para alta confiabilidad el objetivo es tener un mínimo de defectos - inyectando menos defectos (mejor capacitación, educación, planificación) y eliminando la mayoría de los defectos inyectados (pruebas, revisión de código, etc.) ▪ Confiabilidad del software, confiabilidad del sistema y comportamiento de fallos ▪ Ciclo de inyección y eliminación de defectos, y diferentes enfoques para la eliminación de defectos ▪ Comparar el enfoque de "presupuesto de error" (error budget) para la confiabilidad con el enfoque "libre de errores" (error-free) e identificar dominios donde cada uno es relevante.	▪ Describir cómo determinar el nivel de confiabilidad requerido por un sistema de software [Describir] ▪ Explicar los problemas que existen para lograr niveles muy altos de confiabilidad [Explicar] ▪ Comprender enfoques para minimizar fallas que pueden aplicarse en cada etapa del ciclo de vida del software [Explicar]
Lecturas : [**Sommerville2015**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS2H1. Experiencia de Usuario (UX) (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS2H1. Experiencia de Usuario (UX)
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS291. Ingeniería de Software I. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Comprender al Usuario: Objetivos individuales e interacciones con otros (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Capacidades y limitaciones humanas: percepción, cognición, habilidades motoras ▪ Características del usuario: demografía, experiencia, contexto cultural ▪ Objetivos, tareas y contextos de uso del usuario ▪ Aspectos sociales y organizacionales de la interacción ▪ Modelos y teorías cognitivas relevantes para HCI ▪ Aspectos emocionales y afectivos de la interacción ▪ Métodos de investigación de usuarios: entrevistas, encuestas, observaciones	▪ Describir las capacidades y limitaciones humanas que afectan el diseño de interacción [Describir] ▪ Analizar cómo las características del usuario influyen en la interacción con sistemas informáticos [Analizar] ▪ Identificar objetivos y tareas del usuario en contextos de uso específicos [Analizar] ▪ Explicar factores sociales y organizacionales que impactan la adopción y uso de la tecnología [Explicar] ▪ Aplicar modelos cognitivos para predecir y explicar el comportamiento del usuario [Aplicar] ▪ Considerar factores emocionales y afectivos en el diseño de interacción [Analizar] ▪ Realizar investigación de usuarios usando métodos apropiados como entrevistas u observaciones [Evaluar]
Lecturas : [**Norman2013**], [**Johnson2020**]	

Unidad 2: Responsabilidad y Rendición de Cuentas en el Diseño (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Consideraciones éticas en HCI: equidad, transparencia, rendición de cuentas ▪ Diseño para valores y diseño sensible a valores ▪ Innovación responsable y ética del diseño ▪ Rendición de cuentas algorítmica y sesgo en sistemas interactivos ▪ Responsabilidad profesional en la práctica de HCI ▪ Consideraciones legales y regulatorias para sistemas interactivos	▪ Identificar problemas éticos en el diseño e implementación de HCI [Analizar] ▪ Aplicar enfoques de diseño sensible a valores al desarrollo tecnológico [Aplicar] ▪ Discutir prácticas de innovación responsable en HCI [Debatir] ▪ Analizar la rendición de cuentas algorítmica y el sesgo en sistemas interactivos [Analizar] ▪ Describir responsabilidades profesionales en la práctica de HCI [Describir] ▪ Explicar consideraciones legales y regulatorias relevantes para sistemas interactivos [Explicar]
Lecturas : [**Shneiderman2022**], [**Norman2013**]	

Unidad 3: Accesibilidad y Diseño Inclusivo (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios y pautas de accesibilidad ▪ Discapacidades y tecnologías de asistencia ▪ Diseño inclusivo y principios de diseño universal ▪ Métodos de evaluación para accesibilidad ▪ Diseñar para diversas capacidades y contextos ▪ Estándares de accesibilidad y cumplimiento	▪ Aplicar principios de accesibilidad al diseño de sistemas interactivos [Aplicar] ▪ Describir cómo diferentes discapacidades afectan la interacción con la tecnología [Describir] ▪ Diseñar sistemas inclusivos usando principios de diseño universal [Diseñar] ▪ Realizar evaluaciones de accesibilidad de sistemas interactivos [Evaluar] ▪ Crear diseños que acomoden diversas capacidades y contextos de uso [Crear] ▪ Garantizar el cumplimiento con estándares y regulaciones de accesibilidad [Evaluar]
Lecturas : [**Johnson2020**]	

Unidad 4: Evaluación del Diseño (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios y heurísticas de usabilidad ▪ Métodos de evaluación: evaluación heurística, recorrido cognitivo (cognitive walkthrough) ▪ Pruebas con usuarios: planificación, ejecución, análisis ▪ Métodos avanzados de evaluación: experimentos controlados, estudios de campo ▪ Métricas y medición de UX ▪ Métodos de evaluación analítica: modelado y simulación	▪ Aplicar principios y heurísticas de usabilidad para evaluar diseños [Aplicar] ▪ Realizar evaluaciones heurísticas y recorridos cognitivos [Evaluar] ▪ Planificar, conducir y analizar sesiones de pruebas con usuarios [Planear] ▪ Diseñar y conducir experimentos controlados para investigación en HCI [Diseñar] ▪ Medir la experiencia de usuario usando métricas apropiadas [Evaluar] ▪ Usar métodos analíticos para modelar y predecir el desempeño del usuario [Usar]
Lecturas : [**Norman2013**], [**Johnson2020**]	

Unidad 5: Diseño de Sistemas (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Proceso de diseño: diseño centrado en el usuario, diseño iterativo ▪ Diseño de interacción: conceptos, principios, patrones ▪ Técnicas de prototipado: baja fidelidad, alta fidelidad, interactivo ▪ Arquitectura de información y diseño de navegación ▪ Técnicas avanzadas de interacción: gestos, voz, multimodal ▪ Diseño para dominios específicos: móvil, web, computación ubicua ▪ Sistemas de computación colaborativa y social	▪ Aplicar procesos de diseño centrado en el usuario para desarrollar sistemas interactivos [Aplicar] ▪ Diseñar interacciones usando principios y patrones establecidos [Diseñar] ▪ Crear prototipos en diferentes niveles de fidelidad para exploración de diseño [Crear] ▪ Estructurar información y navegación para una interacción efectiva del usuario [Estructurar] ▪ Implementar técnicas avanzadas de interacción como interfaces de gestos o voz [Implementar] ▪ Diseñar sistemas interactivos para dominios y plataformas específicos [Diseñar] ▪ Diseñar sistemas colaborativos que apoyen la interacción grupal [Diseñar]
Lecturas : [**Johnson2020**], [**Norman2013**]	

Unidad 6: Sociedad, ética y Profesión (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Impacto social de las tecnologías interactivas ▪ Problemas éticos en la investigación y práctica de HCI ▪ Práctica profesional en HCI ▪ Consideraciones culturales e interculturales en HCI ▪ Sostenibilidad y HCI ▪ Tendencias futuras y desafíos en HCI	▪ Analizar el impacto social de las tecnologías interactivas en individuos y sociedad [Analizar] ▪ Abordar problemas éticos en proyectos de investigación y diseño de HCI [Evaluar] ▪ Describir prácticas profesionales y trayectorias profesionales en HCI [Describir] ▪ Diseñar sistemas que tomen en cuenta diferencias y contextos culturales [Diseñar] ▪ Incorporar consideraciones de sostenibilidad en el diseño de HCI [Integrar] ▪ Discutir tendencias emergentes y desafíos futuros en HCI [Debatir]
Lecturas : [**Shneiderman2022**], [**Norman2013**]	

Unidad 7: Agentes y Sistemas Cognitivos (8 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitecturas de agentes (por ejemplo, reactivas, en capas, cognitivas) ▪ Teoría de agentes (incluyendo formalismos matemáticos) ▪ Racionalidad, Teoría de juegos: • Agentes decisorios-teóricos • Procesos de decisión de Markov (MDP) • Algoritmos de bandidos ▪ Agentes de software, asistentes personales y acceso a información: • Agentes colaborativos • Agentes de recolección de información • Agentes creíbles (personajes sintéticos, modelado de emociones en agentes) ▪ Agentes de aprendizaje ▪ Sistemas cognitivos: • Arquitecturas cognitivas (por ejemplo, ACT-R, SOAR, ICARUS, FORR) • Capacidades (por ejemplo, percepción, toma de decisiones, predicción, mantenimiento del conocimiento) • Representación, organización, utilización, adquisición y refinamiento del conocimiento • Aplicaciones y evaluación de sistemas cognitivos ▪ Sistemas multiagente: • Agentes colaboradores • Equipos de agentes • Agentes competitivos (por ejemplo, subastas, votación) • Sistemas de enjambre y modelos biológicamente inspirados • Aprendizaje multiagente ▪ Interacción humano-agente: • Metodologías de comunicación (verbal y no verbal) • Problemas prácticos • Aplicaciones: ◦ Agentes comerciales, gestión de cadena de suministro ◦ Problemas éticos de las interacciones de IA con humanos ◦ Regulación y requisitos legales de los sistemas de IA para interactuar con humanos	▪ Caracterizar y contrastar las arquitecturas estándar de agentes [Evaluar] ▪ 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] ▪ Describir los paradigmas principales utilizados por los agentes de aprendizaje [Describir] ▪ Demostrar usando ejemplos apropiados cómo los sistemas multiagente soportan la interacción de agentes [Demostrar] ▪ Construir un agente inteligente usando una arquitectura cognitiva establecida (ACT-R, SOAR) para resolver un problema específico [Aplicar]

Lecturas : [RussellNorvig2021], [Shneiderman2022]

Unidad 8: Procesamiento del Lenguaje Natural (8 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gramáticas deterministas y estocásticas ▪ Algoritmos de análisis sintáctico: • CFGs y analizadores de tabla (por ejemplo, CYK) • CFGs probabilísticos y CYK ponderado ▪ Representación del significado/Semántica: • Representaciones del conocimiento basadas en lógica • Roles semánticos • Representaciones temporales • Creencias, deseos e intenciones ▪ Métodos basados en corpus ▪ N-gramas y HMMs ▪ Suavizado y retroceso ▪ Ejemplos de uso: Etiquetado POS y morfología ▪ Recuperación de información: • Modelo de espacio vectorial: ◦ TF & IDF • Precisión y exhaustividad ▪ Extracción de información ▪ Traducción de lenguaje ▪ Clasificación de texto, categorización: • Modelo de bolsa de palabras ▪ Aprendizaje profundo para PLN: • RNNs • Transformers • Incrustaciones multimodales (por ejemplo, imágenes + texto) • Modelos de lenguaje generativos	▪ Definir y contrastar gramáticas deterministas y estocásticas, proporcionando ejemplos para mostrar la adecuación de cada una [Definir] ▪ Simular, aplicar o implementar algoritmos clásicos y estocásticos para analizar lenguaje natural [Aplicar] ▪ Identificar los desafíos de representar el significado [Analizar] ▪ Listar las ventajas de usar corpus estándar. Identificar ejemplos de corpus actuales para una variedad de tareas de PLN [Listar/Enumerar] ▪ Identificar técnicas para recuperación de información, traducción de lenguaje y clasificación de texto [Analizar] ▪ 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]
Lecturas : [**JurafskyMartin2024**], [**RussellNorvig2021**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI264. Aprendizaje Profundo (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI264. Aprendizaje Profundo
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI263. Introducción al Aprendizaje de Máquina. (6to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- Implementar redes neuronales profundas usando frameworks modernos (PyTorch/TensorFlow)
- Diseñar arquitecturas para problemas específicos (visión, lenguaje, etc.)
- Analizar limitaciones éticas de modelos profundos

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Entrenamiento y Optimización de Redes Neuronales (10 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Backpropagation moderno: grafos computacionales y diferenciación automática ▪ Optimizadores avanzados: momentum, RMSProp, Adam, y programación de la tasa de aprendizaje	▪ Implementar una red neuronal profunda desde cero, incluyendo su paso de backpropagation [Aplicar] ▪ Seleccionar y ajustar un optimizador y su programación de tasa de aprendizaje para la estabilidad del entrenamiento [Aplicar]
Lecturas : [**goodfellow2016dl**]	

Unidad 2: Técnicas de Regularización para Redes Profundas (5 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Regularización por Dropout ▪ Normalización por lotes (Batch Normalization)	▪ Diagnosticar el sobreajuste en un entrenamiento usando diagnósticos de regularización [Analizar] ▪ Aplicar Dropout y Batch Normalization para mejorar la generalización de un modelo [Aplicar]
Lecturas : [**goodfellow2016dl**]	

Unidad 3: Arquitecturas de Redes Convolucionales y Residuales (9 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitecturas de redes neuronales convolucionales profundas ▪ Conexiones residuales y ResNets	▪ Diseñar una arquitectura CNN profunda para una tarea de visión dada [Aplicar] ▪ Adaptar una arquitectura residual a un dominio de visión específico [Aplicar]
Lecturas : [**he2016resnet**]	

Unidad 4: Arquitecturas Transformer y Modelos Generativos (16 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Transformers y el mecanismo de atención ▪ Modelos generativos: autoencoders variacionales (VAEs) y redes generativas adversariales (GANs)	▪ Entrenar un modelo transformer en un dataset real [Aplicar] ▪ Analizar y contrastar arquitecturas de modelos generativos VAE y GAN [Analizar]
Lecturas : [**vaswani2017attention**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

FG211. Ética Profesional (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	FG211. Ética Profesional
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	2
2.4 Horas	:	1 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos Éticos en IA (10 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comparación: Códigos de ética ACM vs IEEE para IA. ▪ Principios de transparencia y justicia algorítmica.	▪ Explicar diferencias entre marcos éticos [Familiarizarse]. ▪ Evaluar compliance en casos reales [Evaluar].
Lecturas : [**ACM-Ethics2018**], [**IEEE-Ethics2020**]	

Unidad 2: Privacidad y Regulaciones (14 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ GDPR aplicado a ML: derecho a explicación. ▪ Auditoría ética de datasets (ej: ImageNet).	▪ Implementar checklist GDPR para modelos [Usar]. ▪ Identificar violaciones en estudios de caso [Evaluar].
Lecturas : [**GDPR2018**], [**AI-Ethics2021**]	

Unidad 3: Sesgos y Equidad (18 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Métricas de equidad (paridad demográfica, igualdad de oportunidades). ▪ Herramientas: IBM Fairness 360, Google What-If.	▪ Medir sesgos en modelos con Python [Usar]. ▪ Proponer estrategias de mitigación [Evaluar].
Lecturas : [**ACM-Cases2020**], [**Mehrabi2021**]	

Unidad 4: Responsabilidad en Sistemas Autónomos (18 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Atribución legal en fallos de IA (ej: vehículos autónomos). ▪ Normas ISO para IA confiable.	▪ Redactar cláusulas de responsabilidad [Evaluar]. ▪ Analizar conflictos legales [Familiarizarse].
Lecturas : [**Bostrom2014**], [**IEEE-Cases2019**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

FG350. Liderazgo y Desempeño (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	FG350. Liderazgo y Desempeño
2.2 Semestre	:	7mo Semestre.
2.3 Créditos	:	2
2.4 Horas	:	2 HT;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	FG106. Teatro. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos del Liderazgo (15 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y concepto de liderazgo. ▪ Principales teorías del liderazgo: transaccional, transformacional y situacional. ▪ Visión integral del ser humano: dimensiones corporal, afectiva y espiritual. ▪ Motivos de la acción humana: extrínsecos, intrínsecos y trascendentes. ▪ La práctica de la virtud en el ejercicio del liderazgo. ▪ El líder como modelo: ética y coherencia personal. ▪ Estilos de liderazgo y su aplicación en distintos contextos.	▪ Analizar y comprender las bases teóricas del ejercicio del liderazgo [Familiarizarse]. ▪ Asumir la actitud correcta para llevar el liderazgo a la práctica [Familiarizarse]. ▪ Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo [Familiarizarse]. ▪ Reconocer la dimensión ética del liderazgo y su impacto en el entorno personal y social [Familiarizarse].
Lecturas : [**Cardona**], [**Ferreiro**], [**Dianine**], [**DSouza**]	

Unidad 2: Formación de Virtudes y Competencias (15 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Teoría de las competencias y su aplicación al liderazgo. ▪ Reconocimiento e inventario de competencias personales. ▪ Plan de desarrollo personal: diagnóstico, metas y seguimiento. ▪ Modelos mentales y su influencia en el comportamiento. ▪ Necesidades y perfiles emocionales. ▪ Vicios motivacionales: identificación y superación. ▪ Inteligencia emocional aplicada al liderazgo. ▪ La toma de decisiones bajo presión y ambigüedad.	▪ Conocer y desarrollar competencias de liderazgo centradas en la eficacia y el servicio [Familiarizarse]. ▪ Reconocer las tendencias personales y grupales necesarias para el ejercicio del liderazgo [Familiarizarse]. ▪ Elaborar un plan de desarrollo personal de competencias de liderazgo [Usar].
Lecturas : [**Wilkinson**], [**Huete**], [**Cardona**], [**Chinchilla**], [**Goleman**]	

Unidad 3: Gestión y Dirección de Equipos (18 horas)	
Resultados esperados: AG-C02,AG-C04	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ La relación personal con el equipo: confianza y comunicación. ▪ Liderazgo integral: misión, visión y valores compartidos. ▪ Delegación y acompañamiento: el arte de motivar y delegar. ▪ Coaching y mentoring como herramientas de desarrollo. ▪ Resolución de conflictos y dinámica grupal. ▪ Las cinco disfunciones de un equipo y cómo superarlas. ▪ Fundamentos de unidad: cohesión y cultura organizacional. ▪ El liderazgo de servicio y su impacto en la organización.	▪ Desarrollar habilidades para el trabajo en equipo eficaz y cohesionado [Familiarizarse]. ▪ Aplicar técnicas de liderazgo situacional para dirigir equipos en distintos contextos [Usar]. ▪ Diseñar estrategias para la resolución de conflictos y la mejora del desempeño grupal [Usar].
Lecturas : [**Goleman**], [**CardonaP**], [**Hersey**], [**Hunsaker**], [**Hawkins**], [**Lencioni**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.

- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS281. Computación en la Sociedad (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS281. Computación en la Sociedad
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	2
2.4 Horas	:	2 HT;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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. (Usage)

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

7. TEMAS

Unidad 1: Fundamentos e Historia Temprana de la Informática (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ La historia de la informática: hardware, software y humano/organizacional. ▪ El papel de la historia en el presente, incluyendo dentro de diferentes contextos sociales, y la relevancia de esta historia en el futuro. ▪ Edad I (Pre-digital): Computación analógica antigua (Stonehenge, mecanismo de Anticitera, reloj de la Catedral de Salisbury, etc.), tablas numéricas calculadas por humanos, Euclides, Lovelace, Babbage, Gödel, Church, Turing, hardware pre-electrónico (electromecánico y mecánico). ▪ Edad II (Computación moderna temprana): ENIAC, UNIVAC, Bombas (Bletchley Park y descifradores de códigos), compañías informáticas (por ejemplo, IBM), mainframes, etc.	▪ 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)] ▪ 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] ▪ Identificar tendencias formativas y consecuentes en la historia del campo de la informática [Analizar] ▪ Identificar las contribuciones de individuos u organizaciones pioneras (laboratorios de investigación, compañías informáticas, oficinas gubernamentales) en el campo de la informática [Analizar]
Lecturas : [**Quinn2019**], [**Baase2017**]	

Unidad 2: Impacto Social e IA (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Implicaciones sociales (por ejemplo, ideologías políticas y culturales) en un mundo hiperconectado donde las capacidades y el impacto de las redes sociales, la inteligencia artificial y la informática en general están evolucionando rápidamente. ▪ Impacto de las aplicaciones informáticas (por ejemplo, redes sociales, aplicaciones de inteligencia artificial) en el bienestar individual y la seguridad de todo tipo (por ejemplo, física, emocional, económica). ▪ Consecuencias de involucrar tecnologías informáticas, particularmente inteligencia artificial, tecnologías biométricas y sistemas de toma de decisiones algorítmicos, en la vida cívica (por ejemplo, tecnología de reconocimiento facial, etiquetas biométricas, algoritmos de distribución de recursos, software policial) y cómo la agencia y supervisión humana son cruciales. ▪ Crecimiento y control de internet, datos, informática e inteligencia artificial.	▪ Describir las diferentes formas en que la tecnología informática (redes, computación móvil, inteligencia artificial) media la interacción social a nivel personal y colectivo [Describir] ▪ Interpretar el contexto social de un diseño dado y su implementación [Interpretar] ▪ Analizar la eficacia de un diseño e implementación dados utilizando datos empíricos [Analizar] ▪ Comprender las implicaciones del uso de la tecnología (por ejemplo, redes sociales) para diferentes identidades, culturas y comunidades [Explicar] ▪ Describir el papel de internet en facilitar la comunicación entre ciudadanos, gobiernos y entre sí [Describir] ▪ Analizar los efectos de la dependencia de la informática en la implementación de la democracia (por ejemplo, entrega de servicios sociales, votación electrónica) [Analizar] ▪ 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]
Lecturas : [**Quinn2019**]	

Unidad 3: Fundamentos y Marcos del Análisis Ético (4 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Evitar falacias y tergiversaciones en la argumentación. ▪ Teorías éticas y toma de decisiones (marcos filosóficos y sociales, por ejemplo [1]). ▪ Reconocimiento del papel que juega la cultura en nuestra comprensión, adopción, diseño y uso de la tecnología informática. ▪ Por qué la ética es importante en la informática y cómo la ética es similar y diferente de las leyes y normas sociales. ▪ Introducción a los marcos éticos (por ejemplo, consecuencialismo como el utilitarismo, no consecuencialismo como el deber, derechos o justicia, centrado en el agente como la virtud o el feminismo, contractualismo, ética del cuidado) y su uso para analizar un dilema ético.	▪ Describir cómo un contexto cultural dado impacta la toma de decisiones [Describir] ▪ Expresar el uso de ejemplos y analogías en la argumentación ética [Analizar] ▪ Analizar (y evitar) falacias lógicas básicas en un argumento [Analizar] ▪ Analizar un argumento para identificar premisas y conclusión [Analizar] ▪ 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] ▪ Justificar una decisión tomada sobre bases éticas [Justificar] ▪ Discutir las ventajas y desventajas de los marcos éticos tradicionales [Debatir] ▪ Analizar dilemas éticos relacionados con la creación y uso de tecnología desde múltiples perspectivas utilizando marcos éticos [Analizar]
Lecturas : [**Quinn2019**], [**Baase2017**]	

Unidad 4: Valores, Naturaleza y Actualización Profesional (4 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Valores de la comunidad y las leyes por las que vivimos. ▪ La naturaleza de ser un profesional, incluyendo cuidado, atención, disciplina, responsabilidad fiduciaria y tutoría (mentoring). ▪ Mantenerse actualizado como profesional de la informática en términos de familiaridad, herramientas, habilidades, marcos legales y profesionales, así como la capacidad y responsabilidad de autoevaluarse y progresar en el campo de la informática.	▪ Identificar problemas éticos que surgen en el diseño de software, prácticas de desarrollo y despliegue de software [Analizar] ▪ Discutir cómo abordar problemas éticos en situaciones específicas [Debatir] ▪ Describir los mecanismos que típicamente existen para que un profesional se mantenga actualizado en asuntos éticos [Describir]
Lecturas : [**Baase2017**]	

Unidad 5: Fundamentos de la Propiedad Intelectual (4 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Derechos de propiedad intelectual. ▪ Fundamentos legales para la protección de la propiedad intelectual. ▪ Fundamentos filosóficos de la propiedad intelectual. ▪ Formas de propiedad intelectual (por ejemplo, derechos de autor, patentes, secretos comerciales, marcas registradas) y los derechos que protegen.	▪ Defender los usos legales y éticos de materiales con derechos de autor [Defender] ▪ Discutir las bases filosóficas de la propiedad intelectual en un contexto apropiado (por ejemplo, país) [Debatir] ▪ Contrastar las protecciones y obligaciones de los derechos de autor, patentes, secretos comerciales y marcas registradas [Contrastar] ▪ Describir la justificación para la protección legal de la propiedad intelectual en el contexto apropiado (por ejemplo, país) [Describir]
Lecturas : [**Quinn2019**]	

Unidad 6: Derechos de PI Digital y Derechos de Autor (6 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propiedad intelectual digital intangible (IDIP). ▪ Limitaciones a las protecciones de derechos de autor, incluyendo uso justo (fair use) y la doctrina de la primera venta (first sale doctrine). ▪ Leyes y tratados de propiedad intelectual que impactan la aplicación de los derechos de autor.	▪ Describir y criticar la legislación y los precedentes destinados a las infracciones de derechos de autor digitales [Describir] ▪ Identificar ejemplos contemporáneos de propiedad intelectual digital intangible [Analizar] ▪ Discutir si un uso de material con derechos de autor es probable que sea uso justo (fair use) [Debatir] ▪ 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] ▪ Discutir la naturaleza de las leyes anti-elusión (anti-circumvention) en el contexto de la protección de derechos de autor [Debatir]
Lecturas : [**Tavani2013**]	

Unidad 7: Datos, Vigilancia y Tecnología de Privacidad (4 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Implicaciones de privacidad de la recopilación generalizada de datos, incluyendo pero no limitado a bases de datos transaccionales, almacenes de datos, sistemas de vigilancia, computación en la nube e inteligencia artificial. ▪ Concepciones de anonimato, seudonimidad e identidad. ▪ Soluciones basadas en tecnología para la protección de la privacidad (por ejemplo, encriptación de extremo a extremo y privacidad diferencial).	▪ Evaluar soluciones a las amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Evaluar] ▪ 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] ▪ Distinguir los conceptos y objetivos del anonimato y la seudonimidad [Distinguir] ▪ Describir las ramificaciones de las protecciones de privacidad basadas en tecnología, incluyendo la privacidad diferencial y la encriptación de extremo a extremo [Describir]
Lecturas : [**Baase2017**]	

Unidad 8: Huella Ambiental de los Sistemas Informáticos (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Impactos ambientales, sociales y culturales de las decisiones de implementación (por ejemplo, objetivos de sostenibilidad, sesgo/resultados algorítmicos, viabilidad económica y consumo de recursos). ▪ Impactos sociales y ambientales locales/regionales/globales de los sistemas informáticos y su uso (por ejemplo, huella de carbono, uso de recursos, residuos electrónicos) debido al hardware (por ejemplo, residuos electrónicos, centros de datos, utilización de elementos y recursos raros, reciclaje) y software (por ejemplo, servicios basados en la nube, blockchain, entrenamiento y uso de modelos de IA). Esto incluye el uso diario de hardware (hardware barato reemplazado con frecuencia) y software (navegación web, correo electrónico y otros servicios con demandas computacionales ocultas/remotas).	▪ 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)] ▪ Describir los impactos sociales y ambientales globales del uso y disposición de computadoras [Describir] ▪ 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] ▪ 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)]
Lecturas : [**Quinn2019**]	

Unidad 9: Modelos y Estrategias Económicas (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos económicos: regulados y no regulados, monopolios, efectos de red (network effects) y mercado abierto; economías del conocimiento y de la atención (knowledge and attention economies). ▪ Estrategias de precios y despliegue: obsolescencia programada (planned obsolescence), suscripciones, freemium, licencias de software, código abierto (open-source), software libre (free software), adware. ▪ Impactos de las diferencias en el acceso a recursos informáticos y el efecto de la oferta y demanda de mano de obra calificada en la calidad de los productos informáticos.	▪ Resumir los efectos sociales de los modelos económicos (por ejemplo, las economías del conocimiento y de la atención) [Resumir] ▪ Describir las diferencias y similitudes de estrategias competitivas (por ejemplo, suscripción vs freemium vs gratis) [Describir] ▪ Discutir ejemplos de brechas digitales [Debatir]
Lecturas : [**Baase2017**]	

Unidad 10: Delitos y Ataques Informáticos (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Delitos informáticos, recurso legal para criminales informáticos e impacto en víctimas y sociedad. ▪ Ingeniería social, fraude habilitado por la informática, robo de identidad y recuperación de estos. ▪ Terrorismo cibernético, hacking criminal y hacktivismo. ▪ Malware, virus, gusanos. ▪ Ataques a infraestructura crítica como redes eléctricas y oleoductos.	▪ Enumerar ejemplos clásicos de delitos informáticos e incidentes de ingeniería social con impacto social [Listar/Enumerar] ▪ Identificar problemas con las leyes que se aplican a los delitos informáticos [Analizar] ▪ Describir la motivación y ramificaciones del terrorismo cibernético, robo de datos, hacktivismo, ransomware y otros ataques [Describir] ▪ Examinar los problemas éticos y legales que rodean el mal uso del acceso y varias violaciones de seguridad [Examinar]
Lecturas : [**Baase2017**]	

Unidad 11: Identidad y Diversidad en la Informática (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Cómo la identidad impacta y es impactada por las tecnologías y entornos informáticos (académicos y profesionales). ▪ Los beneficios de los equipos de desarrollo diversos y los impactos de los equipos que no son diversos.	▪ Definir y distinguir equidad, igualdad, diversidad, inclusión y accesibilidad [Definir] ▪ 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] ▪ 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)]
Lecturas : [**Quinn2019**]	

Unidad 12: Comunicación Técnica y Profesional (2 horas)	
Resultados esperados: AG-C01,AG-C02	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comunicación oral, escrita y electrónica en equipo y grupo. ▪ Materiales de comunicación técnica (por ejemplo, código fuente y documentación, tutoriales, materiales de referencia, documentación de API). ▪ Comunicarse con diferentes partes interesadas, como clientes, liderazgo o el público. ▪ Compensaciones (tradeoffs) en factores competitivos que afectan los canales y elecciones de comunicación.	▪ 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] ▪ Analizar documentación técnica escrita para verificar precisión técnica, concisión, falta de ambigüedad y conciencia de la audiencia [Analizar] ▪ Componer y entregar una presentación formal consciente de la audiencia, accesible y organizada [Componer] ▪ Identificar y describir cualidades de la comunicación efectiva (por ejemplo, virtual, cara a cara, intragrupo, documentos compartidos) [Analizar] ▪ Elegir una forma apropiada de comunicar preocupaciones éticas delicadas [Evaluar (valorar)]
Lecturas : [**Quinn2019**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS3I1. Seguridad en Computación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS3I1. Seguridad en Computación
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS231. Redes y Comunicación. (6to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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 gobernar sistemas seguros, comprendiendo las responsabilidades éticas y profesionales involucradas en la protección de la información y la infraestructura.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Principios y Mentalidad de Seguridad (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollar una mentalidad de seguridad que incorpore conceptos transversales: confidencialidad, integridad, disponibilidad, evaluación de riesgos, pensamiento sistémico, pensamiento adversarial, pensamiento centrado en el humano. ▪ Vulnerabilidades, amenazas, superficies de ataque y vectores de ataque Protección y Seguridad. ▪ Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS) Protección y Seguridad. ▪ Principios y prácticas de protección, por ejemplo, privilegio mínimo, diseño abierto, valores predeterminados seguros ante fallos, defensa en profundidad y confianza cero; y cómo pueden implementarse Principios del Sistema Operativo, Protección y Seguridad, Prácticas de Codificación, Construcción a Escala y Proceso, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Consideraciones de optimización entre seguridad, privacidad, rendimiento y otros objetivos de diseño Diseño de Sistemas, Prácticas de Desarrollo de Software, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Conceptos de confianza (trust) y confiabilidad (trustworthiness).	▪ Evaluar un sistema para posibles ataques que pueda lanzar un adversario [Evaluar] ▪ Diseñar y desarrollar enfoques para proteger un sistema de un conjunto de amenazas identificadas [Diseñar] ▪ Describir cómo se puede evitar el daño a la privacidad del usuario [Describir] ▪ Desarrollar un sistema que incorpore varios principios de seguridad y privacidad [Crear]
Lecturas : [**Pfleeger2015**], [**Stallings2017**]	

Unidad 2: Fundamentos de Criptografía (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diferencias entre las vistas algorítmica, aplicada y matemática de la criptografía. ▪ Preliminares matemáticos: aritmética modular, algoritmo euclidiano, independencia probabilística, conceptos básicos de álgebra lineal, teoría de números, campos finitos, complejidad, análisis asintótico Matemáticas Discretas, álgebra Lineal. ▪ Criptografía básica: criptografía de clave simétrica y de clave pública Estructuras de Datos Fundamentales, Algoritmos Fundamentales, Algoritmos Avanzados, Matemáticas Discretas. ▪ Criptosistemas clásicos, como cifrados de desplazamiento, sustitución, transposición, libros de códigos y máquinas Matemáticas Discretas. ▪ Principio de Kerckhoff y uso de bibliotecas evaluadas (vetted) Prácticas de Codificación, Construcción a Escala y Proceso.	▪ Explicar el papel de la criptografía en el soporte de la seguridad y la privacidad [Explicar] ▪ Discutir los riesgos de inventar los propios métodos criptográficos [Debatir] ▪ Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Debatir] ▪ Implementar y criptoanalizar cifrados clásicos [Implementar]
Lecturas : [**Stallings2017**]	

Unidad 3: Inyección y Validación de Entrada (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Vulnerabilidades y debilidades comunes. ▪ Inyección SQL y otros ataques de inyección. ▪ Técnicas de cross-site scripting y mitigaciones. ▪ Validación de entrada y sanitización de datos Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.	▪ Identificar problemas subyacentes en ejemplos dados de una enumeración de debilidades comunes y explicar cómo pueden evitarse [Analizar] ▪ Aplicar técnicas de validación de entrada y sanitización de datos para mejorar la seguridad de un programa [Aplicar] ▪ Evaluar y prevenir ataques de inyección SQL en una aplicación de base de datos [Evaluar] ▪ Evaluar y prevenir ataques de cross-site scripting contra un sitio web [Evaluar] ▪ Construir un programa que pruebe todos los errores de manejo de entrada [Crear]
Lecturas : [**Pfleeger2015**]	

Unidad 4: Análisis de Malware y Seguridad Avanzada (6 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Técnicas de seguridad ofensiva como defensa. ▪ Técnicas de detección de malware asistidas por IA. ▪ Ransomware: creación, prevención y mitigación. ▪ Uso seguro de componentes de terceros Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Malware: variedades, creación, ingeniería inversa y defensa contra ellos Traducción y Ejecución de Lenguajes, Modelo de Ejecución y Memoria de Sistemas. ▪ Garantía (assurance): pruebas (incluyendo fuzzing y pruebas de penetración), verificación y validación Protección y Seguridad, Conceptos y Construcciones Básicas de Programación, Prácticas de Programación: Pruebas, Depuración y Documentación, Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Análisis estático y dinámico Análisis de Programas y Analizadores, Protección y Seguridad, Evaluación, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Compiladores seguros y generación de código seguro Traducción y Ejecución de Lenguajes, Comportamiento en Tiempo de Ejecución y Sistemas.	▪ Describir diferentes tipos de software malicioso [Describir] ▪ Explicar los riesgos del mal uso de interfaces con código de terceros y cómo usar correctamente el código de terceros [Explicar] ▪ Discutir la necesidad de actualizar el software para corregir vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Debatir] ▪ Aplicar herramientas estáticas y dinámicas para identificar fallos de programación [Aplicar] ▪ Evaluar un sistema para detectar la existencia de malware y eliminarlo [Evaluar] ▪ Implementar técnicas preventivas para reducir la ocurrencia de ransomware [Implementar]
Lecturas : [**SikorskiHonig2012**], [**Eilam2005**], [**NIST800-53**]	

Unidad 5: Análisis de Amenazas e Ingeniería de Seguridad (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Objetivos de la ingeniería de seguridad: construir sistemas que permanezcan confiables a pesar de errores, accidentes o adversarios maliciosos Prácticas de Codificación, Construcción a Escala y Proceso, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Análisis de problemas y análisis situacional para abordar la seguridad del sistema Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Análisis de seguridad, que cubre análisis de requisitos de seguridad; análisis de controles de seguridad; análisis de amenazas; y análisis de vulnerabilidades Análisis de Programas y Analizadores, Evaluación. ▪ Dominios de ataque de seguridad y superficies de ataque, por ejemplo, comunicaciones y redes, hardware, físico, ingeniería social, software y cadena de suministro Seguridad de Red. ▪ Modos, técnicas y tácticas de ataque de seguridad, por ejemplo, abuso de autenticación; fuerza bruta; manipulación de búfer; inyección de código; inserción de contenido; denegación de servicio; escucha; omisión de función; suplantación; ataque de integridad; intercepción; phishing; análisis de protocolo; abuso de privilegio; suplantación (spoofing); e inyección de tráfico Seguridad de Red, Protección y Seguridad, Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas.	▪ Crear un modelo de amenazas para un sistema o diseño de sistema [Crear] ▪ Aplicar análisis situacional para desarrollar soluciones seguras bajo un escenario especificado [Aplicar] ▪ Identificar y mitigar vulnerabilidades y debilidades de seguridad en un sistema [Analizar]
Lecturas : [**Anderson2020**]	

Unidad 6: Técnicas de Informática Forense (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios y metodologías básicas para la informática forense. ▪ Diseño de sistemas para informática forense. ▪ Informática forense en diferentes situaciones: sistemas operativos, sistemas de archivos, informática forense de aplicaciones, informática forense web, informática forense de red, informática forense de dispositivos móviles, uso de auditoría de bases de datos Seguridad de Red. ▪ Ataques a la informática forense y prevención de tales ataques.	▪ Explicar qué es una investigación digital y cómo se puede implementar Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Explicar] ▪ Diseñar e implementar software para apoyar la informática forense [Diseñar] ▪ Extraer datos de un disco duro para cumplir con la ley Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad [Extraer] ▪ Recuperar datos basados en un término de búsqueda dado desde un sistema del que se ha hecho una imagen (imaged) [Analizar] ▪ 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] ▪ Capturar y analizar tráfico de red. Seguridad de Red [Analizar] ▪ Desarrollar enfoques para abordar los desafíos asociados con la informática forense de dispositivos móviles [Crear] ▪ Aplicar herramientas de informática forense para investigar violaciones de seguridad [Aplicar] ▪ Identificar y mitigar métodos anti-forenses [Analizar]
Lecturas : [**Pfleeger2015**]	

Unidad 7: Gobierno, Gestión y Política de Seguridad (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Proteger activos críticos de amenazas. ▪ Gobierno de seguridad (security governance): objetivos organizacionales y evaluación general de riesgos. ▪ Gestión de seguridad (security management): lograr y mantener niveles apropiados de confidencialidad, integridad, disponibilidad, responsabilidad (accountability), autenticidad y confiabilidad Pruebas de Rendimiento y Evaluación Comparativa, Herramientas y Análisis de Pruebas, Conceptos de verificación y validación, Planificación y Tipos de Pruebas, Prácticas Avanzadas de Pruebas. ▪ Política de seguridad (security policy): políticas organizacionales, políticas específicas de temas, políticas específicas de sistemas.	▪ Describir activos críticos y cómo pueden protegerse [Describir] ▪ Diferenciar entre gobierno de seguridad, gestión y controles, dando ejemplos de cada uno [Diferenciar] ▪ Describir un control técnico e implementarlo para mitigar amenazas específicas [Describir]
Lecturas : [**Anderson2020**]	

Unidad 8: Ética, Ley y Sociedad en Seguridad (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Impactos sociales de las fallas en seguridad y privacidad Equidad Digital y Accesibilidad, Impacto Social e IA, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Aplicabilidad de leyes y regulaciones sobre seguridad y privacidad Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Consideraciones éticas profesionales al diseñar sistemas seguros y mantener la privacidad; hacking ético Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Implicaciones de seguridad y privacidad de actores maliciosos de IA/aprendizaje automático, por ejemplo, identificación de deep fakes Formulación del Aprendizaje Automático y Evaluación, Aprendizaje Profundo, RL y Ética, Introducción a la IA: Conceptos y Comportamiento Inteligente, Características del Problema y Naturaleza de los Agentes, Fundamentos de Aprendizaje Automático, Aprendizaje Automático: Datos, Representaciones y Evaluación, Redes Neuronales y Ética, Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad. ▪ Impactos sociales de los dispositivos del Internet de las Cosas (IoT) y otras tecnologías emergentes en la seguridad y la privacidad Libertades Civiles y Expresión, Datos, Vigilancia y Tecnología de Privacidad, Fundamentos Filosóficos y Legislación de Privacidad, Delitos y Ataques Informáticos, Leyes, Políticas y Responsabilidad en Seguridad.	▪ Calcular el impacto de una falla de seguridad en un sistema dado [Calcular] ▪ Construir un sistema que se ajuste a las leyes de seguridad [Crear] ▪ Aplicar un conjunto de regulaciones de privacidad para diseñar un sistema que proteja la privacidad [Aplicar] ▪ Evaluar las ramificaciones legales de un sistema que no se ajusta a las leyes y regulaciones aplicables [Evaluar]
Lecturas : [**Anderson2020**], [**Pfleeger2015**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS3P1. Computación Paralela y Distribuída (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS3P1. Computación Paralela y Distribuída
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	▪ CS212. Análisis y Diseño de Algoritmos. (5to Sem) ▪ CS231. Redes y Comunicación. (6to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Programas: Paralelismo Declarativo (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paralelismo • 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 • Definir orden: por ejemplo, usando relaciones de "sucede antes" o grafos acíclicos dirigidos serie/paralelo que representan programas. • Independencia: determinar cuándo el orden no importa, en términos de conmutatividad, dependencias, precondiciones. • 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.	▪ 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] ▪ Explicar por qué los conceptos de consistencia y tolerancia a fallos no surgen en programas puramente secuenciales [Explicar]
Lecturas : [**Pacheco2021**]	

Unidad 2: Programas: Inicio de Actividades (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Iniciar actividades • 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 • 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. • 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. • Dependiente: Habilitar al completarse otras; por ejemplo, secuenciando conjuntos de acciones paralelas. Ver también: Coordinación. • Granularidad: El costo de ejecución de los cuerpos de las acciones debe superar la sobrecarga de organizarlos.	▪ Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]
Lecturas : [**Pacheco2021**]	

Unidad 3: Programas: Propiedades de Ejecución (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propiedades de Ejecución • Ejecución no determinista de acciones sin orden. • Consistencia: Garantizar el acuerdo entre partes sobre valores y predicados cuando sea necesario para evitar carreras, mantener seguridad y atomicidad, o llegar a consenso. • 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. • Los compromisos son un foco de evaluación. Ver también: Evaluación.	▪ Escribir una función que cuente eventos, como recepciones de paquetes de red, de manera eficiente [Escribir]
Lecturas : [**VanSteen2023**]	

Unidad 4: Programas: Distribución (2 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Distribución • 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 • Un dispositivo puede dividir el tiempo o emular múltiples acciones paralelas mediante menos procesadores mediante planificación y virtualización. Ver también: Planificación • Nombrar o identificar lugares (por ejemplo, IDs de dispositivo) y acciones como partes (por ejemplo, IDs de hilo). • Las actividades a través de lugares pueden comunicarse a través de medios. Ver también: Comunicación	▪ Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir]
Lecturas : [**VanSteen2023**]	

Unidad 5: Programas: Mapeos de Implementación (2 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Una o más de las siguientes asignaciones y mecanismos a través de sistemas en capas: • Paralelismo a nivel de instrucción y datos en la CPU. Ver también: Organización Funcional. • SIMD y paralelismo de datos heterogéneo. Ver también: Arquitecturas Heterogéneas. • Concurrencia planificada en multinúcleo, tareas, actores. Ver también: Planificación. • Clústeres, nubes; aprovisionamiento elástico. Ver también: Aspectos Comunes: Plataformas, APIs y Restricciones, Aspectos Comunes: Lenguajes de Plataforma y Patrones Web. • Sistemas distribuidos en red. Ver también: Aplicaciones en Red. • Tecnologías emergentes como la computación cuántica y la computación molecular.	▪ Explicar las compensaciones entre diferentes estrategias de mapeo en términos de rendimiento, costo y complejidad de implementación [Explicar]
Lecturas : [**Pacheco2021**], [**Hwu2022**]	

Unidad 6: Programación GPU (4 horas)

Resultados esperados: AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Computación de Propósito General en GPU (GPGPU) • Arquitectura de GPU: multiprocesadores de flujo (SMs), núcleos CUDA y modelo de ejecución por warp. • Modelo de ejecución SIMT (Single Instruction, Multiple Threads) y sus diferencias respecto al SIMD de CPU. • Interacción host-dispositivo: transferencia de datos entre memoria de CPU (host) y GPU (dispositivo) vía PCIe. • Casos de uso: computación científica, aprendizaje automático, procesamiento de imágenes y simulación a gran escala. ▪ Modelo de Programación CUDA • Jerarquía de ejecución CUDA: grids, bloques de hilos e hilos individuales; mapeo al hardware GPU. • Definición y lanzamiento de kernels; parámetros de configuración (dimensiones de grid y bloque). • Indexación de hilos: threadIdx, blockIdx, blockDim, gridDim; cálculo de índices globales en 1D, 2D y 3D. • Divergencia de warp: implicaciones de las bifurcaciones dentro de un warp en el rendimiento. • Primitiva de sincronización intra-bloque: __syncthreads(). ▪ Jerarquía de Memoria CUDA • Memoria global: grande, alta latencia, accesible por todos los hilos; patrones de acceso coalescente para el rendimiento. • Memoria compartida: memoria en chip de baja latencia compartida dentro de un bloque de hilos; conflictos de banco. • Registros y memoria local: almacenamiento privado por hilo. • Memoria constante y de textura: cachés de solo lectura optimizadas para patrones de acceso específicos. • Memoria unificada: modelo de programación simplificado con migración automática de datos entre host y dispositivo.	▪ 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] ▪ 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] ▪ 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]

Lecturas : [**Kirk2016**], [**CUDADocs**]

Unidad 7: Comunicación (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Medios • Variedades: canales (paso de mensajes o E/S), memoria compartida, heterogéneos, almacenes de datos. • Dependencia de la disponibilidad y naturaleza del hardware subyacente, conectividad y protocolos; soporte del lenguaje, emulación. Ver también: Interconexión y Comunicación. ▪ Canales • Medios de comunicación de parte a parte explícitos (generalmente nombrados). • 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. • APIs de canales de E/S. ▪ Memoria • 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). • Jerarquías de memoria: Múltiples capas de dominios, alcances y cachés compartidos; localidad: latencia, falso uso compartido. • Propiedades de consistencia: Límites de atomicidad a nivel de bits, coherencia, orden local. ▪ Almacenes de Datos • Estructuras de datos mantenidas cooperativamente que implementan mapas y TADs relacionados. • Variedades: Propiedad exclusiva, compartida, fragmentada, replicada, inmutable, versionada.	▪ 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] ▪ 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] ▪ Escribir un programa paralelo que solicite datos de múltiples sitios y los resuma usando alguna forma de reducción [Escribir] ▪ Comparar el rendimiento de versiones con y sin búfer de un programa productor-consumidor [Comparar]
Lecturas : [**VanSteen2023**], [**Kleppmann2017**]	

Unidad 8: Comunicación: Propiedades y Extensiones (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Una o más de las siguientes propiedades y extensiones: • Topologías: Unicast, Multicast, Buzones, Conmutadores; Enrutamiento a través de redes de interconexión de hardware y software. • Propiedades de concurrencia de medios: Orden, consistencia, idempotencia, superposición de comunicación con computación. • 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). • 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. • Formatos de datos: Serialización, validación, cifrado, compresión. • Políticas de canal: Puntos finales, sesiones, almacenamiento en búfer, respuesta a saturación (esperar vs. descartar), control de velocidad. • 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. • Formalización y análisis de comunicación por canales; por ejemplo, CSP. • Aplicaciones de la teoría de colas para modelar y predecir el rendimiento.	▪ Determinar si un esquema de comunicación dado proporciona propiedades de seguridad suficientes para un uso dado [Determinar] ▪ Dar un ejemplo de un escenario en el que los envíos de mensajes bloqueantes pueden causar un punto muerto (deadlock) [Crear] ▪ Describir al menos una técnica de diseño para evitar fallos de vivacidad en programas que usan múltiples bloqueos [Describir]
Lecturas : [**VanSteen2023**], [**Kleppmann2017**]	

Unidad 9: Memoria y Consistencia (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de memoria, consistencia de datos y tolerancia a fallos: • Modelos de memoria: consistencia secuencial y de liberación/adquisición. • Gestión de memoria; incluyendo la reclamación de datos compartidos; conteo de referencias y alternativas. • 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. • Emular memoria compartida: memoria compartida distribuida, Acceso Directo a Memoria Remota (RDMA). • Consistencia de almacenes de datos: Atomicidad, linealizabilidad, transaccionalidad, coherencia, orden causal, resolución de conflictos, consistencia eventual, cadenas de bloques. • Fallos, particionamiento y fallos parciales; votación; protocolos como Paxos y Raft. • Compromisos de diseño entre consistencia, disponibilidad, tolerancia a partición (fallos); imposibilidad de cumplir con todos a la vez. • Seguridad y confianza: Fallos bizantinos, prueba de trabajo y alternativas.	▪ 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] ▪ Escribir un programa que ilustre el reordenamiento de acceso a memoria o de mensajes [Escribir] ▪ Describir los méritos relativos del control de concurrencia optimista versus conservador bajo diferentes tasas de contención entre actualizaciones [Describir] ▪ Dar un ejemplo de un escenario en el que un intento de actualización optimista podría nunca completarse [Crear] ▪ Modificar un sistema concurrente para usar un almacén de datos más escalable, confiable o disponible [Crear] ▪ 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]
Lecturas : [**VanSteen2023**], [**Kleppmann2017**]	

Unidad 10: Coordinación (7 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Dependencias • La iniciación o progreso de una actividad puede depender de otras actividades, para evitar condiciones de carrera, garantizar la terminación o cumplir otros requisitos. • Garantizar el progreso evitando ciclos de dependencia, usando condiciones monótonas, eliminando dependencias no esenciales. ▪ Constructos de control y patrones de diseño: • Basados en finalización: Barreras, reuniones (joins), incluido el control de terminación. • Habilitados por datos: Colas, diseños productor-consumidor. • Basados en condición: Polling, reintentos, retrocesos, ayuda, suspensión, señalización, tiempos de espera. • Reactivos: Habilitar y activar continuaciones.	▪ Mostrar cómo garantizar que un programa termine correctamente cuando todas las tareas concurrentes de un conjunto hayan finalizado [Diseñar] ▪ Escribir una función que cuente eventos, como entradas de sensores o recepciones de paquetes de red, de manera eficiente [Escribir] ▪ Escribir un programa de filtro/mapeo/reducción en múltiples estilos [Escribir] ▪ Escribir un programa en el que la terminación de un conjunto de acciones paralelas sea seguida por otra [Escribir] ▪ Escribir un servicio que cree un hilo (u otra forma de activación procedimental) para devolver una página web solicitada a cada nuevo cliente [Escribir]
Lecturas : [**Herlihy2020**], [**VanSteen2023**]	

Unidad 11: Coordinación: Sincronización y Atomicidad (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Atomicidad • Instrucciones atómicas, ordenamientos de acceso local impuestos. • Bloqueos y exclusión mutua; granularidad de bloqueo. • Uso de bloqueos en un lenguaje específico; mantener la vivacidad sin introducir carreras. • Evitar punto muerto (*deadlock*): Ordenamiento, mayor granularidad, reintentos aleatorios; retrocesos, encapsulación mediante administradores de bloqueos. • Errores comunes: No bloquear o desbloquear cuando es necesario, mantener bloqueos mientras se invocan operaciones desconocidas. • Evitar bloqueos: replicación, solo lectura, propiedad, y construcciones no bloqueantes.	▪ Mostrar cómo evitar o reparar un error de carrera en un programa dado [Analizar]
Lecturas : [**Herlihy2020**]	

Unidad 12: Coordinación: Propiedades Avanzadas (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Una o más de las siguientes propiedades y extensiones: • Propiedades de progreso incluyendo libre de bloqueo (*lock-free*), libre de espera (*wait-free*), equidad (*fairness*), planificación por prioridad, interacciones con consistencia, confiabilidad. • Rendimiento con respecto a contención, granularidad, convoy, escalabilidad. • Estructuras de datos y algoritmos no bloqueantes. • Propiedad y control de recursos. • 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. • Control basado en transacciones: Optimista y conservador. • Bloqueo distribuido: confiabilidad. • 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. • Especulación, selección, cancelación; consecuencias en observabilidad y seguridad. • Control de recursos usando semáforos y variables de condición. • Flujo de control: Planificación de computaciones, bucles serie-paralelo con líderes (posiblemente elegidos), tuberías y flujos, paralelismo anidado. • Excepciones y fallos. Manejadores, detección, tiempos de espera, tolerancia a fallos, votación.	▪ Escribir un programa que busque especulativamente una solución mediante múltiples actividades, terminando las demás cuando se encuentre una [Escribir] ▪ Escribir un programa en el que una excepción numérica (como división por cero) en una actividad cause la terminación de las demás [Escribir] ▪ 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]
Lecturas : [**Herlihy2020**]	

Unidad 13: Evaluación (7 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Requisitos de seguridad (safety) y vivacidad (*liveness*) en términos de constructos de lógica temporal para expresar "siempre" y "eventualmente" Ver también: Computación Paralela y Distribuida. ▪ Identificar, probar y reparar violaciones, incluyendo formas comunes de errores como no garantizar el orden necesario (errores de carrera), atomicidad (incluyendo errores de "verificar luego actuar") y terminación (bloqueo activo). ▪ Métricas de requisitos de rendimiento para rendimiento (*throughput*), capacidad de respuesta, latencia, disponibilidad, consumo de energía, escalabilidad, uso de recursos, costos de comunicación, espera y control de velocidad, equidad; acuerdos de nivel de servicio. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento. ▪ Impacto en el rendimiento de las opciones de diseño e implementación, incluyendo granularidad, sobrecarga, costos de consenso y consumo de energía. Ver también: Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social. ▪ Estimar limitaciones de escalabilidad, por ejemplo usando la Ley de Amdahl o la Ley de Escalabilidad Universal. Ver también: Eventos, Herramientas y Experimentación, Métricas y Benchmarks de Rendimiento, Modelos Analíticos de Rendimiento.	▪ Revisar una especificación para habilitar el paralelismo y la distribución sin violar otras propiedades o características esenciales [Rediseñar] ▪ Explicar cómo las nociones concurrentes de seguridad (safety) y vivacidad (liveness) extienden sus contrapartes secuenciales [Explicar] ▪ Especificar un conjunto de invariantes que deben mantenerse en cada paso de cómputo de paralelismo masivo [Analizar] ▪ 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] ▪ 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] ▪ 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]
Lecturas : [**Sterling2024**], [**Pacheco2021**]	

Unidad 14: Evaluación: Métodos Formales (3 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Métodos formales de verificación y análisis: • Extensiones a requisitos formales secuenciales como la linealizabilidad. • Especificaciones de protocolo, sesión y transacciones. • Uso de herramientas como Lenguaje Unificado de Modelado (UML), Lógica Temporal de Acciones (TLA), lógicas de programa. • 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. • 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. • Análisis del modelo de Grafo Acíclico Dirigido (DAG) de eficiencia algorítmica (trabajo, span, caminos críticos).	▪ Especificar y medir el comportamiento cuando un servicio es solicitado por un número inesperadamente grande de clientes [Analizar] ▪ Identificar y reparar un problema de rendimiento debido a cuellos de botella secuenciales [Analizar] ▪ Comparar empíricamente el rendimiento (*throughput*) de dos implementaciones de un diseño común (quizás usando un marco de pruebas existente) [Comparar]
Lecturas : [**Sterling2024**], [**Pacheco2021**]	

Unidad 15: Evaluación: Pruebas y Medición (2 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas de prueba y técnicas de medición: • 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. • 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. • Análisis específicos del dominio de aplicación y técnicas de evaluación.	▪ Identificar y reparar un problema de rendimiento debido a latencia de comunicación o datos [Analizar] ▪ Identificar y reparar un problema de rendimiento debido a sobrecarga en la gestión de recursos [Analizar] ▪ Identificar y reparar un problema de confiabilidad o disponibilidad [Analizar]
Lecturas : [**Sterling2024**], [**Pacheco2021**]	

Unidad 16: Algoritmos (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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. • Ejemplos de datos paralelos incluyendo mapeo/reducción. • 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). • Uso de bloqueos, barreras y/o sincronizadores para mantener la vivacidad sin introducir carreras.	▪ Implementar un componente paralelo/distribuido basado en un algoritmo conocido [Implementar] ▪ Escribir un programa de datos paralelos que, por ejemplo, calcule el promedio de un arreglo de números [Escribir] ▪ 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]
Lecturas : [**Pacheco2021**], [**Herlihy2020**]	

Unidad 17: Algoritmos: Dominios de Aplicación (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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). • 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. • 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. • 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. • 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*). • Nube: Agentes de ejecución típicos: Hosts aprovisionados. Mecanismos de comunicación típicos: APIs de servicio. Dominios algorítmicos típicos: Aplicaciones web. Objetivos de ingeniería típicos: Escalabilidad. • 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.	▪ 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] ▪ Mejorar el rendimiento de un componente secuencial introduciendo paralelismo y/o distribución [Crear] ▪ Elegir entre diferentes diseños paralelos/distribuidos para componentes de un sistema dado [Evaluar (valorar)]
Lecturas : [**Pacheco2021**], [**Herlihy2020**]	

Unidad 18: Algoritmos: Dominios Algorítmicos (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Uno o más de los siguientes dominios algorítmicos. Ver también: Estrategias Algorítmicas: • álgebra lineal: Operaciones con vectores y matrices, precisión/estabilidad numérica, aplicaciones en análisis de datos y aprendizaje automático. • Procesamiento de datos: ordenación, búsqueda y recuperación, estructuras de datos concurrentes. • Grafos, búsqueda y combinatoria: Marcado, paralelización de aristas, acotamiento, especulación, análisis basado en redes. • Modelado y simulación: ecuaciones diferenciales; aleatorización, problemas de N-cuerpos, algoritmos genéticos. • Lógica computacional: satisfactibilidad (SAT), programación lógica concurrente. • Gráficos y geometría computacional: Transformaciones, renderizado, trazado de rayos. • 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. • Servicios: Implementar APIs web, moneda electrónica, sistemas de transacción, juegos multijugador.	▪ 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] ▪ Criticar el diseño e implementación de un componente o aplicación existente, o uno desarrollado por compañeros de clase [Críticar (análisis crítico)] ▪ Comparar el rendimiento y la eficiencia energética de múltiples implementaciones de un diseño similar, por ejemplo, multinúcleo versus clúster versus GPU [Comparar]
Lecturas : [**Pacheco2021**], [**Herlihy2020**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán

actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS402. Proyecto de tesis 1 (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS402. Proyecto de tesis 1
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS401. Metodología de la Investigación. (4to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

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

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

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

7. TEMAS

Unidad 1: Profundización en el Estado del Arte Específico (10 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Selección y refinamiento del tópico de investigación en coordinación con el asesor. ▪ Búsqueda exhaustiva multi-fuente: IEEE Xplore, ACM Digital Library, Google Scholar, arXiv y bases de datos especializadas. ▪ Identificación de trabajos seminales y artículos más citados del área. ▪ Construcción del grafo de citaciones y análisis de la evolución temporal del tópico. ▪ Identificación de subáreas, variantes y enfoques principales del campo.	▪ Construir un mapa completo de la literatura de un tópico específico identificando los trabajos fundamentales. [Evaluar] ▪ Analizar la evolución temporal de un área de investigación identificando tendencias y brechas. [Evaluar] ▪ Organizar la literatura en subcategorías coherentes que reflejen la estructura del campo. [Evaluar]
Lecturas : [**ieee**], [**acm**], [**citeseer**]	

Unidad 2: Análisis Crítico y Comparativo de Enfoques (10 horas)	
Resultados esperados: AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis crítico de propuestas: metodología, datasets, métricas y limitaciones de cada enfoque. ▪ Construcción de tablas comparativas de propuestas existentes con criterios objetivos. ▪ Identificación de brechas de investigación (research gaps) no resueltas en el área. ▪ Formulación de la pregunta de investigación propia a partir del análisis del estado del arte. ▪ Marco teórico: síntesis estructurada y coherente del conocimiento relevante del área.	▪ Comparar sistemáticamente múltiples enfoques usando criterios objetivos y cuantificables. [Evaluar] ▪ Identificar brechas de investigación no resueltas en un área específica de computación. [Evaluar] ▪ Formular una pregunta de investigación propia sustentada en el análisis del estado del arte. [Evaluar]
Lecturas : [**ieee**], [**acm**], [**citeseer**]	

Unidad 3: Diseño y Ejecución de Experimentos Preliminares (12 horas)	
Resultados esperados: AG-C05,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diseño experimental: variables, métricas y posibles benchmarks del área. ▪ Reproducción de resultados de trabajos relevantes del estado del arte. ▪ Implementación de baselines para comparación objetiva de enfoques. ▪ Análisis estadístico básico: pruebas de significancia e intervalos de confianza. ▪ Gestión de datos experimentales: datasets, configuraciones, versiones y registro de resultados.	▪ Reproducir los resultados de al menos dos trabajos relevantes del estado del arte del área. [Usar] ▪ Diseñar y ejecutar un experimento comparativo preliminar con baselines definidos. [Usar] ▪ Interpretar y reportar resultados experimentales con rigor estadístico apropiado. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 4: Redacción del Reporte Técnico (12 horas)	
Resultados esperados: AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructura del reporte técnico: introducción, marco teórico, metodología, experimentos, resultados y conclusiones. ▪ Redacción del marco teórico: organización temática y síntesis crítica de la literatura. ▪ Documentación de la metodología y configuración experimental en LaTeX. ▪ Presentación de resultados: tablas, gráficas y análisis comparativos de alta calidad. ▪ Ciclos de revisión: escritura, revisión con asesor e incorporación de retroalimentación.	▪ Redactar un reporte técnico completo con marco teórico y experimentos preliminares. [Usar] ▪ Estructurar y presentar resultados experimentales con visualizaciones claras y precisas. [Usar] ▪ Aplicar ciclos de revisión para mejorar la calidad, coherencia y rigor del documento. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 5: Propuesta de Investigación y Selección de Asesor (8 horas)	
Resultados esperados: AG-C02,AG-C05,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructura de una propuesta de investigación: problema, motivación y objetivos. ▪ Definición del alcance del proyecto de tesis: metodología, contribución esperada y cronograma. ▪ Análisis de viabilidad: recursos, tiempo, riesgos y contingencias del proyecto. ▪ Criterios para la selección del asesor de tesis: experiencia, disponibilidad y alineación temática. ▪ Proceso de formalización del vínculo investigador-asesor en el programa académico.	▪ Elaborar una propuesta de investigación que defina el problema, objetivos y posible solución. [Evaluar] ▪ Seleccionar un asesor con experiencia demostrada en el área de investigación elegida. [Usar] ▪ Definir un plan de tesis viable con objetivos, alcance, cronograma y análisis de riesgos. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 6: Presentación y Defensa del Estado del Arte (8 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Preparación de la presentación oral del reporte técnico ante el comité evaluador. ▪ Técnicas de defensa ante preguntas y cuestionamientos de un comité académico. ▪ Elaboración de póster técnico del estado del arte para presentación en simposio. ▪ Incorporación de retroalimentación del comité evaluador en la revisión final del reporte. ▪ Planificación de la transición a CS403: objetivos, contribución esperada y siguientes pasos.	▪ Presentar y defender oralmente el reporte técnico ante un comité académico con rigor. [Evaluar] ▪ Incorporar constructivamente la retroalimentación del comité en una revisión del reporte. [Usar] ▪ Planificar la transición hacia el proyecto de tesis (CS403) con objetivos y pasos claros. [Usar]
Lecturas : [**ieee**], [**acm**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

EX301. Actividades Extracurriculares (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	EX301. Actividades Extracurriculares
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	2
2.4 Horas	:	1 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- Participar en actividades que complementen su formación profesional.
- Desarrollar habilidades de comunicación y trabajo en equipo.
- Contribuir a iniciativas de impacto social o técnico.

6. RESULTADOS DEL ESTUDIANTE

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. (Familiarity)

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

7. TEMAS

Unidad 1: Actividades Extracurriculares (48 horas)	
Resultados esperados: AG-C01,AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Talleres técnicos o interdisciplinarios. ▪ Asistencia a congresos, seminarios o charlas. ▪ Proyección social (voluntariados, mentorías). ▪ Competencias académicas (hackathons, ferias científicas).	▪ Demostrar participación en al menos 3 actividades [Familiarizarse]. ▪ Elaborar informes reflexivos sobre las experiencias [Usar]. ▪ Colaborar en equipos para proyectos sociales o técnicos [Evaluar].
Lecturas : [**ACM-Extra2019**], [**ABET-Criteria2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

ET201. Formación de Empresas de Base Tecnológica I (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	ET201. Formación de Empresas de Base Tecnológica I
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	FG350. Liderazgo y Desempeño. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este es el primer curso dentro del área de formación de empresas de base tecnológica, tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan elaborar un plan de negocio para una empresa de base tecnológica. El curso está dividido en las siguientes unidades: Introducción, Creatividad, De la idea a la oportunidad, el modelo Canvas, Customer Development y Lean Startup, Aspectos Legales y Marketing, Finanzas de la empresa y Presentación.

Se busca aprovechar el potencial creativo e innovador y el esfuerzo de los alumnos en la creación de nuevas empresas.

5. OBJETIVOS

- Que el alumno conozca como elaborar un plan de negocio para dar inicio a una empresa de base tecnológica.
- Que el alumno sea capaz de realizar, usando modelos de negocio, la concepción y presentación de una propuesta de negocio.

6. RESULTADOS DEL ESTUDIANTE

Sin elementos para esta tabla

7. TEMAS

Unidad 1: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Emprendedor, emprendedurismo e innovación tecnológica ▪ Modelos de negocio ▪ Formación de equipos	▪ Identificar características de los emprendedores [Familiarizarse] ▪ Introducir modelos de negocio [Familiarizarse]
Lecturas : [**byers10**], [**osterwalder10**], [**garzozi14**]	

Unidad 2: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión ▪ Misión ▪ La Propuesta de valor ▪ Creatividad e invención ▪ Tipos y fuentes de innovación ▪ Estrategia y Tecnología ▪ Escala y ámbito	▪ Plantear correctamente la vision y misión de empresa [Usar] ▪ Caracterizar una propuesta de valor innovadora [Evaluar] ▪ Identificar los diversos tipos y fuentes de innovación [Familiarizarse]
Lecturas : [**byers10**], [**blank12**], [**garzozi14**]	

Unidad 3: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estrategia de la Empresa ▪ Barreras ▪ Ventaja competitiva sostenible ▪ Alianzas ▪ Aprendizaje organizacional ▪ Desarrollo y diseño de productos	▪ Conocer estrategias empresariales [Familiarizarse] ▪ Caracterizar barreras y ventajas competitivas [Familiarizarse]
Lecturas : [**byers10**], [**osterwalder10**], [**ries11**], [**garzozi14**]	

Unidad 4: (20 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Creación de un nuevo negocio ▪ El plan de negocio ▪ Canvas ▪ Elementos del Canvas	▪ Conocer los elementos del modelo Canvas [Usar] ▪ Elaborar un plan de negocio basado en el modelo Canvas [Usar]
Lecturas : [**osterwalder10**], [**blank12**], [**garzozi14**]	

Unidad 5: (20 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aceleración versus incubación ▪ Customer Development ▪ Lean Startup	▪ Conocer y aplicar el modelo Customer Development [Usar] ▪ Conocer y aplicar el modelo Lean Startup [Usar]
Lecturas : [**blank12**], [**ries11**], [**garzozi14**]	

Unidad 6: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aspectos Legales y tributarios para la constitución de la empresa ▪ Propiedad intelectual ▪ Patentes ▪ Copyrights y marca registrada ▪ Objetivos de marketing y segmentos de mercado ▪ Investigación de mercado y búsqueda de clientes	▪ Conocer los aspectos legales necesarios para la formación de una empresa tecnológica [Familiarizarse] ▪ Identificar segmentos de mercado y objetivos de marketing [Familiarizarse]
Lecturas : [**byers10**], [**ries11**], [**congreso96**], [**congreso97**], [**garzozi14**]	

Unidad 7: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelo de costos ▪ Modelo de utilidades ▪ Precio ▪ Plan financiero ▪ Formas de financiamiento ▪ Fuentes de capital ▪ Capital de riesgo	▪ Definir um modelo de costos y utilidades [Evaluar] ▪ Conocer las diversas fuentes de financiamento [Familiarizarse]
Lecturas : [**byers10**], [**blank12**], [**garzozi14**]	

Unidad 8: (5 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ The Elevator Pitch ▪ Presentación ▪ Negociación	▪ Conocer las diversas formas de presentar propuestas de negocio [Familiarizarse] ▪ Realizar la presentación de una propuesta de negocio [Usar]
Lecturas : [**byers10**], [**blank12**], [**garzozi14**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI268. Visión Computacional (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI268. Visión Computacional
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI264. Aprendizaje Profundo. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Visión por Computadora es un área clave de la Inteligencia Artificial que permite a las máquinas interpretar y comprender el mundo visual. Este curso cubre técnicas fundamentales para procesamiento de imágenes, extracción de características y reconocimiento de objetos. Con el advenimiento del aprendizaje profundo, la visión por computadora se ha vuelto esencial para vehículos autónomos, imágenes médicas y realidad aumentada. Los estudiantes aprenderán los fundamentos matemáticos y enfoques algorítmicos para resolver problemas de percepción visual del mundo real.

5. OBJETIVOS

- Comprender los fundamentos de la formación de imágenes y la representación digital.
- Dominar técnicas de procesamiento de imágenes como filtrado y detección de bordes.
- Aplicar métodos de extracción de características para identificar patrones clave en imágenes.
- Implementar detección y reconocimiento de objetos usando modelos de aprendizaje profundo.
- Analizar técnicas de visión 3D y estimación de movimiento.

6. RESULTADOS DEL ESTUDIANTE

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

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Introducción a la IA: Conceptos y Comportamiento Inteligente (5 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general de los problemas de IA, Ejemplos de aplicaciones recientes exitosas de IA ▪ Definiciones de agentes con ejemplos (por ejemplo, reactivos, deliberativos) ▪ ¿Qué es el comportamiento inteligente?: • La prueba de Turing y sus defectos • Entrada y salida multimodal • Simulación del comportamiento inteligente • Razonamiento racional versus no racional ▪ Visión general de las Aplicaciones de IA, crecimiento e impacto (económico, social, ético) ▪ Problemas filosóficos ▪ Historia de la IA	▪ Describir la prueba de Turing y el experimento mental de la ”Habitación China” [Explicar] ▪ Diferenciar entre razonamiento/comportamiento óptimo y razonamiento/comportamiento similar al humano [Evaluar] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Explicar]
Lecturas : [**Szeliski2022**], [**Gonzales2018**]	

Unidad 2: Características del Problema y Naturaleza de los Agentes (5 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Características del problema: • Totalmente observable versus parcialmente observable • Agente único versus multiagente • Determinista versus estocástico • Estático versus dinámico • Discreto versus continuo ▪ Naturaleza de los agentes: • Autonomía autónoma, semiautónoma, de iniciativa mixta • Reflexivos, basados en metas y basados en utilidad • Toma de decisiones bajo incertidumbre y con información incompleta • La importancia de la percepción y las interacciones ambientales • Agentes basados en aprendizaje • Agentes corporeizados: ◦ sensores, dinámicas, efectores ▪ Practicar la identificación de características del problema en entornos de ejemplo ▪ Profundidad adicional sobre la naturaleza de los agentes con ejemplos ▪ Profundidad adicional sobre Aplicaciones de IA, Crecimiento e Impacto (económico, social, ético, seguridad)	▪ Enumerar las características de un problema específico [Aplicar]
Lecturas : [**Szeliski2022**], [**Gonzales2018**]	

Unidad 3: Percepción y Visión por Computadora (12 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión por computadora: • Adquisición, representación, procesamiento y propiedades de imágenes • Representación de formas, reconocimiento de objetos y segmentación • Análisis de movimiento • Modelos generativos ▪ Audio y reconocimiento de voz ▪ Tacto y propiocepción ▪ Otras modalidades (por ejemplo, olfato) ▪ Modularidad en el reconocimiento ▪ Enfoques para el reconocimiento de patrones: • Algoritmos de clasificación y medidas de calidad de clasificación • Técnicas estadísticas • Técnicas de aprendizaje profundo	▪ Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir] ▪ 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] ▪ Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar] ▪ 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] ▪ 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] ▪ 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] ▪ 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] ▪ Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar] ▪ 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] ▪ Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir] ▪ Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]
Lecturas : [**Szeliski2022**], [**Prince2012**]	

Unidad 4: Fundamentos de Aprendizaje Automático (5 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: • Aprendizaje supervisado: ◦ Clasificación ◦ Regresión • Aprendizaje por refuerzo • Aprendizaje no supervisado: ◦ Agrupamiento ▪ Ideas fundamentales: • Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. • Fuentes de error e indecidibilidad en el aprendizaje automático ▪ Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: • Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente ▪ El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): • El compromiso entre sesgo (subajuste) - varianza (sobreajuste)	▪ 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] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] ▪ 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] ▪ Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] ▪ Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]
Lecturas : [**Goodfellow2016**], [**Russell2020**]	

Unidad 5: Aprendizaje Automático: Datos, Representaciones y Evaluación (5 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Trabajar con Datos: • Preprocesamiento de datos: ◦ Importancia y peligros de las elecciones de preprocesamiento • Manejo de valores faltantes (imputación, marcar como faltante): ◦ Implicaciones de imputar versus marcar como faltante • Codificación de variables categóricas, codificación de datos de valor real • Normalización/estandarización • énfasis en datos reales, no ejemplos de libros de texto ▪ Representaciones: • Espacios de hipótesis y complejidad • Expansión de características de base simple, como elevar al cuadrado características univariadas • Representaciones de características aprendidas ▪ Evaluación del aprendizaje automático: • Separación de conjuntos de entrenamiento, validación y prueba • Métricas de rendimiento para clasificadores • Estimación del rendimiento en prueba en datos retenidos • Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación • Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones	▪ Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] ▪ Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] ▪ 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] ▪ 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]
Lecturas : [**Goodfellow2016**], [**Russell2020**]	

Unidad 6: Redes Neuronales y Ética (4 horas)	
Resultados esperados: AG-C08	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redes neuronales básicas: • Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos • Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) ▪ ética para el Aprendizaje Automático: • Enfocarse en datos reales, escenarios reales y estudios de caso • Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas	▪ Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]
Lecturas : [**Goodfellow2016**], [**Russell2020**]	

Unidad 7: Formulación del Aprendizaje Automático y Evaluación (3 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: • Función objetivo • Descenso de gradiente • Regularización para evitar sobreajuste (formulación matemática) ▪ Conjuntos de modelos: • Combinación de mayoría ponderada simple ▪ Evaluación de rendimiento: • Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) • Métricas de rendimiento para regresores • Matriz de confusión • Validación cruzada: ◦ Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)	▪ 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] ▪ Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] ▪ Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] ▪ Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] ▪ Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]
Lecturas : [**Goodfellow2016**]	

Unidad 8: Aprendizaje Profundo, RL y Ética (3 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aprendizaje profundo: • Redes feed-forward profundas (solo intuición, sin matemáticas) • Redes neuronales convolucionales (solo intuición, sin matemáticas) • Visualización de representaciones de características aprendidas de redes profundas • Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) ▪ Visión general de los métodos de aprendizaje por refuerzo ▪ Dos o más aplicaciones de algoritmos de aprendizaje automático: • Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego ▪ ética para el Aprendizaje Automático: • Continuar enfocándose en datos reales, escenarios reales y estudios de caso • Privacidad • Equidad • Propiedad intelectual • Explicabilidad	▪ 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] ▪ 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]
Lecturas : [**Goodfellow2016**]	

Unidad 9: Percepción y Visión por Computadora (12 horas)

Resultados esperados: AG-C08

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión por computadora: • Adquisición, representación, procesamiento y propiedades de imágenes • Representación de formas, reconocimiento de objetos y segmentación • Análisis de movimiento • Modelos generativos ▪ Audio y reconocimiento de voz ▪ Tacto y propiocepción ▪ Otras modalidades (por ejemplo, olfato) ▪ Modularidad en el reconocimiento ▪ Enfoques para el reconocimiento de patrones: • Algoritmos de clasificación y medidas de calidad de clasificación • Técnicas estadísticas • Técnicas de aprendizaje profundo	▪ Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir] ▪ 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] ▪ Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar] ▪ 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] ▪ 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] ▪ 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] ▪ 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] ▪ Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar] ▪ 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] ▪ Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir] ▪ Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]
Lecturas : [**Szeliski2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS391. Ingeniería de Software III (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS391. Ingeniería de Software III
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS292. Ingeniería de Software II. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Ingeniería de Software III se enfoca en temas avanzados del ciclo de vida del software, haciendo énfasis en la calidad del software, seguridad por diseño y despliegue en arquitecturas modernas. Este curso prepara a los estudiantes para manejar las complejidades del desarrollo nativo de la nube y la integración de prácticas de seguridad dentro de la canalización DevOps (DevSecOps), asegurando que los productos de software no solo sean funcionales, sino también resilientes y mantenibles.

5. OBJETIVOS

- Implementar métricas avanzadas de aseguramiento de calidad y confiabilidad.
- Integrar prácticas de seguridad a lo largo del ciclo de vida del desarrollo de software.
- Comprender y aplicar principios de DevOps y Despliegue Continuo.
- Diseñar y desplegar aplicaciones nativas de la nube usando microservicios.
- Analizar el impacto de la deuda técnica y las estrategias de rejuvenecimiento de software.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Conceptos de verificación y validación (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conceptos de verificación y validación • Verificación: ¿Estamos construyendo la cosa correctamente? • Validación: ¿Construimos la cosa correcta? ▪ Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? ▪ Objetivos de las pruebas • Usabilidad • Confiabilidad • Conformidad con la especificación • Rendimiento • Seguridad	▪ Explicar por qué las pruebas son importantes [Explicar] ▪ Distinguir entre validación y verificación de programas [Distinguir] ▪ Describir diferentes objetivos de las pruebas [Describir]
Lecturas : [**Sommerville2015**], [**Pressman2019**]	

Unidad 2: Herramientas y Análisis de Pruebas (3 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas y automatización de verificación y validación • Análisis estático • Cobertura de código • Fuzzing • Análisis dinámico y contención de fallos (sanitizadores, etc.) • Registro y seguimiento de fallos ▪ Planificación y generación de pruebas • Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos • Uso de números aleatorios y pseudoaleatorios en las pruebas ▪ Pruebas de sistemas asíncronos, paralelos y concurrentes ▪ Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) ▪ Desarrollo guiado por comportamiento (BDD) • Expresar criterios de aceptación como escenarios Gherkin Given/When/Then • Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo ▪ Pruebas de mutación y análisis estático de calidad de código • Pruebas de mutación para evaluar la efectividad de una suite de pruebas • Detección estática de *code smells* y complejidad ciclomática excesiva	▪ Describir y comparar diferentes herramientas para verificación y validación [Describir] ▪ Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] ▪ Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] ▪ Describir enfoques para la estimación de fallos [Describir] ▪ Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] ▪ Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] ▪ 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] ▪ Describir técnicas para la verificación y validación de artefactos que no son código [Describir] ▪ Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] ▪ Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]
Lecturas : [**Pressman2019**]	

Unidad 3: Pruebas de Rendimiento y Evaluación Comparativa (3 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas de rendimiento y evaluación comparativa • Rendimiento (*throughput*) y latencia • Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) • Aceleración (*speedup*) y escalado (*scaling*) ◦ Ley de Amdahl ◦ Ley de Gustafson ◦ Escalado suave (*soft*) y débil (*weak*) • Identificar y medir figuras de mérito (*figures of merit*) • Cuellos de botella de rendimiento comunes ◦ Limitado por cómputo (*compute-bound*) ◦ Limitado por ancho de banda de memoria ◦ Limitado por latencia • Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) ◦ Estimación de la incertidumbre ◦ Intervalos de confianza • Análisis y presentación (gráficos, etc.) • Técnicas de medición de tiempo (*timing*)	▪ Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] ▪ Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] ▪ Describir cuellos de botella de rendimiento comunes [Describir] ▪ Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] ▪ Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] ▪ 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]
Lecturas : [**Bondi2015**], [**Gregg2020**]	

Unidad 4: Compatibilidad y Versionado de API (12 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ley de Hyrum / Ley de las Interfaces Implícitas ▪ Compatibilidad con versiones anteriores • La compatibilidad no es una propiedad de una sola entidad, es una propiedad de una relación. • La compatibilidad con versiones anteriores debe evaluarse en términos de proveedor + consumidor(es) o con un modelo bien especificado de qué formas de compatibilidad aspira/promete un proveedor. ▪ Control de versiones (versioning) • Versionado Semántico (SemVer) • Desarrollo basado en tronco (*trunk-based development*)	▪ Identificar tanto el comportamiento explícito como el implícito de una interfaz e identificar riesgos potenciales de la Ley de Hyrum [Analizar] ▪ Identificar cambios que pueden considerarse ampliamente ”compatibles con versiones anteriores”, potencialmente con declaraciones explícitas sobre qué uso está o no soportado [Analizar] ▪ Evaluar si un cambio propuesto es lo suficientemente seguro dada la metodología de control de versiones en uso para un proyecto dado [Evaluar]
Lecturas : [**Sommerville2015**]	

Unidad 5: Control de Versiones y CI/CD (12 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Herramientas de prueba, incluyendo herramientas de análisis estático y dinámico. Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Flujo de revisión colaborativa como puerta de calidad • Revisión obligatoria de *pull requests* antes de fusionar • Propiedad del código mediante reglas de tipo CODEOWNERS • Convenciones de mensajes de *commit* (*conventional commits*) • Reglas de protección de ramas que exigen verificaciones antes de fusionar ▪ Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] ▪ 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] ▪ 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] ▪ Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]

Lecturas : [**Kim2016**]

Unidad 6: Principios y Arquitectura de Software (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de sistemas. Confiabilidad del Sistema • Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) • Separación de preocupaciones (separation of concerns) • Ocultación de información (information hiding) • Acoplamiento y cohesión (coupling and cohesion) ▪ Arquitectura de software. Confiabilidad del Sistema • Paradigmas de diseño ◦ Descomposición funcional de arriba hacia abajo/diseño por capas ◦ Arquitectura orientada a datos ◦ Análisis y diseño orientado a objetos ◦ Diseño dirigido por eventos • Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) • Identificar límites y dependencias de componentes ▪ Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema ▪ Malos olores del código (code smells) y otras indicaciones de la calidad del código, distintas de la corrección. Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad	▪ Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] ▪ 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] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] ▪ Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]
Lecturas : [**Fowler2017**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS393. Sistemas de Información (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS393. Sistemas de Información
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS291. Ingeniería de Software I. (5to Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Foundations of Information Systems (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Componentes de los Sistemas de Información: Hardware, Software, Datos, Redes. ▪ Tipos de SI: TPS, MIS, DSS y ESS. ▪ Ventaja competitiva a través de la Tecnología de la Información. ▪ Procesos empresariales e integración de Sistemas de Información.	▪ Identificar los componentes centrales de un sistema de información [Familiarizarse]. ▪ Explicar cómo los SI respaldan los diversos niveles de la jerarquía organizacional [Usar]. ▪ Analizar un caso de negocio para identificar usos estratégicos de TI [Evaluar].
Lecturas : [**Laudon2021**], [**Stair2017**]	

Unidad 2: Data Management and Business Intelligence (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sistemas de gestión de bases de datos y almacenes de datos. ▪ Análisis de Big Data en un contexto empresarial. ▪ Inteligencia Empresarial (BI) y herramientas de Visualización de Datos. ▪ Sistemas de Gestión del Conocimiento.	▪ Diseñar una arquitectura de datos básica para un requisito empresarial [Usar]. ▪ Evaluar la importancia de la calidad de datos y la gobernanza [Evaluar].
Lecturas : [**Laudon2021**], [**Valacich2021**]	

Unidad 3: Enterprise Systems and E-Commerce (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sistemas de Planificación de Recursos Empresariales (ERP). ▪ Gestión de la Cadena de Suministro (SCM) y Gestión de Relaciones con el Cliente (CRM). ▪ Modelos de comercio electrónico: B2B, B2C y C2C. ▪ Comercio móvil y sistemas de pago digital.	▪ Describir las áreas funcionales de un sistema ERP [Familiarizarse]. ▪ Evaluar el impacto del comercio electrónico en los modelos de negocio tradicionales [Evaluar].
Lecturas : [**Stair2017**], [**Valacich2021**]	

Unidad 4: IS Development and Security (12 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ciclo de Vida de Desarrollo de Sistemas (SDLC) y metodologías Ágiles. ▪ Externalización y servicios de SI basados en la nube. ▪ Amenazas a la seguridad de la información y gestión de riesgos. ▪ Problemas éticos y sociales en Sistemas de Información.	▪ Comparar diferentes metodologías de desarrollo de sistemas [Usar]. ▪ Formular pautas éticas para la privacidad y seguridad de los datos [Evaluar].
Lecturas : [**Laudon2021**], [**Stair2017**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS394. Arquitectura de Software y DevOps para Software en la Nube (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS394. Arquitectura de Software y DevOps para Software en la Nube
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS292. Ingeniería de Software II. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

El diseño de software moderno exige combinar principios sólidos de arquitectura con prácticas ágiles de entrega continua y operación en la nube. Este curso integra tres áreas que en la industria actual son inseparables: arquitectura de software, DevOps y computación en la nube.

El estudiante aprenderá a diseñar sistemas a escala usando patrones arquitectónicos reconocidos, a automatizar los ciclos de construcción, prueba y despliegue mediante pipelines de CI/CD, y a provisionar y operar infraestructura en plataformas de nube pública mediante contenedores, orquestación y herramientas de infraestructura como código.

El curso pone especial énfasis en la confiabilidad y observabilidad de sistemas distribuidos, cubriendo métricas, trazas distribuidas, presupuestos de error y prácticas de ingeniería de confiabilidad del sitio (SRE), tal como se aplican en organizaciones de tecnología de alto rendimiento.

5. OBJETIVOS

- Diseñar sistemas de software escalables y mantenibles aplicando patrones arquitectónicos modernos.
- Implementar pipelines de integración y entrega continua (CI/CD) para acelerar y asegurar el ciclo de vida del software.
- Desplegar y gestionar aplicaciones en contenedores sobre plataformas de nube pública usando herramientas estándar de la industria.
- Provisionar infraestructura de nube de forma reproducible y versionada mediante infraestructura como código (IaC).
- Aplicar prácticas de observabilidad y SRE para garantizar la confiabilidad de sistemas en producción.

6. RESULTADOS DEL ESTUDIANTE

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

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

Unidad 1: Principios y Arquitectura de Software (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de sistemas. Confiabilidad del Sistema • Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) • Separación de preocupaciones (separation of concerns) • Ocultación de información (information hiding) • Acoplamiento y cohesión (coupling and cohesion) ▪ Arquitectura de software. Confiabilidad del Sistema • Paradigmas de diseño ◦ Descomposición funcional de arriba hacia abajo/diseño por capas ◦ Arquitectura orientada a datos ◦ Análisis y diseño orientado a objetos ◦ Diseño dirigido por eventos • Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) • Identificar límites y dependencias de componentes ▪ Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema ▪ Modelos estructurales y de comportamiento de diseños de software ▪ Modelado de diseño, por ejemplo con diagramas de clases, diagramas de entidad-relación o diagramas de secuencia ▪ Principios de diseño de API • Consistencia ◦ Las API consistentes son más fáciles de aprender y menos propensas a errores ◦ La consistencia es tanto interna (entre diferentes partes de la API) como externa (siguiendo patrones comunes de API) • Capacidad de composición (*composability*) • Documentación de contratos ◦ Las operaciones de la API deben describir su efecto en el sistema, pero generalmente no su implementación ◦ Precondiciones, postcondiciones e invariantes • Capacidad de expansión (*expandability*) • Reporte de errores ◦ Los errores deben ser claros, predecibles y accionables	▪ Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] ▪ Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar] ▪ 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] ▪ Traducir un diseño de software en lenguaje natural a diagramas de clases [Traducir]

Unidad 2: Calidad y Evaluación del Diseño (8 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Evaluar compensaciones de diseño (tradeoffs) (por ejemplo, eficiencia vs confiabilidad, seguridad vs usabilidad) ▪ Estilos arquitectónicos: en capas, orientado a eventos, microkernel, orientado a servicios, pipeline. ▪ Atributos de calidad: disponibilidad, escalabilidad, rendimiento, seguridad, mantenibilidad. ▪ Trazabilidad de requisitos • Comprender qué requisitos son satisfechos por un diseño ▪ Medición y análisis de la calidad del diseño	▪ Contrastar dos diseños de software a través de diferentes cualidades, como eficiencia o usabilidad [Contrastar] ▪ Identificar qué requisitos son satisfechos por un diseño de software proporcionado [Analizar] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] ▪ Seleccionar el estilo arquitectónico más adecuado dados los atributos de calidad requeridos por un sistema. [Evaluar]
Lecturas : [**Richards2020**]	

Unidad 3: Diseño de API y Modelado (6 horas)

Resultados esperados: AG-C09,AG-C11

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de microservicios: cohesión, acoplamiento débil y fronteras de servicio. ▪ Descomposición de monolitos: diseño orientado al dominio (DDD). ▪ Principios de diseño de API • Consistencia ◦ Las API consistentes son más fáciles de aprender y menos propensas a errores ◦ La consistencia es tanto interna (entre diferentes partes de la API) como externa (siguiendo patrones comunes de API) • Capacidad de composición (*composability*) • Documentación de contratos ◦ Las operaciones de la API deben describir su efecto en el sistema, pero generalmente no su implementación ◦ Precondiciones, postcondiciones e invariantes • Capacidad de expansión (*expandability*) • Reporte de errores ◦ Los errores deben ser claros, predecibles y accionables ◦ La entrada que no coincida con el contrato debe producir un error ◦ Los errores que pueden gestionarse de manera confiable sin reportarse deben gestionarse ▪ API Gateway y malla de servicios (*service mesh*). ▪ Patrones de resiliencia: Circuit Breaker, Retry, Bulkhead, Timeout. ▪ Almacenes de Datos • Estructuras de datos mantenidas cooperativamente que implementan mapas y TADs relacionados. • Variedades: Propiedad exclusiva, compartida, fragmentada, replicada, inmutable, versionada.	▪ Diseñar una API para un solo componente de un sistema de software grande, incluyendo la identificación y documentación de las invariantes, el contrato y las condiciones de error de cada operación [Diseñar] ▪ Evaluar una descripción de API en términos de consistencia, capacidad de composición y capacidad de expansión [Evaluar] ▪ Diseñar un sistema basado en microservicios descomponiendo una aplicación monolítica en servicios independientes. [Usar] ▪ Aplicar patrones de resiliencia para tolerar fallos en sistemas distribuidos. [Usar]
Lecturas : [**Newman2021**]	

Unidad 4: Virtualización (6 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Contenedores y su comparación con máquinas virtuales. ▪ Tipos de virtualización (incluyendo hardware/software, SO, servidor, servicio, red). Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento ▪ Virtualización portátil; emulación vs aislamiento. Ver también: Latencia, Caché y Jerarquía de Memoria, Virtualización y Aislamiento ▪ Máquinas virtuales y escapes de contenedores, peligros desde una perspectiva de seguridad. Ver también: Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad ▪ Dockerfile: buenas prácticas para construir imágenes eficientes y seguras. ▪ Composición de aplicaciones multi-contenedor con docker-compose.	▪ Explicar la diferencia entre emulación y aislamiento [Explicar] ▪ Evaluar las compensaciones de la virtualización [Evaluar] ▪ Construir y desplegar aplicaciones en contenedores usando Docker. [Usar] ▪ Aplicar buenas prácticas de seguridad en la construcción de imágenes Docker. [Usar]
Lecturas : [**Burns2022**]	

Unidad 5: Coordinación (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Arquitectura de Kubernetes: plano de control y nodos trabajadores. ▪ Objetos fundamentales: Pods, Deployments, Services, ConfigMaps, Secrets. ▪ Escalado horizontal automático (*HPA*) y autocuración. ▪ Dependencias • La iniciación o progreso de una actividad puede depender de otras actividades, para evitar condiciones de carrera, garantizar la terminación o cumplir otros requisitos. • Garantizar el progreso evitando ciclos de dependencia, usando condiciones monótonas, eliminando dependencias no esenciales. ▪ Gestión de paquetes con Helm.	▪ Desplegar y gestionar aplicaciones en contenedores sobre un clúster de Kubernetes. [Usar] ▪ Configurar el escalado automático y la autocuración para cargas de trabajo nativas de la nube. [Usar] ▪ Modificar un sistema concurrente para usar un almacén de datos más escalable, confiable o disponible [Crear]
Lecturas : [**Burns2022**]	

Unidad 6: Control de Versiones y CI/CD (6 horas)

Resultados esperados: AG-C11,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ 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 ▪ Conceptos y mecanismos de integración de herramientas. Prácticas de Desarrollo de Software ▪ Cultura DevOps: flujo, retroalimentación y aprendizaje continuo.	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ 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] ▪ Demostrar la capacidad de usar herramientas de software para apoyar el desarrollo de un producto de software de tamaño mediano [Demostrar]
Lecturas : [**Kim2021**]	

Unidad 7: Control de Versiones y CI/CD (6 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de IaC: reproducibilidad, idempotencia y versionamiento de infraestructura. ▪ Enfoques declarativo vs. imperativo. ▪ Terraform: proveedores, recursos, módulos y gestión de estado. ▪ Herramientas nativas de nube: AWS CloudFormation, Azure ARM Templates, GCP Deployment Manager. ▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes.	▪ Escribir y aplicar configuraciones de Terraform para aprovisionar infraestructura en la nube. [Usar] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ Comparar herramientas de IaC según sus capacidades de gestión de estado y modularidad. [Evaluar]
Lecturas : [**Morris2021**], [**Kim2021**]	

Unidad 8: Aplicaciones en Red (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de servicio en la nube: IaaS, PaaS, SaaS, FaaS. ▪ Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación ▪ Principales proveedores: AWS, Google Cloud, Azure — servicios clave. ▪ Servicios de cómputo: máquinas virtuales, funciones serverless, contenedores administrados. ▪ Servicios de almacenamiento: objetos, bloque y bases de datos administradas. ▪ Redes en la nube: VPCs, balanceadores de carga, CDN.	▪ Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar] ▪ Comparar los modelos IaaS, PaaS, SaaS y FaaS para seleccionar el más adecuado a una arquitectura dada. [Evaluar] ▪ Diseñar una aplicación nativa de la nube usando servicios administrados de un proveedor principal. [Usar]
Lecturas : [**Kim2021**], [**Burns2022**]	

Unidad 9: Ingeniería de Confiabilidad (6 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Análisis basado en mediciones de la confiabilidad del software - telemetría, monitorización y alertas, paneles de control, métricas de calificación de lanzamientos, etc. ▪ Los tres pilares de la observabilidad: métricas, logs y trazas distribuidas. ▪ Trazas distribuidas e identificadores de correlación. ▪ Prácticas de ingeniería de confiabilidad de software - incluyendo revisiones, pruebas, verificación de modelos práctica (practical model checking) ▪ Estrategias de alertas y prácticas de guardia (*on-call*).	▪ Describir cómo determinar el nivel de confiabilidad requerido por un sistema de software [Describir] ▪ Identificar métodos que conducirán a la realización de una arquitectura de software que logre un nivel especificado de confiabilidad [Analizar] ▪ Implementar observabilidad en una aplicación de nube usando métricas, logs y trazas distribuidas. [Usar]
Lecturas : [**Kim2021**]	

Unidad 10: Confiabilidad del Sistema (6 horas)	
Resultados esperados: AG-C09,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Concepto de confiabilidad como probabilidad de fallo o tiempo medio entre fallos, y fallas (faults) como causa de fallos (failures) ▪ 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. ▪ Técnicas y modelos de tolerancia a fallos de software • Diferencias contextuales en la tolerancia a fallos (por ejemplo, hacer fallar un sistema crítico para la aviación se evita firmemente, hacer fallar un sistema de procesamiento de datos antes de que se escriban datos corruptos en el almacenamiento es muy valioso) ▪ Confiabilidad mediante redundancia: • verificación y reintento (*check and retry*) Soporte de Fiabilidad, Tolerancia a fallos • codificación redundante (códigos de corrección de errores, CRC, FEC, RAID) Jerarquía de Memoria, Bases de Datos Distribuidas/Computación en la Nube, Soporte de Fiabilidad • duplicación/reflejo (*mirroring*)/réplicas Bases de Datos Distribuidas/Computación en la Nube ▪ Confiabilidad del software, confiabilidad del sistema y comportamiento de fallos ▪ Indicadores de nivel de servicio (SLI), objetivos (SLO) y acuerdos (SLA).	▪ Identificar formas de aplicar redundancia para lograr tolerancia a fallos [Analizar] ▪ Identificar dependencias de punto único de fallo (single-point-of-failure, SPF) en un diseño de sistema [Analizar] ▪ Explicar los problemas que existen para lograr niveles muy altos de confiabilidad [Explicar] ▪ Definir SLIs, SLOs y presupuestos de error para un servicio en producción. [Usar]
Lecturas : [**Kim2021**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán

actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS395. Calidad de Software (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS395. Calidad de Software
2.2 Semestre	:	8vo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS292. Ingeniería de Software II. (7mo Sem)

3. PROFESORES

Titular

- Ernesto Cuadros-Vargas <ecuadrosv@uni.edu.pe>
  - Doctor en Ciencia de la Computación, ICMC-USP, Brasil, 2004.
  - Master en Ciencia de la Computación, ICMC-USP, Brasil, 1998.

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Entregar software meramente ”funcionalza no es suficiente para la industria: las organizaciones modernas exigen sistemas confiables, seguros, mantenibles y construidos mediante procesos de ingeniería disciplinados y auditables. Este curso enseña la calidad del software como una disciplina de ingeniería práctica y de primer nivel, ejercida mediante las mismas herramientas y flujos de trabajo usados por equipos profesionales en GitHub: protección de ramas, revisión de código obligatoria, pruebas automatizadas en todos los niveles, integración continua/entrega continua (CI/CD), análisis estático y observabilidad en producción.

Los estudiantes avanzan desde modelos fundamentales de calidad hacia flujos de trabajo colaborativos avanzados, estrategias de pruebas automatizadas (unitarias, de integración, de extremo a extremo y guiadas por comportamiento), pipelines de CI/CD, análisis estático y refactorización, y aspectos de calidad en producción como seguridad de la cadena de suministro, rendimiento e ingeniería de confiabilidad. El curso cierra con énfasis en la cultura DevOps: las prácticas y valores compartidos del equipo (Definition of Done, postmortems sin culpa, aprendizaje continuo) que sostienen la calidad a escala.

5. OBJETIVOS

- Evaluar la calidad del software usando un modelo de características de calidad reconocido internacionalmente y razonar sobre el costo de la mala calidad y la deuda técnica.
- Aplicar un flujo de trabajo profesional de Git/GitHub basado en revisión, incluyendo protección de ramas, CODEOWNERS y conventional commits.
- Diseñar e implementar suites de pruebas unitarias, de integración, de extremo a extremo y guiadas por comportamiento, incluyendo desarrollo guiado por pruebas.
- Construir y operar pipelines de CI/CD que impongan puertas de calidad automatizadas, escaneo de dependencias y versionado semántico.
- Aplicar análisis estático, pruebas de mutación y técnicas de refactorización para identificar y eliminar code smells y deuda técnica.
- Analizar aspectos de calidad en producción, incluyendo pruebas de rendimiento y prácticas de confiabilidad/observabilidad.

- Adoptar prácticas de cultura DevOps – Definition of Done, postmortems sin culpa y aprendizaje continuo del equipo – para sostener la calidad.

6. RESULTADOS DEL ESTUDIANTE

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

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Calidad y Evaluación del Diseño (4 horas)

Resultados esperados: AG-C09,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diseño de datos Modelado de Datos • Estructuras de datos • Sistemas de almacenamiento ▪ Trazabilidad de requisitos • Comprender qué requisitos son satisfechos por un diseño ▪ Modelado de diseño, por ejemplo con diagramas de clases, diagramas de entidad-relación o diagramas de secuencia ▪ Medición y análisis de la calidad del diseño ▪ Principios de diseño y codificación seguros Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad • Principio de privilegio mínimo • Principio de valores predeterminados seguros ante fallos (fail-safe defaults) • Principio de aceptabilidad psicológica ▪ Evaluar compensaciones de diseño (tradeoffs) (por ejemplo, eficiencia vs confiabilidad, seguridad vs usabilidad) ▪ Modelo de características de calidad ISO/IEC 25010 y costo de la mala calidad • Características de calidad: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad, portabilidad • Costo de la mala calidad y la deuda técnica como preocupación de negocio/económica, no solo técnica	▪ Diseñar un conjunto de estructuras de datos para implementar una superficie de API proporcionada [Diseñar] ▪ Identificar qué requisitos son satisfechos por un diseño de software proporcionado [Analizar] ▪ Traducir un diseño de software en lenguaje natural a diagramas de clases [Traducir] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor los principios de privilegio mínimo y valores predeterminados seguros ante fallos [Crear] ▪ Contrastar dos diseños de software a través de diferentes cualidades, como eficiencia o usabilidad [Contrastar] ▪ Evaluar un sistema de software frente a las características de calidad ISO/IEC 25010 y estimar el costo de negocio de la mala calidad/deuda técnica [Evaluar]
Lecturas : [**ISO25010**], [**Sommerville2015**]	

Unidad 2: Control de Versiones y CI/CD (12 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Herramientas de prueba, incluyendo herramientas de análisis estático y dinámico. Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Flujo de revisión colaborativa como puerta de calidad • Revisión obligatoria de *pull requests* antes de fusionar • Propiedad del código mediante reglas de tipo CODEOWNERS • Convenciones de mensajes de *commit* (*conventional commits*) • Reglas de protección de ramas que exigen verificaciones antes de fusionar ▪ Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] ▪ 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] ▪ 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] ▪ Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]
Lecturas : [**Kim2021**]	

Unidad 3: Planificación y Tipos de Pruebas (8 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Tipos de pruebas • Unitarias • Integración • Validación • Sistema ▪ Diferencias estilísticas entre el código de prueba y el código de producción: DAMP vs DRY - se justifica más duplicación en el código de prueba. ▪ Planificación y generación de pruebas • Generación de casos de prueba, a partir de modelos formales, especificaciones, etc. • Cobertura de pruebas ◦ Matrices de prueba ◦ Cobertura de código - ¿cuánto del código está probado? ◦ Cobertura de entorno - ¿cuántas arquitecturas de hardware, sistemas operativos, navegadores, etc. están probados? • Datos y entradas de prueba	▪ Comparar y contrastar los diferentes tipos y niveles de pruebas (regresión, unitarias, integración, sistemas y aceptación) [Comparar] ▪ Describir técnicas para crear un plan de pruebas y generar casos de prueba [Describir] ▪ Crear un plan de pruebas para un segmento de código de tamaño mediano que incluya una matriz de pruebas y generación de datos y entradas de prueba [Crear] ▪ Implementar un plan de pruebas para un segmento de código de tamaño mediano [Implementar]
Lecturas : [**Sommerville2015**]	

Unidad 4: Prácticas Avanzadas de Pruebas (4 horas)

Resultados esperados: AG-C11

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Desarrollo de pruebas • Desarrollo guiado por pruebas • Pruebas orientadas a objetos, uso de mocks e inyección de dependencias • Técnicas de prueba de caja opaca (anteriormente, caja negra) y caja transparente (anteriormente, caja blanca) • Herramientas de prueba, incluyendo cobertura de código, análisis estático y fuzzing ▪ Verificación y validación en el ciclo de desarrollo • Revisiones de código • Automatización de pruebas, incluyendo automatización de herramientas • Pruebas pre-commit y post-commit • Compensaciones entre cobertura de pruebas y rendimiento/latencia de las pruebas • Seguimiento y priorización de defectos: reproducibilidad de los defectos reportados ▪ Desafíos de verificación y validación específicos del dominio • Pruebas de rendimiento y evaluación comparativa (benchmarking) • Asincronía, paralelismo y concurrencia • Crítico para la seguridad (*safety-critical*) • Numérico	▪ Identificar los principios fundamentales de los métodos de desarrollo guiado por pruebas y explicar el papel de las pruebas automatizadas en estos métodos [Analizar] ▪ Discutir problemas relacionados con las pruebas de software orientado a objetos [Debatir] ▪ Describir el uso de mocks y la inyección de dependencias y su aplicación [Describir] ▪ Realizar, como parte de una actividad en equipo, una revisión de código de un segmento de código de tamaño mediano [Evaluar] ▪ Describir el papel que las herramientas pueden desempeñar en la validación del software [Describir] ▪ Automatizar las pruebas en un proyecto de software pequeño [Diseñar] ▪ Explicar los roles, pros y contras de las pruebas pre-commit y post-commit [Explicar] ▪ Discutir las compensaciones entre la cobertura de pruebas y el rendimiento/latencia de las pruebas y cómo esto puede afectar la verificación [Debatir] ▪ Usar una herramienta de seguimiento de defectos para gestionar defectos de software en un proyecto de software pequeño [Usar] ▪ Discutir las limitaciones de las pruebas en ciertos dominios [Debatir]
Lecturas : [**Beck2002**]	

Unidad 5: Herramientas y Análisis de Pruebas (8 horas)

Resultados esperados: AG-C11

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas y automatización de verificación y validación • Análisis estático • Cobertura de código • Fuzzing • Análisis dinámico y contención de fallos (sanitizadores, etc.) • Registro y seguimiento de fallos ▪ Planificación y generación de pruebas • Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos • Uso de números aleatorios y pseudoaleatorios en las pruebas ▪ Pruebas de sistemas asíncronos, paralelos y concurrentes ▪ Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) ▪ Desarrollo guiado por comportamiento (BDD) • Expresar criterios de aceptación como escenarios Gherkin Given/When/Then • Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo ▪ Pruebas de mutación y análisis estático de calidad de código • Pruebas de mutación para evaluar la efectividad de una suite de pruebas • Detección estática de *code smells* y complejidad ciclomática excesiva	▪ Describir y comparar diferentes herramientas para verificación y validación [Describir] ▪ Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] ▪ Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] ▪ Describir enfoques para la estimación de fallos [Describir] ▪ Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] ▪ Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] ▪ 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] ▪ Describir técnicas para la verificación y validación de artefactos que no son código [Describir] ▪ Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] ▪ Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]
Lecturas : [**Wynne2012**]	

Unidad 6: Compatibilidad y Versionado de API (4 horas)

Resultados esperados: AG-C11,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ley de Hyrum / Ley de las Interfaces Implícitas ▪ Compatibilidad con versiones anteriores • La compatibilidad no es una propiedad de una sola entidad, es una propiedad de una relación. • La compatibilidad con versiones anteriores debe evaluarse en términos de proveedor + consumidor(es) o con un modelo bien especificado de qué formas de compatibilidad aspira/promete un proveedor. ▪ Control de versiones (versioning) • Versionado Semántico (SemVer) • Desarrollo basado en tronco (*trunk-based development*)	▪ Identificar tanto el comportamiento explícito como el implícito de una interfaz e identificar riesgos potenciales de la Ley de Hyrum [Analizar] ▪ Identificar cambios que pueden considerarse ampliamente "compatibles con versiones anteriores", potencialmente con declaraciones explícitas sobre qué uso está o no soportado [Analizar] ▪ Evaluar si un cambio propuesto es lo suficientemente seguro dada la metodología de control de versiones en uso para un proyecto dado [Evaluar]

Lecturas : [**Sommerville2015**]

Unidad 7: Técnicas de Refactorización (4 horas)

Resultados esperados: AG-C09,AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Refactorización • Patrones de refactorización estándar (renombrar, integrar, extraer, etc.) • Uso de herramientas de refactorización en el IDE • Aplicación de herramientas de análisis estático (para identificar código que necesita refactorización, generar cambios, etc.) • Valor de la refactorización como remedio para la deuda técnica ▪ Refactorización "a Gran Escala": técnicas cuando un cambio de refactorización es demasiado grande para comprometerlo de forma segura (proyectos grandes), o cuando es imposible sincronizar el cambio entre proveedor + todos los consumidores (múltiples repositorios, consumidores con código privado). • Expresar tanto las API antiguas como las nuevas para que puedan coexistir. • Minimizar el tamaño de los cambios de comportamiento. • ¿Por qué se requieren estas técnicas?. Por ejemplo, "consumidores de API que puedo ver" vs "consumidores que no puedo ver").	▪ Considerar las entradas de herramientas de análisis estático y/o los principios de Diseño de Software para identificar código que necesita refactorización [Analizar] ▪ Refactorizar la implementación de una interfaz para mejorar el diseño, claridad, etc., con impacto mínimo/cero en los usuarios existentes [Crear] ▪ Planificar una refactorización compleja de múltiples pasos para cambiar el comportamiento predeterminado de una API de manera segura [Planear]

Lecturas : [**Fowler2017**]

Unidad 8: Pruebas de Rendimiento y Evaluación Comparativa (4 horas)

Resultados esperados:

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas de rendimiento y evaluación comparativa • Rendimiento (*throughput*) y latencia • Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) • Aceleración (*speedup*) y escalado (*scaling*) ◦ Ley de Amdahl ◦ Ley de Gustafson ◦ Escalado suave (*soft*) y débil (*weak*) • Identificar y medir figuras de mérito (*figures of merit*) • Cuellos de botella de rendimiento comunes ◦ Limitado por cómputo (*compute-bound*) ◦ Limitado por ancho de banda de memoria ◦ Limitado por latencia • Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) ◦ Estimación de la incertidumbre ◦ Intervalos de confianza • Análisis y presentación (gráficos, etc.) • Técnicas de medición de tiempo (*timing*)	▪ Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] ▪ Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] ▪ Describir cuellos de botella de rendimiento comunes [Describir] ▪ Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] ▪ Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] ▪ 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]
Lecturas : [**Bondi2015**], [**Gregg2020**]	

Unidad 9: Ingeniería de Confiabilidad (4 horas)	
Resultados esperados: AG-C11,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Modelos de confiabilidad de software ▪ Técnicas y modelos de tolerancia a fallos de software • Diferencias contextuales en la tolerancia a fallos (por ejemplo, hacer fallar un sistema crítico para la aviación se evita firmemente, hacer fallar un sistema de procesamiento de datos antes de que se escriban datos corruptos en el almacenamiento es muy valioso) ▪ Prácticas de ingeniería de confiabilidad de software - incluyendo revisiones, pruebas, verificación de modelos práctica (practical model checking) ▪ Identificación de dominios de fallos dependientes e independientes, y su impacto en la confiabilidad del sistema ▪ Análisis basado en mediciones de la confiabilidad del software - telemetría, monitorización y alertas, paneles de control, métricas de calificación de lanzamientos, etc.	▪ Demostrar la capacidad de aplicar múltiples métodos para desarrollar estimaciones de confiabilidad para un sistema de software [Demostrar] ▪ Identificar métodos que conducirán a la realización de una arquitectura de software que logre un nivel especificado de confiabilidad [Analizar] ▪ Identificar formas de aplicar redundancia para lograr tolerancia a fallos [Analizar] ▪ Identificar dependencias de punto único de fallo (single-point-of-failure, SPF) en un diseño de sistema [Analizar]
Lecturas : [**Kim2021**], [**Forsgren2018**]	

Unidad 10: Construcción a Escala y Proceso (4 horas)

Resultados esperados: AG-C12

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas a mayor escala • Dobles de prueba (test doubles) (*stubs*, *mocks*, *fakes*) • Inyección de dependencias (*dependency injection*) ▪ Secuenciación del trabajo, incluyendo identificación de dependencias, hitos y retiro de riesgo • Identificación de dependencias: identificar las dependencias entre diferentes tareas • Hitos: una colección de tareas que sirve como marcador de progreso cuando se completa. Idealmente, el hito abarca una unidad útil de funcionalidad. • Retiro de riesgo: identificar qué elementos de un proyecto son riesgosos y priorizar la finalización de tareas que aborden esos riesgos. ▪ Problemas potenciales de seguridad en programas Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada • Desbordamientos de búfer y otros tipos de desbordamientos • Condiciones de carrera (race conditions) • Inicialización incorrecta, incluida la elección de privilegios • Validación de entrada (input validation) ▪ Documentación (autogenerada) ▪ Contexto de desarrollo: "campo verde" (*green field*) vs base de código existente • Análisis de impacto del cambio (*change impact analysis*) • Realización del cambio (*change actualization*) ▪ Gestión de lanzamientos (release management) ▪ Prácticas de DevOps • Definition of Done como umbral de calidad compartido para el trabajo terminado • Postmortems sin culpa (*blameless postmortems*) para aprender de incidentes/fallas • Cultura de calidad en equipo: propiedad compartida, retroalimentación y aprendizaje continuo	▪ Reescribir un programa simple para eliminar vulnerabilidades comunes, como desbordamientos de búfer, desbordamientos de enteros y condiciones de carrera [Aplicar] ▪ Escribir un componente de software que realice alguna tarea no trivial y sea resistente a errores de entrada y de tiempo de ejecución [Escribir]

Lecturas : [**Forsgren2018**]

Unidad 11: Proyecto Final: Demostración de un Pipeline de Calidad de Extremo a Extremo (8 horas)	
Resultados esperados: AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integración de todas las prácticas del curso en un único repositorio en vivo: protección de ramas, revisión obligatoria de PRs, suites de pruebas automatizadas (unitarias, de integración, E2E/BDD), pipeline de CI/CD con puertas de calidad, escaneo de dependencias, análisis estático y observabilidad. ▪ Revisión en vivo del repositorio: demostración y defensa del pipeline de calidad del equipo ante compañeros/instructor. ▪ Retrospectiva y postmortem sin culpa del recorrido de calidad del proyecto.	▪ Integrar control de versiones, pruebas automatizadas, CI/CD, análisis estático y observabilidad en un único pipeline de calidad de extremo a extremo para un repositorio real. [Usar] ▪ Defender el diseño y la efectividad del pipeline de calidad de un equipo mediante una revisión en vivo del repositorio. [Usar] ▪ Realizar un postmortem sin culpa para identificar mejoras de proceso para proyectos futuros. [Evaluar]
Lecturas : [**Forsgren2018**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS370. Big Data (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS370. Big Data
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS3P1. Computación Paralela y Distribuída. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

En la era actual, el volumen, velocidad y variedad de los datos superan las capacidades de los sistemas tradicionales de almacenamiento y procesamiento. Este curso introduce los conceptos y tecnologías fundamentales del Big Data. Cubre sistemas de archivos distribuidos, frameworks de procesamiento paralelo, bases de datos NoSQL y flujo de datos en tiempo real. Los estudiantes aprenderán a diseñar arquitecturas escalables para extraer valor de conjuntos de datos masivos.

5. OBJETIVOS

- Comprender las 5 V's del Big Data y su impacto en el diseño de sistemas.
- Diseñar e implementar soluciones escalables usando el paradigma MapReduce.
- Dominar el uso de sistemas de archivos distribuidos como HDFS.
- Analizar diferentes patrones arquitectónicos para Big Data (Lambda y Kappa).
- Aplicar técnicas de procesamiento de flujos para análisis de datos en tiempo real.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Conceptos Fundamentales de Sistemas de Bases de Datos (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Propósito y ventajas de los sistemas de bases de datos ▪ Componentes de los sistemas de bases de datos ▪ Diseño de funciones fundamentales de DBMS (por ejemplo, mecanismos de consulta, gestión de transacciones, gestión de búfer, métodos de acceso) ▪ Arquitectura de base de datos, independencia de datos y abstracción de datos ▪ Gestión de transacciones ▪ Normalización ▪ Enfoques para gestionar grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce) ▪ Cómo soportar aplicaciones solo de CRUD ▪ Bases de datos distribuidas/sistemas basados en la nube ▪ Datos estructurados, semiestructurados y no estructurados ▪ Uso de un lenguaje de consulta declarativo ▪ Sistemas que soportan contenido estructurado y/o de flujo	▪ Identificar al menos cuatro ventajas que proporciona el uso de un sistema de base de datos [Analizar] ▪ Enumerar los componentes de un sistema de base de datos (relacional) [Enumerar] ▪ Seguir una consulta a medida que es procesada por los componentes de un sistema de base de datos (relacional) [Analizar] ▪ Defender el valor de la independencia de datos [Defender] ▪ Componer una consulta simple de selección-proyección-unión en SQL [Componer] ▪ Enumerar las cuatro propiedades de un gestor de transacciones correcto [Enumerar] ▪ Describir las ventajas de eliminar datos duplicados repetidos [Describir] ▪ Esbozar cómo MapReduce usa paralelismo para procesar datos eficientemente [Esquematizar/Esbozar] ▪ Evaluar las diferencias entre bases de datos estructuradas y semiestructuradas/no estructuradas [Evaluar]
Lecturas : [**White2015**], [**Karau2017**]	

Unidad 2: Bases de Datos Distribuidas/Computación en la Nube (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ DBMS Distribuido: • Almacenamiento de datos distribuido • Procesamiento de consultas distribuido • Modelo de transacción distribuida • Soluciones homogéneas y heterogéneas • Bases de datos distribuidas cliente-servidor ▪ DBMS Paralelo: • Arquitecturas de DBMS paralelo: memoria compartida, disco compartido, nada compartido • Aceleración y escalamiento, por ejemplo, uso del modelo de procesamiento MapReduce • Replicación de datos y modelos de consistencia débil	▪ Describir los componentes clave de un DBMS distribuido, incluyendo almacenamiento de datos distribuido, procesamiento de consultas y gestión de transacciones [Describir] ▪ Analizar las ventajas y desventajas entre arquitecturas de DBMS paralelo: memoria compartida, disco compartido y nada compartido [Analizar] ▪ Describir estrategias de replicación de datos y modelos de consistencia débil en sistemas de bases de datos distribuidas [Describir]
Lecturas : [**White2015**], [**Lin2010**]	

Unidad 3: Sistemas NoSQL (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ ¿Por qué NoSQL? (por ejemplo, Desajuste de impedancia entre Aplicación [CRUD] y RDBMS) ▪ Modelo de datos Clave-Valor y Documento ▪ Sistemas de almacenamiento (por ejemplo, sistemas Clave-Valor, Lagos de Datos (*Data Lakes*)) ▪ Modelos de Distribución (Fragmentación y Replicación) ▪ Bases de Datos de Grafos ▪ Modelos de Consistencia (Actualización y Lectura, consistencia de quórum, teorema CAP) ▪ Modelo de procesamiento (por ejemplo, MapReduce, map-reduce multi-etapa, map-reduce incremental) ▪ Estudios de Caso: Sistema de almacenamiento en la nube (por ejemplo, S3); Bases de datos de grafos; "Cuándo no usar NoSQL"	▪ Desarrollar un caso de uso para el uso de NoSQL sobre RDBMS [Crear] ▪ Describir las características definitorias detrás de los modelos de datos basados en Clave-Valor y Documentos [Describir]
Lecturas : [**Sadalage2012**], [**Kleppmann2017**]	

Unidad 4: Análisis de Datos (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Técnicas exploratorias de datos (motivación, representación, estadísticas descriptivas, visualizaciones) ▪ Ciclo de vida de ciencia de datos: comprensión del negocio, comprensión de datos, preparación de datos, modelado, evaluación, despliegue y aceptación del usuario ▪ Algoritmos de minería de datos y aprendizaje automático: por ejemplo, clasificación, agrupamiento, asociación, regresión ▪ Adquisición y gobernanza de datos ▪ Consideraciones de seguridad y privacidad de datos ▪ Equidad y sesgo de datos ▪ Técnicas de visualización de datos y su uso en análisis de datos ▪ Resolución de Entidades	▪ Describir varios enfoques de exploración de datos, incluyendo visualización, para comprender conjuntos de datos no familiares [Describir] ▪ Aplicar varios enfoques de exploración de datos para comprender conjuntos de datos no familiares [Aplicar] ▪ Describir algoritmos básicos de aprendizaje automático/minería de datos y cuándo son apropiados para su uso [Describir] ▪ Aplicar varios algoritmos de aprendizaje automático/minería de datos [Aplicar] ▪ Describir consideraciones legales y éticas en la adquisición, uso y modificación de conjuntos de datos [Describir] ▪ Describir problemas de equidad y sesgo en la recolección y uso de datos [Describir]
Lecturas : [**zaharia2012resilient**], [**chambers2018spark**], [**Kleppmann2017**], [**Psaltis2017**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.

- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS400. Prácticas Pre-profesionales (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS400. Prácticas Pre-profesionales
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	2
2.4 Horas	:	1 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	110Cr

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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. (Usage)

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

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

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

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

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

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

7. TEMAS

Unidad 1: Evaluación de Desempeño en Entorno Laboral (48 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Integración del estudiante en un equipo de trabajo profesional. ▪ Aplicación de conocimientos técnicos en proyectos reales. ▪ Evaluación de competencias éticas y profesionales en el ámbito laboral. ▪ Presentación de informes y retroalimentación por parte de la empresa y la universidad.	▪ Demostrar habilidades técnicas en un entorno laboral real [Evaluar]. ▪ Aplicar principios éticos y profesionales en sus actividades [Evaluar]. ▪ Trabajar efectivamente en equipo y comunicarse de manera profesional [Evaluar].
Lecturas : [**ACMInternshipGuide**], [**IEEEProfessionalPractices**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS403. Proyecto de tesis 2 (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS403. Proyecto de tesis 2
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS402. Proyecto de tesis 1. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

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

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

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

7. TEMAS

Unidad 1: Formalización del Plan de Tesis (10 horas)	
Resultados esperados: AG-C05,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructura del plan de tesis según el formato institucional. ▪ Definición precisa del problema de investigación y pregunta central de tesis. ▪ Formulación de objetivos generales y específicos verificables. ▪ Cronograma de actividades con hitos hasta la defensa final de tesis. ▪ Análisis de viabilidad: recursos, tiempos, riesgos y contingencias del proyecto.	▪ Producir el plan de tesis en el formato exigido por la institución académica. [Evaluar] ▪ Definir un problema de investigación original y objetivos verificables. [Evaluar] ▪ Elaborar un cronograma detallado con hitos y análisis de riesgos hasta la defensa de tesis. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 2: Marco Teórico y Estado del Arte Avanzado (10 horas)	
Resultados esperados: AG-C04,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Organización temática y síntesis crítica de la literatura del área de tesis. ▪ Redacción del capítulo de marco teórico en LaTeXcon estructura coherente. ▪ Identificación y justificación de las brechas de investigación del problema. ▪ Análisis comparativo de enfoques: tablas y diagramas de clasificación de la literatura. ▪ Integración de referencias con BibTeX y gestión de versiones con Git.	▪ Redactar el capítulo de marco teórico con síntesis estructurada y crítica de la literatura. [Evaluar] ▪ Justificar la originalidad del problema en relación con el estado del arte. [Evaluar] ▪ Construir tablas comparativas que evidencien las brechas de investigación. [Evaluar]
Lecturas : [**ieee**], [**acm**], [**citeseer**]	

Unidad 3: Diseño de la Metodología Original (10 horas)	
Resultados esperados: AG-C05,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición del enfoque metodológico: experimental, analítico o constructivo. ▪ Especificación de la contribución original: algoritmo, sistema, modelo o metodología. ▪ Diseño del protocolo experimental: variables, métricas, datasets y benchmarks. ▪ Criterios de validación y generalización de resultados de la propuesta. ▪ Alineación de la metodología con los objetivos del plan de tesis.	▪ Diseñar la metodología de investigación alineada con los objetivos. [Evaluar] ▪ Especificar la contribución original del proyecto con criterios verificables. [Evaluar] ▪ Definir el protocolo experimental completo con variables, métricas y benchmarks. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 4: Implementación y Experimentos Preliminares (12 horas)	
Resultados esperados: AG-C05,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Implementación de la propuesta original: prototipo, algoritmo o modelo de tesis. ▪ Reproducción de baselines relevantes del estado del arte para comparación. ▪ Ejecución de experimentos preliminares con análisis estadístico básico. ▪ Registro y documentación de configuraciones, versiones y resultados. ▪ Identificación de limitaciones y ajuste iterativo de la propuesta.	▪ Implementar un prototipo funcional de la propuesta original de tesis. [Usar] ▪ Ejecutar experimentos preliminares comparativos con baselines definidos. [Usar] ▪ Analizar e interpretar resultados con rigor estadístico apropiado. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 5: Redacción de Capítulos Centrales de la Tesis (10 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redacción del capítulo de metodología y diseño experimental en LaTeX. ▪ Redacción del capítulo de trabajos relacionados con análisis crítico comparativo. ▪ Presentación de resultados preliminares: tablas, figuras y análisis comparativo. ▪ Ciclos de revisión con el asesor: incorporación de retroalimentación estructurada. ▪ Coherencia y consistencia narrativa entre capítulos de la tesis.	▪ Redactar los capítulos de metodología y trabajos relacionados con rigor académico. [Evaluar] ▪ Presentar resultados preliminares con visualizaciones claras y análisis comparativo. [Usar] ▪ Aplicar ciclos de revisión con el asesor para mejorar la calidad del documento. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 6: Presentación del Avance y Planificación Final (8 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Presentación oral del avance de tesis ante el comité evaluador. ▪ Defensa de metodología y resultados preliminares ante preguntas del comité. ▪ Incorporación de retroalimentación del comité en la revisión del documento. ▪ Planificación de actividades restantes hacia la defensa final de tesis. ▪ Gestión ética: autoría, integridad académica y confidencialidad de datos.	▪ Presentar y defender oralmente el avance de tesis ante un comité académico. [Evaluar] ▪ Incorporar la retroalimentación del comité en una revisión fundamentada del documento. [Evaluar] ▪ Planificar las actividades restantes con objetivos claros hacia la defensa final. [Usar]
Lecturas : [**ieee**], [**acm**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI365. Modelos Generativos Avanzados en IA (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI365. Modelos Generativos Avanzados en IA
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI264. Aprendizaje Profundo. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- Implementar pipelines de generación de imágenes/texto/video
- Diseñar sistemas de edición asistida por IA
- Evaluar riesgos de contenido sintético

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Fundamentos Generativos (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Teoría de GANs (minimax, Wasserstein) ▪ Autoregressive models (PixelCNN, Transformers) ▪ Métricas de evaluación (FID, Inception Score)	▪ Entrenar GANs básicas [Usar] ▪ Analizar modos de falla en generación [Evaluar]
Lecturas : [**goodfellow2014gan**]	

Unidad 2: Modelos de Difusión (24 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Procesos de difusión (forward/reverse) ▪ Stable Diffusion y arquitecturas latentes ▪ Técnicas de control (ControlNet, LoRA)	▪ Implementar samplers de difusión [Usar] ▪ Fine-tuning para dominios específicos [Evaluar]
Lecturas : [**sohl-dickstein2023diffusion**], [**rombach2022highres**]	

Unidad 3: Ética en Medios Sintéticos (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Deepfakes y desinformación ▪ Derechos de autor en contenido generado ▪ Marcado de autenticidad (C2PA, watermarking)	▪ Detectar contenido sintético [Familiarizarse] ▪ Implementar sistemas de trazabilidad [Usar]
Lecturas : [**sohl-dickstein2023diffusion**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CB309. Bioinformática (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CB309. Bioinformática
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	Ninguno

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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).

5. OBJETIVOS

- 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

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Introducción a la Biología Molecular (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Revisión de la química orgánica: moléculas y macromoléculas, azúcares, acidos nucleicos, nuclótidos, ARN, ADN, proteínas, aminoácidos y nivels de estructura en las proteinas. ▪ El dogma de la vida: del ADN a las proteinas, transcripción, traducción, síntesis de proteinas ▪ Estudio del genoma: Mapas y secuencias, técnicas específicas	▪ Lograr un conocimiento general de los tópicos más importantes en Biología Molecular. [Familiarizarse] ▪ Entender que los problemas biológicos son un desafío al mundo computacional. [Evaluar]
Lecturas : [**Clote2000**], [**Setubal1997**]	

Unidad 2: Comparación de Secuencias (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Secuencias de nucléotidos y secuencias de aminoácidos. ▪ Alineamiento de secuencias, el problema de alineamiento por pares, búsqueda exhaustiva, Programación dinámica, alineamiento global, alineamiento local, penalización por gaps ▪ Comparación de múltiples secuencias: suma de pares, análisis de complejidad por programación dinámica, heurísticas de alineamiento, algoritmo estrella, algoritmos de alineamiento progresivo.	▪ Entender y solucionar el problema de alineamiento de un par de secuencias. [Usar] ▪ Comprender y solucionar el problema de alineamiento de múltiples secuencias. [Usar] ▪ Conocer los diversos algoritmos de alineamiento de secuencias existentes en la literatura. [Familiarizarse]
Lecturas : [**Clote2000**], [**Setubal1997**], [**Pevzner2000**]	

Unidad 3: Árboles Filogenéticos (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Filogenia: Introducción y relaciones filogenéticas. ▪ Arboles Filogenéticos: definición, tipo de árboles, problema de búsqueda y reconstrucción de árboles ▪ Métodos de Reconstrucción: métodos por parsimonia, métodos por distancia, métodos por máxima verosimilitud, confianza de los árboles reconstruidos	▪ Comprender el concepto de filogenia, árboles filogenéticos y la diferencia metodológica entre biología y biología molecular. [Familiarizarse] ▪ Comprender el problema de reconstrucción de árboles filogenéticos, conocer y aplicar los principales algoritmos para reconstrucción de árboles filogenéticos. [Evaluar]
Lecturas : [**Clote2000**], [**Setubal1997**], [**Pevzner2000**]	

Unidad 4: Ensamblaje de Secuencias de ADN (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Fundamento biológico: caso ideal, dificultades, métodos alternativos para secuenciamiento de ADN ▪ Modelos formales de ensamblaje: *Shortest Common Superstring*, *Reconstruction*, *Multicontig* ▪ Algoritmos para ensamblaje de secuencias: representación de overlaps, caminos para crear *superstrings*, algoritmo voraz, grafos acíclicos. ▪ Heurísticas para ensamblaje: búsqueda de sobreposiciones, ordenación de fragmentos, alineamientos y consenso.	▪ Comprender el desafío computacional que ofrece el problema de Ensamblaje de Secuencias. [Familiarizarse] ▪ Entender el principio de modelo formal para ensamblaje. [Evaluar] ▪ Conocer las principales heurísticas para el problema de ensambjale de secuencias ADN [Usar]
Lecturas : [**Setubal1997**], [**Aluru2006**]	

Unidad 5: Estructuras secundarias y terciarias (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estructuras moleculares: primaria, secundaria, terciaria, cuaternaria. ▪ Predicción de estructuras secundarias de ARN: modelo formal, energia de pares, estructuras con bases independientes, solución con Programación Dinámica, estructuras con bucles. ▪ *Protein folding*: Estructuras en proteinas, problema de *protein folding*. ▪ *Protein Threading*: Definiciones, Algoritmo *Branch & Bound*, *Branch & Bound* para *protein threading*. ▪ *Structural Alignment*: definiciones, algoritmo DALI	▪ Conocer las estructuras protéicas y la necesidad de métodos computacionales para la predicción de la geometría. [Familiarizarse] ▪ Conocer los algoritmos de solución de problemas de predicción de estructuras secundarias ARN, y de estructuras en proteínas. [Evaluar]
Lecturas : [**Setubal1997**], [**Clote2000**], [**Aluru2006**]	

Unidad 6: Modelos Probabilísticos en Biología Molecular (4 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Probabilidad: Variables aleatorias, Cadenas de Markov, Algoritmo de Metropoli-Hasting, Campos Aleatorios de Markov y Muestreador de Gibbs, Máxima Verosimilitud. ▪ 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. ▪ Filogenia Probabilística: Modelos probabilísticos de evolución, verosimilitud de alineamientos, verosimilitud para inferencia, comparación de métodos probailísticos y no probabilísticos	▪ Revisar conceptos de Modelos Probabilísticos y comprender su importancia en Biología Molecular Computacional. [Evaluar] ▪ Conocer y aplicar Modelos Ocultos de Markov para varios análisis en Biología Molecular. [Usar] ▪ Conocer la aplicación de modelos probabilísticos en Filogenia y comparalos con modelos no probabilísticos [Evaluar]
Lecturas : [**Durbin1998**], [**Clote2000**], [**Aluru2006**], [**Krogh1994**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

ET301. Formación de Empresas de Base Tecnológica II (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	ET301. Formación de Empresas de Base Tecnológica II
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	ET201. Formación de Empresas de Base Tecnológica I. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

Este curso tiene como objetivo dotar al futuro profesional de conocimientos, actitudes y aptitudes que le permitan formar su propia empresa de desarrollo de software y/o consultoría en informática. El curso está dividido en tres unidades: Valorización de Proyectos, Marketing de Servicios y Negociaciones. En la primera unidad se busca que el alumno pueda analizar y tomar decisiones en relación a la viabilidad de un proyecto y/o negocio.

En la segunda unidad se busca preparar al alumno para que este pueda llevar a cabo un plan de marketing satisfactorio del bien o servicio que su empresa pueda ofrecer al mercado. La tercera unidad busca desarrollar la capacidad negociadora de los participantes a través del entrenamiento vivencial y práctico y de los conocimientos teóricos que le permitan cerrar contrataciones donde tanto el cliente como el proveedor resulten ganadores. Consideramos estos temas sumamente críticos en las etapas de lanzamiento, consolidación y eventual relanzamiento de una empresa de base tecnológica.

5. OBJETIVOS

- Que el alumno comprenda y aplique la terminología y conceptos fundamentales de ingeniería económica que le permitan valorizar un proyecto para tomar la mejor decisión económica.
- Que el alumno adquiera las bases para formar su propia empresa de base tecnológica.

6. RESULTADOS DEL ESTUDIANTE

Sin elementos para esta tabla

7. TEMAS

Unidad 1: (20 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción ▪ Proceso de toma de decisiones ▪ El valor del dinero en el tiempo ▪ Tasa de interés y tasa de rendimiento ▪ Interés simple e interés compuesto ▪ Identificación de costos ▪ Flujo de Caja Neto ▪ Tasa de Retorno de Inversión (TIR) ▪ Valor Presente Neto (VPN) ▪ Valorización de Proyectos	▪ Permitir al alumno tomar decisiones sobre como invertir mejor los fondos disponibles, fundamentadas en el análisis de los factores tanto económicos como no económicos que determinen la viabilidad de un emprendimiento. [Evaluar]
Lecturas : [**blank06**]	

Unidad 2: (30 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción ▪ Importancia del marketing en las empresas de servicios ▪ El Proceso estratégico. ▪ El Plan de Marketing ▪ Marketing estratégico y marketing operativo ▪ Segmentación, targeting y posicionamiento de servicios en mercados competitivos ▪ Ciclo de vida del producto ▪ Aspectos a considerar en la fijación de precios en servicios ▪ El rol de la publicidad, las ventas y otras formas de comunicación ▪ El comportamiento del consumidor en servicios ▪ Fundamentos de marketing de servicios ▪ Creación del modelo de servicio ▪ Gestión de la calidad de servicio	▪ Brindar las herramientas al alumno para que pueda identificar, analizar y aprovechar las oportunidades de marketing que generan valor en un emprendimiento. [Usar] ▪ Lograr que el alumno conozca, entienda e identifique criterios, habilidades, métodos y procedimientos que permitan una adecuada formulación de estrategias de marketing en sectores y medios específicos como lo es una empresa de base tecnológica. [Usar]
Lecturas : [**kotler06**], [**love09**]	

Unidad 3: (10 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción. ¿Qué es una negociación? ▪ Teoría de las necesidades de la negociación ▪ La proceso de la negociación ▪ Estilos de negociación ▪ Teoría de juegos ▪ El método Harvard de negociación	▪ Conocer los puntos clave en el proceso de negociación. [Usar] ▪ Establecer una metodología de negociación eficaz. [Usar] ▪ Desarrollar destrezas y habilidades que permitan llevar a cabo una negociación exitosa. [Usar]
Lecturas : [**fish96**], [**dasi06**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI369. Robótica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI369. Robótica
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI268. Visión Computacional. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Robótica (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general: problemas y progreso: • Sistemas robóticos de última generación, incluyendo sus sensores y una visión general de su procesamiento sensorial • Arquitecturas de control de robots, por ejemplo, control deliberativo versus reactivo y vehículos de Braitenberg • Modelado del mundo y modelos del mundo • Incertidumbre inherente en la percepción y en el control ▪ Sensores y efectores: • Sensores: por ejemplo, LIDAR, sonar, visión, profundidad, estereoscópico, cámaras de eventos, micrófonos, háptica • Efectores: por ejemplo, ruedas, brazos, pinzas ▪ Marcos de coordenadas, traslación y rotación (2D y 3D) ▪ Espacio de configuración y mapas ambientales ▪ Interpretación de datos sensoriales inciertos ▪ Localización y mapeo ▪ Navegación y control ▪ Cinemática directa e inversa ▪ Planificación de trayectorias de movimiento y optimización de trayectorias ▪ Manipulación y agarre ▪ Control y dinámica de articulaciones ▪ Control basado en visión ▪ Coordinación y colaboración múltiple de robots ▪ Interacción humano-robot: • Espacios de trabajo compartidos • Equipos humano-robot e HRI físico • Robots de asistencia social • Predicción de movimiento/tarea/meta • Colaboración y comunicación (explícita vs implícita, verbal o simbólica vs no verbal o visual) • Confianza ▪ Aplicaciones y Problemas Sociales, Económicos y éticos: • Implicaciones sociales, económicas, de derecho al trabajo • Implicaciones éticas y de privacidad de las aplicaciones robóticas • Responsabilidad en robótica autónoma	▪ 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] ▪ Integrar sensores, actuadores y software en un robot diseñado para emprender una tarea específica [Aplicar] ▪ Programar un robot para realizar tareas simples usando arquitecturas de control deliberativas, reactivas y/o híbridas [Aplicar] ▪ Implementar algoritmos fundamentales de planificación de movimiento dentro de un espacio de configuración de robot [Aplicar] ▪ Caracterizar las incertidumbres asociadas con sensores y actuadores robóticos comunes; articular estrategias para mitigar estas incertidumbres [Caracterizar] ▪ Listar las diferencias entre las representaciones del entorno externo de los robots, incluyendo sus fortalezas y limitaciones [Listar/Enumerar] ▪ 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] ▪ Describir al menos un enfoque para coordinar las acciones y percepción de varios robots para realizar una única tarea [Describir] ▪ 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] ▪ Analizar los problemas sociales, económicos y éticos de una aplicación robótica del mundo real [Analizar]

Unidad 2: Percepción y Visión por Computadora (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión por computadora: • Adquisición, representación, procesamiento y propiedades de imágenes • Representación de formas, reconocimiento de objetos y segmentación • Análisis de movimiento • Modelos generativos ▪ Audio y reconocimiento de voz ▪ Tacto y propiocepción ▪ Otras modalidades (por ejemplo, olfato) ▪ Modularidad en el reconocimiento ▪ Enfoques para el reconocimiento de patrones: • Algoritmos de clasificación y medidas de calidad de clasificación • Técnicas estadísticas • Técnicas de aprendizaje profundo	▪ Resumir la importancia del reconocimiento de imágenes y objetos en IA e indicar varias aplicaciones significativas de esta tecnología [Resumir] ▪ 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] ▪ Implementar reconocimiento de objetos 2D basado en representaciones de forma basadas en contorno y/o región [Aplicar] ▪ 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] ▪ 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] ▪ 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] ▪ 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] ▪ Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Aplicar] ▪ 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] ▪ Describir al menos tres enfoques de clasificación, sus prerrequisitos para aplicabilidad, sus fortalezas y sus limitaciones [Describir] ▪ Implementar y evaluar una solución de aprendizaje profundo para problemas en visión por computadora, como reconocimiento de objetos o escenas [Aplicar]
Lecturas : [**Thrun2005**], [**Siegwart2011**]	

Unidad 3: Robótica (24 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Visión general: problemas y progreso: • Sistemas robóticos de última generación, incluyendo sus sensores y una visión general de su procesamiento sensorial • Arquitecturas de control de robots, por ejemplo, control deliberativo versus reactivo y vehículos de Braitenberg • Modelado del mundo y modelos del mundo • Incertidumbre inherente en la percepción y en el control ▪ Sensores y efectores: • Sensores: por ejemplo, LIDAR, sonar, visión, profundidad, estereoscópico, cámaras de eventos, micrófonos, háptica • Efectores: por ejemplo, ruedas, brazos, pinzas ▪ Marcos de coordenadas, traslación y rotación (2D y 3D) ▪ Espacio de configuración y mapas ambientales ▪ Interpretación de datos sensoriales inciertos ▪ Localización y mapeo ▪ Navegación y control ▪ Cinemática directa e inversa ▪ Planificación de trayectorias de movimiento y optimización de trayectorias ▪ Manipulación y agarre ▪ Control y dinámica de articulaciones ▪ Control basado en visión ▪ Coordinación y colaboración múltiple de robots ▪ Interacción humano-robot: • Espacios de trabajo compartidos • Equipos humano-robot e HRI físico • Robots de asistencia social • Predicción de movimiento/tarea/meta • Colaboración y comunicación (explícita vs implícita, verbal o simbólica vs no verbal o visual) • Confianza ▪ Aplicaciones y Problemas Sociales, Económicos y éticos: • Implicaciones sociales, económicas, de derecho al trabajo • Implicaciones éticas y de privacidad de las aplicaciones robóticas • Responsabilidad en robótica autónoma	▪ 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] ▪ Integrar sensores, actuadores y software en un robot diseñado para emprender una tarea específica [Aplicar] ▪ Programar un robot para realizar tareas simples usando arquitecturas de control deliberativas, reactivas y/o híbridas [Aplicar] ▪ Implementar algoritmos fundamentales de planificación de movimiento dentro de un espacio de configuración de robot [Aplicar] ▪ Caracterizar las incertidumbres asociadas con sensores y actuadores robóticos comunes; articular estrategias para mitigar estas incertidumbres [Caracterizar] ▪ Listar las diferencias entre las representaciones del entorno externo de los robots, incluyendo sus fortalezas y limitaciones [Listar/Enumerar] ▪ 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] ▪ Describir al menos un enfoque para coordinar las acciones y percepción de varios robots para realizar una única tarea [Describir] ▪ 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] ▪ Analizar los problemas sociales, económicos y éticos de una aplicación robótica del mundo real [Analizar]

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS351. Tópicos en Computación Gráfica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS351. Tópicos en Computación Gráfica
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS251. Computación Gráfica. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Sombreado y Renderizado Avanzado (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Sombreado avanzado: sombreado basado en física, dispersión subsuperficial (subsurface scattering) ▪ Iluminación global (global illumination): radiosidad, mapeo de fotones (photon mapping), trazado de caminos (path tracing) ▪ Técnicas de renderizado en tiempo real: nivel de detalle (level of detail), culling, oclusión ▪ Programación GPU y sombreadores (shaders)	▪ Implementar modelos de sombreado avanzados para materiales realistas [Implementar] ▪ Comparar algoritmos de iluminación global y sus compensaciones [Comparar] ▪ Optimizar el renderizado en tiempo real usando técnicas como nivel de detalle y *culling* [Crear] ▪ Desarrollar sombreadores GPU para efectos de renderizado personalizados [Crear]
Lecturas : [**Marschner2021**], [**Shirley2021**]	

Unidad 2: Modelado Geométrico (10 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Representaciones geométricas: polígonos, curvas, superficies, formas implícitas ▪ Técnicas de modelado: curvas spline, superficies de subdivisión, geometría sólida constructiva ▪ Modelado procedural y generación ▪ Algoritmos geométricos: intersección, distancia, descomposición	▪ Describir diferentes representaciones geométricas y sus compensaciones (trade-offs) [Describir] ▪ Implementar técnicas de modelado como curvas spline o superficies de subdivisión [Implementar] ▪ Crear modelos procedurales usando algoritmos o reglas [Crear] ▪ Aplicar algoritmos geométricos para resolver problemas como pruebas de intersección o procesamiento de mallas (mesh) [Aplicar]
Lecturas : [**Marschner2021**]	

Unidad 3: Animación por Computadora (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de animación: tiempo (timing), espaciado (spacing), anticipación, seguimiento (follow-through) ▪ Animación por fotogramas clave (keyframe) e interpolación ▪ Animación procedural: sistemas de partículas, comportamiento de bandadas (flocking), basada en física ▪ Animación de personajes: animación esquelética, skinning, cinemática inversa ▪ Captura de movimiento (motion capture) y retargeting	▪ Aplicar principios de animación para crear movimiento creíble [Aplicar] ▪ Implementar animación por fotogramas clave con técnicas de interpolación [Implementar] ▪ Crear animaciones procedurales como sistemas de partículas o simulaciones basadas en física [Crear] ▪ Implementar animación de personajes usando sistemas esqueléticos y skinning [Implementar] ▪ Procesar datos de captura de movimiento para retargeting de animación [Implementar]
Lecturas : [**Parent2012**]	

Unidad 4: Simulación (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Simulación basada en física: cuerpos rígidos, fluidos, tela ▪ Métodos numéricos para simulación: integración, detección de colisiones ▪ Restricciones y optimizaciones de simulación en tiempo real ▪ Aplicaciones en videojuegos, efectos visuales y visualización científica	▪ Implementar simulaciones basadas en física para cuerpos rígidos o fluidos [Implementar] ▪ Aplicar métodos de integración numérica para simulaciones dinámicas [Aplicar] ▪ Optimizar simulaciones para rendimiento en tiempo real [Crear] ▪ Desarrollar simulaciones para aplicaciones específicas como videojuegos o visualización científica [Crear]
Lecturas : [**Parent2012**]	

Unidad 5: Inmersión (5 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conceptos de realidad virtual (VR) y realidad aumentada (AR) ▪ Tecnologías de visualización inmersiva: pantallas montadas en la cabeza (head-mounted displays), CAVEs ▪ Seguimiento (tracking) y entrada para entornos inmersivos ▪ Presencia y comodidad en sistemas inmersivos	▪ Explicar los conceptos centrales y diferencias entre VR y AR [Explicar] ▪ Evaluar tecnologías de visualización inmersiva y sus aplicaciones [Evaluar] ▪ Implementar soluciones de seguimiento y entrada para entornos inmersivos [Implementar] ▪ Diseñar experiencias inmersivas que maximicen la presencia y minimicen la incomodidad [Diseñar]
Lecturas : [**LaValle2023**]	

Unidad 6: Interacción (5 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas de interacción: manipulación directa, basada en gestos, tangible ▪ Diseño de interfaz de usuario para aplicaciones gráficas ▪ Interacción multimodal: combinando retroalimentación visual, auditiva, háptica ▪ Evaluación de técnicas de interacción: usabilidad, experiencia de usuario	▪ Comparar diferentes paradigmas de interacción y su idoneidad para varias tareas [Comparar] ▪ Diseñar interfaces de usuario para aplicaciones intensivas en gráficos [Diseñar] ▪ Implementar sistemas de interacción multimodal incorporando múltiples canales de retroalimentación [Implementar] ▪ Realizar evaluaciones de usabilidad de técnicas de interacción [Evaluar]
Lecturas : [**LaValle2023**]	

Unidad 7: Computación Tangible/Física (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Interfaces tangibles y computación física ▪ Sensores y actuadores para sistemas interactivos ▪ Sistemas embebidos para arte y diseño interactivo ▪ Instalaciones y entornos interactivos	▪ Diseñar interfaces tangibles que unan interacciones físicas y digitales [Diseñar] ▪ Integrar sensores y actuadores para crear sistemas responsivos [Integrar] ▪ Desarrollar sistemas embebidos para proyectos de arte o diseño interactivo [Crear] ▪ Crear instalaciones interactivas que involucren a los usuarios en espacios físicos [Crear]
Lecturas : [**Noble2012**]	

Unidad 8: Sociedad, ética y Profesión (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Consideraciones éticas en gráficos e interacción: representación, accesibilidad, privacidad ▪ Impacto social de las tecnologías inmersivas y la visualización ▪ Práctica profesional en industrias de gráficos e interactivas	▪ Analizar problemas éticos relacionados con representación, accesibilidad y privacidad en gráficos e interacción [Analizar] ▪ Discutir las implicaciones sociales de las tecnologías inmersivas y la visualización de datos [Debatir] ▪ Describir prácticas y estándares profesionales en industrias de gráficos e interactivas [Describir]
Lecturas : [**Marschner2021**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS353. Computación Cuántica (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS353. Computación Cuántica
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS3I1. Seguridad en Computación. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Conceptos fundamentales (10 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Cúbits y representación en la esfera de Bloch. ▪ Postulados de la mecánica cuántica. ▪ Compuertas cuánticas (Hadamard, CNOT, Toffoli).	▪ Modelar matemáticamente un cúbit [Usar]. ▪ Simular operaciones básicas en Qiskit [Evaluar].
Lecturas : [**NielsenChuang**], [**QiskitTextbook**]	

Unidad 2: Arquitecturas Cuánticas (10 horas)

Resultados esperados: AG-C10,AG-C11

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios: • El principio de dualidad onda-partícula • El principio de incertidumbre en el experimento de doble rendija • ¿Qué es un Qubit? Superposición, interferencia y medición. Fotones como qubits • Sistemas de dos qubits, Entrelazamiento, estados de Bell, El teorema de No Señalización ▪ Axiomas de la MC: principio de superposición, axioma de medición, evolución unitaria ▪ Compuertas de un solo qubit para el modelo de circuito de computación cuántica: X, Z, H ▪ Compuertas de dos qubits y productos tensoriales, trabajando con matrices ▪ El Teorema de No Clonación. El protocolo de Teleportación Cuántica ▪ Algoritmos: • Algoritmos cuánticos simples: Bernstein-Vazirani, algoritmo de Simon • Implementar Deutsch-Josza con Interferómetros Mach-Zehnder • Factorización cuántica (Algoritmo de Shor) • Búsqueda cuántica (Algoritmo de Grover) ▪ Aspectos de implementación: • La implementación física de qubits • Control clásico de una Unidad de Procesamiento Cuántico (QPU) • Mitigación y control de errores, NISQ y más allá • Enfoques de medición ▪ Aplicaciones Emergentes: • Cifrado post-cuántico • La Internet Cuántica • Computación cuántica adiabática (AQC) y recocido cuántico	▪ 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] ▪ Discutir la naturaleza a nivel cuántico que es inherentemente probabilística [Debatir] ▪ 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] ▪ Describir el paralelismo cuántico y el papel de la interferencia constructiva vs destructiva en algoritmos cuánticos dada la naturaleza probabilística de las mediciones [Describir] ▪ Analizar un fragmento de código proporcionando el papel de la transformada cuántica de Fourier (QFT) en el algoritmo de Shor [Analizar] ▪ Escribir un programa para implementar el algoritmo de Shor en un simulador, destacando los componentes clásicos y aspectos del algoritmo de Shor [Escribir] ▪ 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] ▪ Contrastar AQC con el modelo de compuertas de computación cuántica y los problemas para los que cada uno es más adecuado resolver [Contrastar]

Lecturas : [**QuantumHardwareBook**], [**IBMQuantumRoadmap2023**]

Unidad 3: Estrategias Algorítmicas (12 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Computación cuántica ▪ Algoritmo de Deutsch-Jozsa. ▪ Búsqueda de Grover (aplicada a optimización combinatoria). ▪ Factorización de Shor y su impacto en RSA.	▪ Implementar Grover para problemas SAT [Usar].
Lecturas : [**PreskillNotes**], [**QiskitTextbook**]	

Unidad 4: Lenguajes Formales y Autómatas (10 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Computación cuántica: • Postulados de la mecánica cuántica: ◦ Espacio de estados ◦ Evolución del estado ◦ Composición de estados ◦ Medición del estado • Representaciones de vector columna de qubits • Representaciones matriciales de operaciones cuánticas • Compuertas cuánticas simples (por ejemplo, XNOT, CNOT) ▪ Algoritmo de Optimización Cuántica Aproximada (QAOA). ▪ Aplicaciones en logística/finanzas (Optimización de Cartera).	▪ Para un sistema cuántico dar ejemplos que expliquen los siguientes postulados: • Espacio de Estados - estado del sistema representado como un vector unitario en espacio de Hilbert • Evolución del Estado - el uso de operadores unitarios para evolucionar el estado del sistema • Composición de Estados - el uso de producto tensorial para componer estados del sistema • Medición del Estado - la salida probabilística de medir un estado del sistema. [Familiarizarse]. ▪ Diseñar un QAOA para problemas de enrutamiento [Evaluar].
Lecturas : [**QMLBook**], [**Bravyi2022**]	

Unidad 5: Ethics and Post-Quantum Security (6 horas)	
Resultados esperados: AG-C10,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Criptografía post-cuántica (Estandarización NIST PQC). ▪ Supremacía cuántica: debate técnico/ético. ▪ Patentes y geopolítica cuántica.	▪ Proponer estrategias de migración post-cuántica [Familiarizarse].
Lecturas : [**QuantumSupremacyPaper**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS392. Tópicos en Ingeniería de Software (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS392. Tópicos en Ingeniería de Software
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	2 HT; 2 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS391. Ingeniería de Software III. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Colaboración y Dinámica de Equipo (4 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Comunicación efectiva, incluida la oral y escrita, así como formal (correo electrónico, documentos, comentarios, presentaciones) e informal (chat de equipo, reuniones). Comunicación Técnica y Profesional, Comunicación en Equipo y Cultura ▪ Causas comunes de conflicto en el equipo y enfoques para la resolución de conflictos. ▪ Programación cooperativa: • Programación por parejas (pair programming) o Enjambre (swarming) • Revisión de código • Colaboración mediante control de versiones ▪ Roles y responsabilidades en un equipo de software: Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo • Ventajas del trabajo en equipo • Riesgos y complejidad de dicha colaboración ▪ Procesos del equipo: responsabilidades para las tareas, estimación de esfuerzo, estructura de reuniones, cronograma de trabajo	▪ Seguir prácticas efectivas de comunicación en equipo [Analizar] ▪ 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] ▪ Facilitar una estrategia de resolución de conflictos y resolución de problemas en un entorno de equipo [Analizar] ▪ Colaborar efectivamente en desarrollo/programación cooperativa [Analizar] ▪ Proponer y delegar los roles y responsabilidades necesarios en un equipo de desarrollo de software [Proponer] ▪ Redactar y seguir una agenda para una reunión de equipo [Componer] ▪ 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]
Lecturas : [**Sommerville2015**]	

Unidad 2: Control de Versiones y CI/CD (4 horas)

Resultados esperados: AG-C03,AG-C09

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gestión de configuración de software y control de versiones: Prácticas de Desarrollo de Software • Configuración en control de versiones, builds/configuración reproducibles. • Estrategias de ramificación en control de versiones. Ramas de desarrollo vs ramas de lanzamiento. Desarrollo basado en tronco (*trunk-based development*). • Estrategias de fusión/rebase, cuando sean relevantes. ▪ Gestión de lanzamientos (*release management*). ▪ Herramientas de prueba, incluyendo herramientas de análisis estático y dinámico. Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada ▪ Automatización de procesos de software: • Sistemas de construcción (*build systems*): el valor de builds rápidos, herméticos y reproducibles, comparar/contrastar enfoques para construir un proyecto. • 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). • 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. • Gestión de dependencias: actualización de dependencias externas/aguas arriba, gestión de paquetes, SemVer. ▪ Flujo de revisión colaborativa como puerta de calidad • Revisión obligatoria de *pull requests* antes de fusionar • Propiedad del código mediante reglas de tipo CODEOWNERS • Convenciones de mensajes de *commit* (*conventional commits*) • Reglas de protección de ramas que exigen verificaciones antes de fusionar ▪ Escaneo de vulnerabilidades de dependencias y de la cadena de suministro integrado en los pipelines de CI	▪ Describir la diferencia entre la gestión de configuración de software centralizada y distribuida [Describir] ▪ Describir cómo el control de versiones puede usarse para ayudar en la gestión de lanzamientos de software [Describir] ▪ Identificar elementos de configuración y usar una herramienta de control de código fuente en un proyecto pequeño basado en equipo [Analizar] ▪ Describir cómo las herramientas de prueba estáticas y dinámicas disponibles pueden integrarse en el entorno de desarrollo de software [Describir] ▪ 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] ▪ 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] ▪ Usar un escáner de vulnerabilidades de dependencias/cadena de suministro como parte de un pipeline de CI para detectar dependencias vulnerables [Usar]

Lecturas : [**Pressman2014**]

Unidad 3: Ingeniería de Requisitos (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Describir requisitos funcionales usando, por ejemplo, casos de uso o historias de usuario. • Usar al menos un método para documentar y estructurar requisitos funcionales. • Comprender cómo el método apoya el diseño y la implementación. • Fortalezas y debilidades de usar un enfoque específico. ▪ Propiedades de los requisitos, incluyendo consistencia, validez, completitud y viabilidad. ▪ Obtención (*elicitation*) de requisitos. • Fuentes de requisitos, por ejemplo, usuarios, administradores o personal de soporte. • Métodos de recopilación de requisitos, por ejemplo, encuestas, entrevistas o análisis de comportamiento. ▪ Requisitos no funcionales, por ejemplo, seguridad, usabilidad o rendimiento, también llamados Atributos de Calidad. Diseño Sostenible y Computación Ubicua, Huella Ambiental de los Sistemas Informáticos, Efectos Sistémicos y Contexto Social ▪ Identificación y gestión de riesgos, incluyendo consideraciones éticas en torno al producto propuesto. Códigos de Ética y Responsabilidad Legal, Responsabilidad Profesional y Política Pública, Valores, Naturaleza y Actualización Profesional, Disidencia Ética y Contexto Cultural, Ética en el Lugar de Trabajo ▪ Comunicar y/o formalizar especificaciones de requisitos.	▪ Comparar diferentes métodos de obtención de requisitos a lo largo de múltiples ejes [Comparar] ▪ 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] ▪ Identificar qué comportamientos son requeridos, permitidos o prohibidos a partir de un conjunto dado de requisitos y una lista de comportamientos candidatos [Analizar] ▪ Recopilar un conjunto de requisitos para un sistema de software simple [Analizar] ▪ Identificar áreas de un sistema de software que deben cambiarse, dada una descripción del sistema y un conjunto de nuevos requisitos a implementar [Analizar] ▪ Identificar los requisitos funcionales y no funcionales en un conjunto de requisitos [Analizar]
Lecturas : [**Sommerville2015**]	

Unidad 4: Principios y Arquitectura de Software (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de diseño de sistemas. Confiabilidad del Sistema • Niveles de abstracción (por ejemplo, diseño arquitectónico y diseño detallado) • Separación de preocupaciones (separation of concerns) • Ocultación de información (information hiding) • Acoplamiento y cohesión (coupling and cohesion) ▪ Arquitectura de software. Confiabilidad del Sistema • Paradigmas de diseño ◦ Descomposición funcional de arriba hacia abajo/diseño por capas ◦ Arquitectura orientada a datos ◦ Análisis y diseño orientado a objetos ◦ Diseño dirigido por eventos • Arquitecturas estándar (por ejemplo, cliente-servidor y arquitecturas de microservicios incluyendo discusiones sobre REST, n-capas, tuberías y filtros, Modelo Vista Controlador) • Identificar límites y dependencias de componentes ▪ Programación a gran escala vs programación a pequeña escala (programming in the large vs programming in the small). Confiabilidad del Sistema ▪ Malos olores del código (code smells) y otras indicaciones de la calidad del código, distintas de la corrección. Diseño e Ingeniería de Controles de Seguridad, Análisis de Amenazas e Ingeniería de Seguridad, Computación Confiable e Ingeniería de Privacidad	▪ Identificar la arquitectura de software estándar de un diseño de alto nivel dado [Analizar] ▪ 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] ▪ Adaptar un diseño de sistema defectuoso para que siga mejor principios como la separación de preocupaciones o la ocultación de información [Crear] ▪ Identificar las dependencias entre un conjunto de componentes de software en un diseño arquitectónico [Analizar]
Lecturas : [**Sommerville2015**]	

Unidad 5: Prácticas de Codificación (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas prácticas a pequeña escala Prácticas de Desarrollo de Software • Pruebas unitarias • Desarrollo guiado por pruebas (Test-driven development): esto es particularmente valioso para los estudiantes psicológicamente, ya que es mucho más fácil involucrarse constructivamente con el desafío de identificar entradas desafiantes para una API dada (casos límite, casos extremos) a priori. Si implementan primero, el instinto es a menudo evitar intentar hacer fallar su nueva creación, mientras que un enfoque de pruebas primero les da la satisfacción intelectual de detectar los casos problemáticos y luego ver cómo pasan más pruebas durante el proceso de desarrollo. ▪ Documentación Prácticas de Desarrollo de Software • Documentación de la interfaz: describir los requisitos de la interfaz, potencialmente incluyendo contratos (formales o informales), pre y post condiciones, invariantes. • 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.) ◦ Aclarar el flujo de datos, cómputo, etc., centrándose en lo que el código es. ◦ Identificar partes sutiles/complicadas del código y refactorizar para que sean autoexplicativas si es posible o proporcionar comentarios apropiados para aclarar. ▪ Estilo de codificación Prácticas de Desarrollo de Software • Guías de estilo • Comentarios • Nomenclatura (*naming*) ▪ "Mejores Prácticas" para la codificación: técnicas, patrones/idiomas, mecanismos para construir programas de calidad Prácticas de Desarrollo de Software, Flujo de Información y No Interferencia, Inyección y Validación de Entrada, Seguridad de Memoria y Tipos, Análisis de Malware y Seguridad Avanzada • Prácticas de codificación defensiva • Prácticas y principios de codificación segura • Usar mecanismos de manejo de excepciones para hacer los programas más robustos, tolerantes a fallos	▪ Escribir pruebas unitarias apropiadas para un componente pequeño (varias funciones, un solo tipo, etc.) [Escribir] ▪ Escribir comentarios apropiados de interfaz y (si es necesario) de implementación para un componente pequeño [Escribir] ▪ Describir técnicas, patrones de codificación y mecanismos para implementar diseños para lograr propiedades deseadas como confiabilidad, eficiencia y robustez [Describir] ▪ Escribir código robusto utilizando mecanismos de manejo de excepciones [Escribir] ▪ Describir prácticas de codificación segura y defensiva [Describir] ▪ Seleccionar y usar un estándar de codificación definido en un proyecto de software pequeño [Usar] ▪ Comparar y contrastar estrategias de integración, incluyendo integración de arriba hacia abajo (top-down), de abajo hacia arriba (bottom-up) y sándwich [Comparar] ▪ Describir el proceso de analizar e implementar cambios en la base de código desarrollada para un proyecto específico [Describir] ▪ Describir el proceso de analizar e implementar cambios en una base de código existente grande [Describir]

Unidad 6: Compatibilidad y Versionado de API (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ley de Hyrum / Ley de las Interfaces Implícitas ▪ Compatibilidad con versiones anteriores • La compatibilidad no es una propiedad de una sola entidad, es una propiedad de una relación. • La compatibilidad con versiones anteriores debe evaluarse en términos de proveedor + consumidor(es) o con un modelo bien especificado de qué formas de compatibilidad aspira/promete un proveedor. ▪ Control de versiones (versioning) • Versionado Semántico (SemVer) • Desarrollo basado en tronco (*trunk-based development*)	▪ Identificar tanto el comportamiento explícito como el implícito de una interfaz e identificar riesgos potenciales de la Ley de Hyrum [Analizar] ▪ Identificar cambios que pueden considerarse ampliamente "compatibles con versiones anteriores", potencialmente con declaraciones explícitas sobre qué uso está o no soportado [Analizar] ▪ Evaluar si un cambio propuesto es lo suficientemente seguro dada la metodología de control de versiones en uso para un proyecto dado [Evaluar]
Lecturas : [**Fowler2018**]	

Unidad 7: Conceptos de verificación y validación (6 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Conceptos de verificación y validación • Verificación: ¿Estamos construyendo la cosa correctamente? • Validación: ¿Construimos la cosa correcta? ▪ Por qué importan las pruebas: ¿El componente sigue siendo funcional a medida que el código evoluciona? ▪ Objetivos de las pruebas • Usabilidad • Confiabilidad • Conformidad con la especificación • Rendimiento • Seguridad	▪ Explicar por qué las pruebas son importantes [Explicar] ▪ Distinguir entre validación y verificación de programas [Distinguir] ▪ Describir diferentes objetivos de las pruebas [Describir]
Lecturas : [**Pressman2014**]	

Unidad 8: Herramientas y Análisis de Pruebas (3 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Herramientas y automatización de verificación y validación • Análisis estático • Cobertura de código • Fuzzing • Análisis dinámico y contención de fallos (sanitizadores, etc.) • Registro y seguimiento de fallos ▪ Planificación y generación de pruebas • Estimación de fallos y terminación de pruebas, incluyendo siembra de defectos • Uso de números aleatorios y pseudoaleatorios en las pruebas ▪ Pruebas de sistemas asíncronos, paralelos y concurrentes ▪ Verificación y validación de artefactos que no son código (documentación, materiales de capacitación) ▪ Desarrollo guiado por comportamiento (BDD) • Expresar criterios de aceptación como escenarios Gherkin Given/When/Then • Ejecutar escenarios BDD como pruebas automatizadas de extremo a extremo ▪ Pruebas de mutación y análisis estático de calidad de código • Pruebas de mutación para evaluar la efectividad de una suite de pruebas • Detección estática de *code smells* y complejidad ciclomática excesiva	▪ Describir y comparar diferentes herramientas para verificación y validación [Describir] ▪ Automatizar el uso de diferentes herramientas en un proyecto de software pequeño [Diseñar] ▪ Explicar cómo y cuándo se deben usar números aleatorios en las pruebas [Explicar] ▪ Describir enfoques para la estimación de fallos [Describir] ▪ Estimar el número de fallos en una aplicación de software pequeña basándose en la densidad de fallos y la siembra de fallos [Estimar] ▪ Describir técnicas y problemas con las pruebas de software asíncrono, concurrente y paralelo [Describir] ▪ 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] ▪ Describir técnicas para la verificación y validación de artefactos que no son código [Describir] ▪ Escribir escenarios Gherkin Given/When/Then para los criterios de aceptación de una funcionalidad y ejecutarlos como pruebas automatizadas de extremo a extremo [Escribir] ▪ Analizar el puntaje de mutación de una suite de pruebas y refactorizar las pruebas para eliminar los mutantes sobrevivientes [Analizar]
Lecturas : [**Pressman2014**]	

Unidad 9: Pruebas de Rendimiento y Evaluación Comparativa (3 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Pruebas de rendimiento y evaluación comparativa • Rendimiento (*throughput*) y latencia • Degradación bajo carga (pruebas de estrés, manejo FIFO vs LIFO de solicitudes) • Aceleración (*speedup*) y escalado (*scaling*) ◦ Ley de Amdahl ◦ Ley de Gustafson ◦ Escalado suave (*soft*) y débil (*weak*) • Identificar y medir figuras de mérito (*figures of merit*) • Cuellos de botella de rendimiento comunes ◦ Limitado por cómputo (*compute-bound*) ◦ Limitado por ancho de banda de memoria ◦ Limitado por latencia • Métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) ◦ Estimación de la incertidumbre ◦ Intervalos de confianza • Análisis y presentación (gráficos, etc.) • Técnicas de medición de tiempo (*timing*)	▪ Describir el rendimiento (*throughput*) y la latencia y proporcionar ejemplos de cada uno [Describir] ▪ Explicar la aceleración (speedup) y las diferentes formas de escalado (scaling) y cómo se calculan [Explicar] ▪ Describir cuellos de botella de rendimiento comunes [Describir] ▪ Describir métodos estadísticos y mejores prácticas para la evaluación comparativa (benchmarking) de software [Describir] ▪ Explicar técnicas y desafíos para medir el tiempo al construir una evaluación comparativa (benchmark) [Explicar] ▪ 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]
Lecturas : [**Bondi2015**], [**Gregg2020**]	

Unidad 10: Conceptos de Confiabilidad (5 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Concepto de confiabilidad como probabilidad de fallo o tiempo medio entre fallos, y fallas (faults) como causa de fallos (failures) ▪ Identificar requisitos de confiabilidad para diferentes tipos de software ▪ Fallos de software causados por defectos/errores (bugs), por lo que para alta confiabilidad el objetivo es tener un mínimo de defectos - inyectando menos defectos (mejor capacitación, educación, planificación) y eliminando la mayoría de los defectos inyectados (pruebas, revisión de código, etc.) ▪ Confiabilidad del software, confiabilidad del sistema y comportamiento de fallos ▪ Ciclo de inyección y eliminación de defectos, y diferentes enfoques para la eliminación de defectos ▪ Comparar el enfoque de "presupuesto de error" (error budget) para la confiabilidad con el enfoque "libre de errores" (error-free) e identificar dominios donde cada uno es relevante.	▪ Describir cómo determinar el nivel de confiabilidad requerido por un sistema de software [Describir] ▪ Explicar los problemas que existen para lograr niveles muy altos de confiabilidad [Explicar] ▪ Comprender enfoques para minimizar fallas que pueden aplicarse en cada etapa del ciclo de vida del software [Explicar]
Lecturas : [**Pressman2014**]	

Unidad 11: Métodos Formales (5 horas)	
Resultados esperados: AG-C03,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Especificación formal de interfaces • Especificación de pre y post condiciones • Lenguajes formales para escribir y analizar pre y post condiciones. ▪ áreas de problemas bien atendidas por los métodos formales • Programación libre de bloqueos (*lock-free*), condiciones de carrera de datos • Sistemas asíncronos y distribuidos, bloqueos (*deadlock*), bloqueos vivos (*livelock*), etc. ▪ Comparación con otras herramientas y técnicas para la detección de defectos • Pruebas • *Fuzzing* ▪ Enfoques formales para el modelado y análisis de software • Verificadores de modelos (model checkers) • Buscadores de modelos (model finders)	▪ 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] ▪ Aplicar técnicas de especificación y análisis formal a diseños y programas de software de baja complejidad [Aplicar] ▪ Explicar los beneficios y desventajas potenciales del uso de lenguajes de especificación formal [Explicar]
Lecturas : [**Sommerville2015**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS3P3. Internet de las Cosas (Electivo)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS3P3. Internet de las Cosas
2.2 Semestre	:	9no Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Electivo
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS3P1. Computación Paralela y Distribuída. (8vo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Soporte de Movilidad (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Principios de comunicación celular (por ejemplo, 4G, 5G) ▪ Principios de LANs inalámbricas (principalmente 802.11) ▪ Comunicación dispositivo a dispositivo (por ejemplo, comunicación IoT) ▪ Redes inalámbricas multiescalón (por ejemplo, redes ad hoc, oportunistas, tolerantes a retrasos)	▪ Describir algunos aspectos de comunicación celular como el registro [Describir] ▪ Describir cómo 802.11 soporta usuarios móviles [Describir] ▪ Describir usos prácticos de comunicación dispositivo a dispositivo, así como multiescalón [Describir] ▪ Describir un tipo de red móvil como ad hoc [Describir]
Lecturas : [**Hanes2017**], [**Buyya2016**]	

Unidad 2: Ética, Ley y Sociedad en Seguridad (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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. ▪ 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.	▪ Analizar los impactos sociales de la vigilancia generalizada por IoT [Evaluar]. ▪ Aplicar principios de privacidad por diseño a arquitecturas de datos de IoT [Usar].
Lecturas : [**Hanes2017**]	

Unidad 3: Aplicaciones en Red (24 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación ▪ Protocolos para redes restringidas (CoAP, MQTT, LoRaWAN).	▪ Definir los principios de denominación, direccionamiento y localización de recursos [Definir] ▪ Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar] ▪ Describir los detalles de un protocolo de capa de aplicación [Describir] ▪ Implementar una aplicación simple cliente-servidor basada en sockets [Implementar]
Lecturas : [**Hanes2017**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS3P2. Cloud Computing (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS3P2. Cloud Computing
2.2 Semestre	:	10mo Semestre.
2.3 Créditos	:	3
2.4 Horas	:	1 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS370. Big Data. (9no Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La computación moderna depende en gran medida de arquitecturas distribuidas y plataformas nativas de la nube. Este curso proporciona una exploración en profundidad de los servicios de computación en la nube, conceptos de bases de datos distribuidas y plataformas web. Los estudiantes aprenderán a diseñar arquitecturas escalables, comprender las implicaciones de la consistencia de datos distribuidos y evaluar los desafíos de privacidad y seguridad inherentes a los entornos basados en la nube.

5. OBJETIVOS

- Comprender la arquitectura y los modelos de despliegue de la computación en la nube.
- Diseñar soluciones de bases de datos distribuidas considerando consistencia y disponibilidad.
- Desarrollar aplicaciones utilizando plataformas y frameworks web modernos basados en la nube.
- Analizar las implicaciones de seguridad y privacidad en los servicios distribuidos y en la nube.

6. RESULTADOS DEL ESTUDIANTE

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

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

7. TEMAS

Unidad 1: Arquitectura Cloud y Gestión de Datos (12 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Plataformas, frameworks o meta-frameworks web • Servicios en la nube • API, Componentes Web (Web Components) ▪ Servicios de computación Sistemas NoSQL • Alojamiento en la Nube (Cloud Hosting) • Escalabilidad (por ejemplo, Autoescalado (*Autoscaling*), Clústeres) • Estimación de costos para servicios ▪ Arquitectura • Monolitos vs Microservicios • Micro-frontends • Arquitecturas Dirigidas por Eventos (Event-Driven) vs RESTful: ventajas y desventajas • Sin servidor (Serverless), computación en la nube bajo demanda	▪ Diseñar e implementar una aplicación basada en web utilizando un diseño de arquitectura de microservicios [Diseñar] ▪ Describir las restricciones, como alojamiento, servicios y escalabilidad, relacionadas con las plataformas web [Describir]
Lecturas : [**Marinescu2022**], [**Buyya2013**]	

Unidad 2: Bases de Datos Distribuidas/Computación en la Nube (10 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Bases de datos distribuidas/sistemas basados en la nube	▪ Describir los componentes clave de un DBMS distribuido, incluyendo almacenamiento de datos distribuido, procesamiento de consultas y gestión de transacciones [Describir] ▪ Analizar las ventajas y desventajas entre arquitecturas de DBMS paralelo: memoria compartida, disco compartido y nada compartido [Analizar] ▪ Describir estrategias de replicación de datos y modelos de consistencia débil en sistemas de bases de datos distribuidas [Describir]
Lecturas : [**Buyya2013**]	

Unidad 3: Computación Paralela y Distribuida (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Seguridad (safety) y vivacidad (liveness): • Condiciones de carrera (race conditions) • Dependencias/precondiciones • Modelos de fallos (fault models) • Terminación ▪ Modelos de programación: • Modelo de actores (*actor models*) • Modelos procedurales y reactivos • Modelos de programación síncrona/asíncrona • Paralelismo de datos (*data parallelism*) ▪ Propiedades: • Propiedades basadas en orden: ◦ Conmutatividad ◦ Independencia • Propiedades basadas en consistencia: ◦ Atomicidad ◦ Consenso ▪ Control de ejecución: • Espera asíncrona *Async await* • Promesas (*promises*) • Hilos (*threads*) ▪ Comunicación y coordinación: • Mutexes • Paso de mensajes (*message-passing*) • Memoria compartida (*shared memory*) • Cobegin-coend • Monitores • Canales (*channels*) • Hilos (*threads*) • Guardas (*guards*) ▪ Futuros (futures) ▪ Soporte del lenguaje para paralelismo de datos como forall, desenrollado de bucles (loop unrolling), map/reduce ▪ Efecto de los modelos de consistencia de memoria en la semántica del lenguaje y la generación correcta de código ▪ Interfaces de Programación de Aplicaciones de Transferencia de Estado Representacional (REST APIs) ▪ Tecnologías y enfoques: computación en la nube (*cloud computing*), computación de alto rendimiento, computación cuántica, computación ubicua	▪ Explicar por qué los lenguajes de programación no garantizan la consistencia secuencial en presencia de condiciones de carrera (data races) y qué deben hacer los programadores como resultado [Explicar] ▪ Implementar programas concurrentes correctos usando múltiples modelos de programación, como memoria compartida, actores, futuros, construcciones de sincronización y primitivas de paralelismo de datos [Implementar] ▪ Usar un modelo de paso de mensajes para analizar un protocolo de comunicación [Usar] ▪ Usar construcciones de sincronización como métodos monitorizados/sincronizados en un programa simple [Usar] ▪ Modelar dependencia de datos usando construcciones de programación simples que involucran variables, lectura y escritura [Implementar] ▪ Modelar dependencia de control usando construcciones simples como selección e iteración [Implementar] ▪ Explicar cómo las APIs REST integran aplicaciones y automatizan procesos [Explicar] ▪ Explicar beneficios, limitaciones y desafíos relacionados con la computación distribuida y paralela [Explicar]

Unidad 4: Aplicaciones en Red (6 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Paradigmas de aplicaciones distribuidas (por ejemplo, cliente/servidor, peer-to-peer, nube, edge y fog). Ver también: Comunicación, Coordinación	▪ Definir los principios de denominación, direccionamiento y localización de recursos [Definir] ▪ Analizar las necesidades de demandas específicas de aplicaciones en red [Analizar] ▪ Describir los detalles de un protocolo de capa de aplicación [Describir] ▪ Implementar una aplicación simple cliente-servidor basada en sockets [Implementar]
Lecturas : [**Marinescu2022**]	

Unidad 5: Algoritmos: Dominios de Aplicación (4 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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). • 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. • 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. • 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. • 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*). • Nube: Agentes de ejecución típicos: Hosts aprovisionados. Mecanismos de comunicación típicos: APIs de servicio. Dominios algorítmicos típicos: Aplicaciones web. Objetivos de ingeniería típicos: Escalabilidad. • 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.	▪ 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] ▪ Mejorar el rendimiento de un componente secuencial introduciendo paralelismo y/o distribución [Crear] ▪ Elegir entre diferentes diseños paralelos/distribuidos para componentes de un sistema dado [Evaluar (valorar)]
Lecturas : [**Marinescu2022**]	

Unidad 6: Principios y Mentalidad de Seguridad (8 horas)	
Resultados esperados: AG-C09,AG-C11	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aplicaciones de una mentalidad de seguridad: web, nube y dispositivos móviles Diseño del Sistema, Aspectos Comunes: Plataformas, APIs y Restricciones, Aspectos Comunes: Lenguajes de Plataforma y Patrones Web. ▪ 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.	▪ Analizar los riesgos de seguridad asociados con aplicaciones en la nube distribuidas [Usar]. ▪ Evaluar las implicaciones de privacidad del almacenamiento de datos en entornos de nube multiinquilino [Evaluar].
Lecturas : [**Marinescu2022**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

CS404. Taller de Investigación (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	CS404. Taller de Investigación
2.2 Semestre	:	10mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	CS403. Proyecto de tesis 2. (9no Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

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.

5. OBJETIVOS

- 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.

6. RESULTADOS DEL ESTUDIANTE

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

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

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

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

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

7. TEMAS

Unidad 1: Conclusión de Experimentos y Análisis de Resultados (12 horas)	
Resultados esperados: AG-C05,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Ejecución completa del protocolo experimental definido en el plan de tesis. ▪ Análisis estadístico final: pruebas de significancia, intervalos de confianza y efecto. ▪ Comparación objetiva con baselines y estado del arte usando métricas establecidas. ▪ Interpretación de resultados: hallazgos, limitaciones y amenazas a la validez. ▪ Documentación reproducible: código, datasets, configuraciones y resultados.	▪ Completar la ejecución del protocolo experimental con todos los casos definidos. [Evaluar] ▪ Analizar resultados con rigor estadístico e interpretar hallazgos con respecto a la hipótesis. [Evaluar] ▪ Documentar de forma reproducible los experimentos, configuraciones y resultados. [Usar]
Lecturas : [**ieee**], [**acm**]	

Unidad 2: Redacción de Capítulos Finales de la Tesis (12 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redacción del capítulo de resultados: tablas, figuras y análisis comparativo de alta calidad. ▪ Redacción del capítulo de conclusiones: hallazgos, contribuciones originales y limitaciones. ▪ Redacción del capítulo de trabajo futuro: extensiones y problemas abiertos. ▪ Redacción de la introducción final: motivación, problema y resumen de contribuciones. ▪ Redacción del resumen ejecutivo (abstract) en español e inglés.	▪ Redactar el capítulo de resultados con visualizaciones claras y análisis comparativo. [Evaluar] ▪ Formular las conclusiones y contribuciones originales de la tesis con precisión. [Evaluar] ▪ Escribir el abstract y la introducción final articulando el problema y los aportes. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 3: Formato, Integración y Calidad del Documento (10 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Verificación del cumplimiento del formato institucional de tesis en LaTeX. ▪ Revisión de calidad de figuras, tablas y ecuaciones en el documento final. ▪ Consistencia narrativa y terminológica entre todos los capítulos del documento. ▪ Revisión y actualización de la bibliografía: cobertura, formato BibTeX y citas correctas. ▪ Generación del documento final: PDF, portada, índices y apéndices.	▪ Verificar que el documento cumple íntegramente con el formato institucional de tesis. [Evaluar] ▪ Corregir la consistencia narrativa y terminológica de todos los capítulos. [Evaluar] ▪ Producir el documento PDF final con portada, índices y bibliografía completos. [Usar]
Lecturas : [**ieee**], [**acm**]	

Unidad 4: Revisiones con el Asesor y Ciclos de Corrección (10 horas)	
Resultados esperados: AG-C02,AG-C08,AG-C09	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Dinámica de revisión estructurada: entrega, revisión y respuesta al asesor. ▪ Incorporación de observaciones del asesor: priorización y respuesta argumentada. ▪ Revisión de la solidez de la propuesta y la validez de los experimentos según el asesor. ▪ Manejo de desacuerdos académicos con el asesor de forma constructiva y ética. ▪ Criterios de aprobación del asesor para la autorización de sustentación.	▪ Incorporar sistemáticamente las observaciones del asesor con respuestas argumentadas. [Evaluar] ▪ Mejorar la solidez científica del documento a través de ciclos de revisión. [Evaluar] ▪ Obtener la aprobación formal del asesor para la sustentación del documento. [Evaluar]
Lecturas : [**ieee**], [**acm**]	

Unidad 5: Preparación y Ensayo de la Sustentación (8 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Diseño de la presentación oral: estructura, narrativa y manejo del tiempo. ▪ Preparación de diapositivas técnicas: figuras, resultados y comparaciones clave. ▪ Técnicas de defensa ante preguntas del jurado: argumentación y manejo de objeciones. ▪ Ensayo de la sustentación con el asesor y/o pares académicos. ▪ Gestión del estrés y comunicación efectiva en contextos de evaluación académica.	▪ Diseñar y exponer una presentación oral clara y estructurada del trabajo de tesis. [Evaluar] ▪ Defender la metodología y resultados ante preguntas críticas del jurado. [Evaluar] ▪ Demostrar dominio completo del tema mediante ensayos de sustentación previos. [Usar]
Lecturas : [**ieee**], [**acm**]	

Unidad 6: Entrega Formal y Pre-Sustentación (8 horas)	
Resultados esperados: AG-C02,AG-C04,AG-C05	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Proceso institucional de entrega de la tesis: documentación y plazos reglamentarios. ▪ Pre-sustentación ante autoridades de la facultad: evaluación del borrador final. ▪ Incorporación de correcciones del comité presustentador en el documento. ▪ Trámites administrativos: designación de jurado y programación de la sustentación. ▪ Ética en la diseminación: acceso abierto, repositorio institucional y derechos de autoría.	▪ Entregar el documento de tesis completo cumpliendo todos los requisitos institucionales. [Evaluar] ▪ Presentar el borrador final ante el comité presustentador e incorporar sus correcciones. [Evaluar] ▪ Completar los trámites administrativos para la programación de la sustentación pública. [Usar]
Lecturas : [**ieee**], [**acm**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI367. Tópicos en Inteligencia Artificial (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI367. Tópicos en Inteligencia Artificial
2.2 Semestre	:	10mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI264. Aprendizaje Profundo. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Inteligencia Artificial es un campo en rápida evolución. Este curso explora temas avanzados y contemporáneos que están dando forma actualmente a la industria y la investigación. Se pone énfasis en la IA Generativa, Modelos de Lenguaje Grandes (LLMs) y Aprendizaje por Refuerzo Avanzado. Los estudiantes analizarán las arquitecturas detrás de los sistemas modernos de IA, como los Transformers, y discutirán los desafíos técnicos y sociales de desplegar agentes inteligentes autónomos a gran escala.

5. OBJETIVOS

- Comprender la arquitectura y el proceso de entrenamiento de Modelos de Lenguaje Grandes.
- Dominar los principios de la IA Generativa para síntesis de texto e imágenes.
- Implementar algoritmos avanzados de Aprendizaje por Refuerzo para la toma de decisiones.
- Aplicar técnicas de ingeniería de prompts y ajuste fino a modelos existentes.
- Evaluar críticamente los últimos artículos de investigación y tendencias en IA.

6. RESULTADOS DEL ESTUDIANTE

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

7. TEMAS

Unidad 1: Fundamentos de Aprendizaje Automático (6 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: • Aprendizaje supervisado: ◦ Clasificación ◦ Regresión • Aprendizaje por refuerzo • Aprendizaje no supervisado: ◦ Agrupamiento ▪ Ideas fundamentales: • Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. • Fuentes de error e indecidibilidad en el aprendizaje automático ▪ Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: • Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente ▪ El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): • El compromiso entre sesgo (subajuste) - varianza (sobreajuste)	▪ 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] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] ▪ 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] ▪ Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] ▪ Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]
Lecturas : [**Goodfellow2016**], [**Foster2022**]	

Unidad 2: Aprendizaje Automático: Datos, Representaciones y Evaluación (5 horas)

Resultados esperados: AG-C06

Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Trabajar con Datos: • Preprocesamiento de datos: ◦ Importancia y peligros de las elecciones de preprocesamiento • Manejo de valores faltantes (imputación, marcar como faltante): ◦ Implicaciones de imputar versus marcar como faltante • Codificación de variables categóricas, codificación de datos de valor real • Normalización/estandarización • énfasis en datos reales, no ejemplos de libros de texto ▪ Representaciones: • Espacios de hipótesis y complejidad • Expansión de características de base simple, como elevar al cuadrado características univariadas • Representaciones de características aprendidas ▪ Evaluación del aprendizaje automático: • Separación de conjuntos de entrenamiento, validación y prueba • Métricas de rendimiento para clasificadores • Estimación del rendimiento en prueba en datos retenidos • Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación • Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones	▪ Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] ▪ Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] ▪ 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] ▪ 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]
Lecturas : [**Goodfellow2016**], [**Foster2022**]	

Unidad 3: Redes Neuronales y Ética (5 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redes neuronales básicas: • Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos • Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) ▪ ética para el Aprendizaje Automático: • Enfocarse en datos reales, escenarios reales y estudios de caso • Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas	▪ Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]
Lecturas : [**Goodfellow2016**], [**Foster2022**]	

Unidad 4: Formulación del Aprendizaje Automático y Evaluación (3 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Formulación del aprendizaje automático simple como un problema de optimización, como regresión lineal por mínimos cuadrados o regresión logística: • Función objetivo • Descenso de gradiente • Regularización para evitar sobreajuste (formulación matemática) ▪ Conjuntos de modelos: • Combinación de mayoría ponderada simple ▪ Evaluación de rendimiento: • Otras métricas para clasificación (por ejemplo, error, precisión, exhaustividad) • Métricas de rendimiento para regresores • Matriz de confusión • Validación cruzada: ◦ Ajuste de parámetros (búsqueda en cuadrícula/aleatoria, mediante validación cruzada)	▪ 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] ▪ Evaluar el rendimiento de un sistema de aprendizaje simple en un conjunto de datos del mundo real [Evaluar] ▪ Caracterizar el estado del arte en teoría del aprendizaje, incluyendo sus logros y limitaciones [Explicar] ▪ Explicar el problema del sobreajuste, junto con técnicas para detectar y manejar el problema [Explicar] ▪ Explicar el triple compromiso entre el tamaño de un espacio de hipótesis, el tamaño del conjunto de entrenamiento y la precisión del rendimiento [Explicar]
Lecturas : [**Goodfellow2016**]	

Unidad 5: Aprendizaje Profundo, RL y Ética (3 horas)	
Resultados esperados:	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Aprendizaje profundo: • Redes feed-forward profundas (solo intuición, sin matemáticas) • Redes neuronales convolucionales (solo intuición, sin matemáticas) • Visualización de representaciones de características aprendidas de redes profundas • Otras arquitecturas (NN generativas, NN recurrentes, transformers, etc.) ▪ Visión general de los métodos de aprendizaje por refuerzo ▪ Dos o más aplicaciones de algoritmos de aprendizaje automático: • Por ejemplo, medicina y salud, economía, visión, lenguaje natural, robótica, juego ▪ ética para el Aprendizaje Automático: • Continuar enfocándose en datos reales, escenarios reales y estudios de caso • Privacidad • Equidad • Propiedad intelectual • Explicabilidad	▪ 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] ▪ 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]
Lecturas : [**Goodfellow2016**]	

Unidad 6: Procesamiento del Lenguaje Natural (20 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Gramáticas deterministas y estocásticas ▪ Algoritmos de análisis sintáctico: • CFGs y analizadores de tabla (por ejemplo, CYK) • CFGs probabilísticos y CYK ponderado ▪ Representación del significado/Semántica: • Representaciones del conocimiento basadas en lógica • Roles semánticos • Representaciones temporales • Creencias, deseos e intenciones ▪ Métodos basados en corpus ▪ N-gramas y HMMs ▪ Suavizado y retroceso ▪ Ejemplos de uso: Etiquetado POS y morfología ▪ Recuperación de información: • Modelo de espacio vectorial: ◦ TF & IDF • Precisión y exhaustividad ▪ Extracción de información ▪ Traducción de lenguaje ▪ Clasificación de texto, categorización: • Modelo de bolsa de palabras ▪ Aprendizaje profundo para PLN: • RNNs • Transformers • Incrustaciones multimodales (por ejemplo, imágenes + texto) • Modelos de lenguaje generativos	▪ Definir y contrastar gramáticas deterministas y estocásticas, proporcionando ejemplos para mostrar la adecuación de cada una [Definir] ▪ Simular, aplicar o implementar algoritmos clásicos y estocásticos para analizar lenguaje natural [Aplicar] ▪ Identificar los desafíos de representar el significado [Analizar] ▪ Listar las ventajas de usar corpus estándar. Identificar ejemplos de corpus actuales para una variedad de tareas de PLN [Listar/Enumerar] ▪ Identificar técnicas para recuperación de información, traducción de lenguaje y clasificación de texto [Analizar] ▪ 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]
Lecturas : [**Tunstall2022**], [**Russell2020**]	

Unidad 7: Fundamentos de Aprendizaje Automático (6 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Definición y ejemplos de una amplia variedad de tareas de aprendizaje automático: • Aprendizaje supervisado: ◦ Clasificación ◦ Regresión • Aprendizaje por refuerzo • Aprendizaje no supervisado: ◦ Agrupamiento ▪ Ideas fundamentales: • Teorema de no hay almuerzo gratis: ningún aprendiz puede resolver todos los problemas; las decisiones de diseño de representación tienen consecuencias. • Fuentes de error e indecidibilidad en el aprendizaje automático ▪ Un aprendizaje supervisado simple basado en estadísticas como regresión lineal o árboles de decisión: • Enfocarse en cómo funcionan sin entrar en detalles matemáticos o de optimización; suficiente para entender y usar implementaciones existentes correctamente ▪ El problema del sobreajuste/control de la complejidad de la solución (regularización, poda - solo intuición): • El compromiso entre sesgo (subajuste) - varianza (sobreajuste)	▪ 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] ▪ Diferenciar los términos: IA, aprendizaje automático y aprendizaje profundo [Evaluar] ▪ 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] ▪ Identificar el sobreajuste en el contexto de un problema y curvas de aprendizaje y describir soluciones al sobreajuste [Evaluar] ▪ Explicar cómo funciona el aprendizaje automático como un proceso de optimización/búsqueda [Explicar]
Lecturas : [**Sutton2018**]	

Unidad 8: Aprendizaje Automático: Datos, Representaciones y Evaluación (5 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Trabajar con Datos: • Preprocesamiento de datos: ◦ Importancia y peligros de las elecciones de preprocesamiento • Manejo de valores faltantes (imputación, marcar como faltante): ◦ Implicaciones de imputar versus marcar como faltante • Codificación de variables categóricas, codificación de datos de valor real • Normalización/estandarización • énfasis en datos reales, no ejemplos de libros de texto ▪ Representaciones: • Espacios de hipótesis y complejidad • Expansión de características de base simple, como elevar al cuadrado características univariadas • Representaciones de características aprendidas ▪ Evaluación del aprendizaje automático: • Separación de conjuntos de entrenamiento, validación y prueba • Métricas de rendimiento para clasificadores • Estimación del rendimiento en prueba en datos retenidos • Ajuste de los parámetros de un modelo de aprendizaje automático con un conjunto de validación • Importancia de entender qué está haciendo un modelo, dónde están sus peligros/limitaciones y las implicaciones de sus decisiones	▪ Aplicar dos o más algoritmos de aprendizaje estadístico simple a una tarea de clasificación y medir la precisión de los clasificadores [Aplicar] ▪ Implementar un algoritmo de aprendizaje estadístico y el proceso de optimización correspondiente para entrenar el clasificador y obtener una predicción en nuevos datos [Aplicar] ▪ 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] ▪ 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]
Lecturas : [**Sutton2018**]	

Unidad 9: Redes Neuronales y Ética (5 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Redes neuronales básicas: • Fundamentos de comprensión de cómo funcionan las redes neuronales y su proceso de entrenamiento, sin detalles de los cálculos • Introducción básica a redes neuronales generativas (por ejemplo, modelos de lenguaje grandes) ▪ ética para el Aprendizaje Automático: • Enfocarse en datos reales, escenarios reales y estudios de caso • Sesgo de conjunto de datos/algorítmico/de evaluación y consecuencias no deseadas	▪ Describir el proceso de entrenamiento de redes neuronales y las representaciones aprendidas resultantes [Explicar]
Lecturas : [**Sutton2018**]	

Unidad 10: Aplicaciones e Impacto Social de la IA (6 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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): • Formular y evaluar una aplicación específica como un problema de IA: ◦ Cómo lidiar con problemas mal especificados o mal planteados • Disponibilidad/escasez y limpieza de datos: ◦ Limpieza y preprocesamiento básico de datos ◦ Sesgo del conjunto de datos • Sesgo algorítmico • Sesgo de evaluación • Evaluación de las implicaciones sociales de la aplicación ▪ Modelos generativos profundos desplegados: • 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. • Visión general de alto nivel de modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard, etc.), sus usos y sus limitaciones/peligros. ▪ Visión general del impacto social de la IA: • ética • Equidad • Confianza/explicabilidad • Privacidad y uso de datos de entrenamiento • Autonomía humana y supervisión/regulaciones/requisitos legales • Sostenibilidad	▪ 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]
Lecturas : [**Russell2020**], [**Bostrom2014**]	

Unidad 11: Aplicaciones Adicionales e Impacto Social de la IA (6 horas)	
Resultados esperados: AG-C06	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ 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): • Formular y evaluar una aplicación específica como un problema de IA: ◦ Cómo lidiar con problemas mal especificados o mal planteados • Disponibilidad/escasez y limpieza de datos: ◦ Limpieza y preprocesamiento básico de datos ◦ Sesgo del conjunto de datos • Sesgo algorítmico • Sesgo de evaluación • Evaluación de las implicaciones sociales de la aplicación ▪ Profundidad adicional sobre modelos generativos profundos desplegados: • Introducción a cómo funcionan los modelos generativos de imágenes profundas (por ejemplo, a 2023, DALL-E, Midjourney, Stable Diffusion) incluyendo discusión de atención • Introducción a cómo funcionan los modelos de lenguaje grandes (por ejemplo, a 2023, ChatGPT, Bard) incluyendo discusión de atención • Idea de modelos fundamentales, cómo usarlos, y los beneficios/problemas con entrenarlos a partir de grandes datos ▪ Análisis y discusión del impacto social de la IA: • ética • Equidad • Confianza/explicabilidad • Privacidad y uso de datos de entrenamiento • Autonomía humana y supervisión/regulaciones/requisitos legales • Sostenibilidad	▪ 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] ▪ 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]
Lecturas : [**Russell2020**], [**Bostrom2014**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA

Universidad Nacional de Ingeniería (UNI)
Escuela Profesional de
Ciencia de la Computación
Sílabo 2027-I

1. CURSO

AI368. Computación Evolutiva (Obligatorio)

2. INFORMACIÓN GENERAL

2.1 Curso	:	AI368. Computación Evolutiva
2.2 Semestre	:	10mo Semestre.
2.3 Créditos	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duración del periodo	:	16 semanas
2.6 Condición	:	Obligatorio
2.7 Modalidad de aprendizaje	:	Presencial
2.8 Prerrequisitos	:	AI264. Aprendizaje Profundo. (7mo Sem)

3. PROFESORES

Atención previa coordinación con el profesor

4. INTRODUCCIÓN AL CURSO

La Computación Evolutiva es un subcampo de la Inteligencia Artificial inspirado en la selección natural y la evolución biológica. Este curso explora algoritmos metaheurísticos de optimización como Algoritmos Genéticos, Estrategias de Evolución y Programación Genética. Estas técnicas son esenciales para resolver problemas complejos de optimización y búsqueda donde los métodos analíticos tradicionales son insuficientes. Los estudiantes aprenderán a diseñar, implementar y analizar sistemas evolutivos aplicados a ingeniería, ciencia de datos y agentes autónomos.

5. OBJETIVOS

- Entender los principios de la selección natural aplicados a la computación.
- Dominar el diseño de representaciones, operadores y funciones de aptitud (fitness).
- Implementar y ajustar Algoritmos Genéticos y Estrategias de Evolución.
- Analizar la convergencia y el rendimiento de algoritmos bioinspirados.
- Aplicar técnicas evolutivas a problemas de optimización multiobjetivo.

6. RESULTADOS DEL ESTUDIANTE

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

AG-C12) Usa enfoques sistémicos para seleccionar, desarrollar, aplica, integrar y administrar tecnologías seguras de computación para lograr los objetivos del usuario. (Usage)

7. TEMAS

Unidad 1: Fundamentos de la Computación Evolutiva (12 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Introducción a los Algoritmos Evolutivos (EAs) y metáforas biológicas. ▪ El ciclo general del Algoritmo Evolutivo: Inicialización, Selección, Variación, Reemplazo. ▪ Componentes: Poblaciones, cromosomas, genes, alelos. ▪ Paisajes de aptitud (*fitness, landscapes*), mapeo genotipo-fenotipo y espacios de búsqueda.	▪ Explicar la relación entre la evolución biológica y los procesos de búsqueda artificial [Familiarizarse] ▪ Identificar los componentes de un algoritmo evolutivo genérico y sus roles [Usar] ▪ Describir el concepto de paisajes de aptitud y su impacto en el rendimiento de los EAs [Familiarizarse]
Lecturas : [**Eiben2015**], [**Back1996**]	

Unidad 2: Algoritmos Genéticos y sus Variantes (16 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Algoritmos Genéticos Canónicos (CGA): representación binaria, esquemas de selección (ruleta, torneo). ▪ Operadores Genéticos: Cruza (un punto, uniforme, aritmética), Mutación (bit-flip, Gaussiana). ▪ Teorema de Esquemas e Hipótesis de Bloques de Construcción. ▪ Variantes avanzadas de AG: AG con codificación real, Programación Genética (GP) con representaciones de árbol.	▪ Diseñar representaciones y operadores apropiados para problemas específicos de optimización [Evaluar] ▪ Implementar un Algoritmo Genético Canónico para optimización de funciones [Usar] ▪ Aplicar Programación Genética a tareas de regresión simbólica o programación automática [Evaluar]
Lecturas : [**Eiben2015**], [**Mitchell1998**]	

Unidad 3: Estrategias de Evolución y Programación Evolutiva (18 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Estrategias de Evolución (ES): mecanismos de selección (μ, λ) y $(\mu + \lambda)$. ▪ Auto-adaptación de parámetros de estrategia (tamaños de paso de mutación). ▪ Adaptación de Matriz de Covarianza (CMA-ES). ▪ Programación Evolutiva (EP) y su enfoque en la evolución de comportamiento.	▪ Implementar Estrategias de Evolución con mutación auto-adaptativa para optimización continua [Evaluar] ▪ Comparar el rendimiento de AG vs ES en problemas de referencia [Usar] ▪ Explicar el papel de la adaptación de parámetros de estrategia en la velocidad de convergencia [Familiarizarse]
Lecturas : [**Eiben2015**], [**Deb2001**]	

Unidad 4: Algoritmos Evolutivos Multiobjetivo e Híbridos (18 horas)	
Resultados esperados: AG-C08,AG-C12	
Temas	**Aprendizaje esperado (*Learning Outcomes*)**
▪ Optimización Multiobjetivo: optimalidad de Pareto, relaciones de dominancia. ▪ EAs Multiobjetivo: NSGA-II, SPEA2, MOEA/D. ▪ Mecanismos de preservación de diversidad y estrategias de archivo. ▪ Algoritmos Meméticos: Hibridización con búsqueda local. ▪ Técnicas de manejo de restricciones en búsqueda evolutiva.	▪ Resolver problemas multiobjetivo usando métodos de selección basados en Pareto [Evaluar] ▪ Diseñar un algoritmo memético combinando búsqueda global y local [Evaluar] ▪ Evaluar las compensaciones entre convergencia y diversidad en MOEAs [Usar]
Lecturas : [**Eiben2015**], [**Deb2001**]	

8. PLAN DE TRABAJO

8.1 Metodología

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.2 Sesiones Teóricas

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

8.3 Sesiones Prácticas

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

9. SISTEMA DE EVALUACIÓN

Sesiones Teóricas:
Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Sesiones de Laboratorio:
Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, realizarán actividades que les permita aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrá retos que permitan evaluar el desempeño de los alumnos.

Exposiciones individuales o grupales:
Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:
A lo largo del curso se proporcionan diferentes lecturas, las cuales podrían ser evaluadas. El uso de herramientas Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

La nota final NF depende de de las siguientes notas:

- EP: Examen Parcial.
- EF: Examen Final.
- P: Evaluacion permanente (Promedio de prácticas y Trabajos calificados).

La nota final NF se calcula así:

$$NF = \frac{EP + EF + P}{3}$$

10. BIBLIOGRAFÍA BÁSICA