

Universidad Nacional de Moquegua (UNAM)  
Facultad de Ingenierías  
Departamento de -  
Escuela Profesional de Ciencia de la Computación



## Libro de Sílabos

– 2026-I–

Moquegua: 26 de diciembre de 2025

# Equipo de Trabajo

## **Dr. Anibal F. Flores Garcia**

Presidente de la Comisión de Nuevos Programas DAISI

Director del Departamento Académico de Ingeniería de Sistemas e Informática

email: [afloresg@unam.edu.pe](mailto:afloresg@unam.edu.pe)

## **M.Sc. Honorio Apaza Alanoca**

Miembro de la Comisión de Nuevos Programas DAISI

email: [hapazaa@unam.edu.pe](mailto:hapazaa@unam.edu.pe)

## **Ernesto Cuadros-Vargas (Editor)**

Orador distinguido para la *Association of Computing Machinery* (ACM)

Miembro del Directorio de Gobernadores de la Sociedad de Computación del IEEE (2020-2023)

Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula 2020 (CS2020)*

Miembro del *Steering Committee de ACM/IEEE-CS Computing Curricula for Computer Science (CS2013)*

Presidente de la Sociedad Peruana de Computación (SPC) 2001-2007, 2009

email: [ecuadros@spc.org.pe](mailto:ecuadros@spc.org.pe)

# Índice general

<b>1. Primer Semestre</b>	<b>5</b>
1.1. CS111. Introducción a la Programación . . . . .	6
1.2. BMA101. Álgebra Lineal . . . . .	13
1.3. BMA102. Cálculo Diferencial . . . . .	16
1.4. BCH101. Química I . . . . .	19
1.5. FG001. Electivo Formación General . . . . .	22
1.6. BEI101. Inglés I . . . . .	24
<b>2. Segundo Semestre</b>	<b>29</b>
2.1. CS100. Introducción a la Ciencia de la Computación . . . . .	30
2.2. CS112. Programación Orientada a Objetos I . . . . .	44
2.3. CS1D1. Estructuras Discretas . . . . .	52
2.4. BMA103. Cálculo Integral . . . . .	58
2.5. BFI101. Física I . . . . .	61
2.6. BEI102. Inglés II . . . . .	65
<b>3. Tercer Semestre</b>	<b>69</b>
3.1. CS113. Programación Orientada a Objetos II . . . . .	70
3.2. CS2B1. Desarrollo Basado en Plataformas . . . . .	74
3.3. AI161. IA Aplicada . . . . .	79
3.4. BMA104. Cálculo Diferencial e Integral Avanzado . . . . .	83
3.5. ST251. Estadística y Probabilidades . . . . .	87
3.6. BEI201. Inglés III . . . . .	91
<b>4. Cuarto Semestre</b>	<b>95</b>
4.1. CS210. Algoritmos y Estructuras de Datos . . . . .	96
4.2. CS211. Teoría de la Computación . . . . .	99
4.3. CS221. Arquitectura de Computadores . . . . .	103
4.4. CS271. Bases de Datos I . . . . .	109
4.5. CS401. Metodología de la Investigación . . . . .	114
4.6. MA202. Métodos Numéricos . . . . .	116
4.7. BEI202. Inglés IV . . . . .	119
<b>5. Quinto Semestre</b>	<b>123</b>
5.1. CS212. Análisis y Diseño de Algoritmos . . . . .	124
5.2. CS261. Inteligencia Artificial . . . . .	129
5.3. CS272. Bases de Datos II . . . . .	138
5.4. CS291. Ingeniería de Software I . . . . .	142

5.5. CS2S1. Sistemas Operativos . . . . .	154
5.6. BEI203. Inglés V . . . . .	162
<b>6. Sexto Semestre</b>	<b>167</b>
6.1. CS231. Redes y Comunicación . . . . .	168
6.2. CS311. Programación Competitiva . . . . .	172
6.3. CS312. Estructuras de Datos Avanzadas . . . . .	176
6.4. CS342. Compiladores . . . . .	180
6.5. AI263. Introducción al Aprendizaje de Máquina . . . . .	185
6.6. FI201. Física Computacional . . . . .	188
<b>7. Séptimo Semestre</b>	<b>191</b>
7.1. CS251. Computación Gráfica . . . . .	192
7.2. CS292. Ingeniería de Software II . . . . .	199
7.3. CS2H1. Experiencia de Usuario (UX) . . . . .	205
7.4. AI264. Aprendizaje Profundo . . . . .	211
7.5. BBI101. Biología . . . . .	213
7.6. FG211. Ética Profesional . . . . .	217
<b>8. Octavo Semestre</b>	<b>221</b>
8.1. CS281. Computación en la Sociedad . . . . .	222
8.2. CS3I1. Seguridad en Computación . . . . .	231
8.3. CS3P1. Computación Paralela y Distribuida . . . . .	241
8.4. CS402. Proyecto de tesis 1 . . . . .	247
8.5. FG106. Teatro . . . . .	249
8.6. EX301. Actividades Extracurriculares . . . . .	253
8.7. AI268. Visión Computacional . . . . .	255
8.8. CS391. Ingeniería de Software III . . . . .	257
8.9. CS393. Sistemas de Infomación . . . . .	263
<b>9. Noveno Semestre</b>	<b>265</b>
9.1. CS370. Big Data . . . . .	266
9.2. CS400. Prácticas Pre-profesionales . . . . .	269
9.3. CS403. Proyecto de tesis 2 . . . . .	271
9.4. AI365. Modelos Generativos Avanzados en IA . . . . .	274
9.5. CB309. Bioinformática . . . . .	276
9.6. CS351. Tópicos en Computación Gráfica . . . . .	280
9.7. CS353. Computación Cuántica . . . . .	283
9.8. AI369. Robótica . . . . .	286
9.9. CS392. Tópicos en Ingeniería de Software . . . . .	288
9.10. CS3P3. Internet de las Cosas . . . . .	293
<b>10. Décimo Semestre</b>	<b>299</b>
10.1. CS3P2. Cloud Computing . . . . .	300
10.2. CS404. Taller de Investigación . . . . .	304
10.3. AI367. Tópicos en Inteligencia Artificial . . . . .	306
10.4. AI368. Computación Evolutiva . . . . .	308
10.5. FG350. Liderazgo y Desempeño . . . . .	310

---

# Capítulo 1

# Primer Semestre

## 1. CURSO

CS111-CS2023. Introducción a la Programación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS111-CS2023. Introducción a la Programación
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este es el primer curso en la secuencia de los cursos introductorios a la Ciencia de la Computación. En este curso se pretende cubrir los conceptos señalados por la Computing Curricula ACM/IEEE-CS 2023. La programación es uno de los pilares de la Ciencia de la Computación; cualquier profesional del Área, necesitará programar para concretizar sus modelos y propuestas. Este curso introducción a los participantes en los conceptos fundamentales de este arte. Lo tópicos incluyen tipos de datos, estructuras de control, funciones, listas, recursividad y la mecánica de la ejecución, prueba y depuración.

## 5. OBJETIVOS

- Introducir los conceptos fundamentales de programación.
- Desarrollar su capacidad de abstracción utilizar un lenguaje de programación.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

**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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Historia (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Pre-historia – El mundo antes de 1946.</li> <li>• Historia del hardware, software, redes.</li> <li>• Pioneros de la Computación.</li> <li>• Historia de Internet.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse]</li> <li>• Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse]</li> <li>• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]</li> <li>• Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Evaluar]</li> </ul>

**Lecturas :** [BB19], [Gut13], [Zel10]

<b>Unidad 2: Sistemas de tipos básicos (2 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tipos como conjunto de valores junto con un conjunto de operaciones. <ul style="list-style-type: none"> <li>– Tipos primitivos (p.e. números, booleanos)</li> <li>– Composición de tipos construidos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias)</li> </ul> </li> <li>• Asociación de tipos de variables, argumentos, resultados y campos.</li> <li>• Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos.</li> </ul>	<ul style="list-style-type: none"> <li>• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse]</li> <li>• Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse]</li> <li>• Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse]</li> <li>• Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar]</li> <li>• Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]</li> <li>• Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]</li> </ul>

**Lecturas :** [Gut13], [Zel10]

Unidad 3: Conceptos Fundamentales de Programación (9 horas)	
Resultados esperados: 1	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Sintaxis y semántica básica de un lenguaje de alto nivel.</li> <li>• Variables y tipos de datos primitivos (ej., números, caracteres, booleanos)</li> <li>• Expresiones y asignaciones.</li> <li>• Operaciones básicas I/O.</li> <li>• Estructuras de control condicional e iterativas.</li> <li>• Funciones definidas por el usuario.</li> <li>• Paso de funciones y parámetros.</li> <li>• Concepto de recursividad.</li> </ul>	<ul style="list-style-type: none"> <li>• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]</li> <li>• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]</li> <li>• Escribe programas que usan tipos de datos primitivos [Usar]</li> <li>• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]</li> <li>• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]</li> <li>• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Familiarizarse]</li> <li>• Describe el concepto de recursividad y da ejemplos de su uso [Evaluar]</li> <li>• Identifica el caso base y el caso general de un problema basado en recursividad [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Gut13], [Zel10]	

Unidad 4: Análisis Básico (2 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.</li> <li>• Definición formal de la Notación Big O.</li> <li>• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.</li> <li>• Uso de la notación Big O.</li> <li>• Análisis de algoritmos iterativos y recursivos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique a qué se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]</li> <li>• En el contexto de algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Familiarizarse]</li> <li>• Indique la definición formal de Big O [Familiarizarse]</li> <li>• Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar]</li> <li>• Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar]</li> </ul>
<b>Lecturas :</b> [Gut13], [Zel10]	

**Unidad 5: Algoritmos y Estructuras de Datos fundamentales (8 horas)**

**Resultados esperados: 1**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo.</li> <li>Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción)</li> <li>Algoritmos de ordenamiento con peor caso o caso promedio en <math>O(N \lg N)</math></li> <li>Algoritmos de búsqueda secuencial y binaria.</li> </ul>	<ul style="list-style-type: none"> <li>Implementar algoritmos numéricos básicos [Usar]</li> <li>Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar]</li> <li>Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y <math>O(N \log N)</math> [Usar]</li> <li>Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento y búsqueda.</li> <li>Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse]</li> <li>Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto específico [Evaluar]</li> <li>Trazar y/o implementar un algoritmo de comparación de string [Usar]</li> </ul>

**Lecturas :** [Gut13], [Zel10]

**Unidad 6: Programación orientada a objetos (4 horas)**

**Resultados esperados: 1**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Lenguajes orientados a objetos para la encapsulación: Privacidad y visibilidad de miembros de la clase.</li> <li>• Definición de las categorías, campos, métodos y constructores.</li> <li>• Subclases y herencia.</li> <li>• Asignación dinámica: definición de método de llamada.</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar e implementar una clase [Usar]</li> <li>• Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Familiarizarse]</li> <li>• Comparar y contrastar (1) el enfoque procedural/funcional- definiendo una función por cada operación con el cuadro de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Familiarizarse]</li> <li>• Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse]</li> <li>• Usar mecanismos de encapsulación orientada a objetos [Familiarizarse].</li> </ul>

**Lecturas :** [Gut13], [Zel10]

<b>Unidad 7: Algoritmos y Diseño (9 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> <li>Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones)</li> </ul> </li> <li>Rol de los algoritmos en el proceso de solución de problemas</li> <li>Estrategias de solución de problemas <ul style="list-style-type: none"> <li>Funciones matemáticas iterativas y recursivas</li> <li>Recorrido iterativo y recursivo en estructura de datos</li> <li>Estrategias Divide y Conquistar</li> </ul> </li> <li>Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> <li>Abstracción</li> <li>Descomposición de Programa</li> <li>Encapsulamiento y camuflaje de información</li> <li>Separación de comportamiento y aplicación</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse]</li> <li>Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse]</li> <li>Crea algoritmos para resolver problemas simples [Usar]</li> <li>Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar]</li> <li>Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar]</li> <li>Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar]</li> <li>Implementa un algoritmo de divide y vencerás para resolver un problema [Usar]</li> <li>Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar]</li> <li>Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar]</li> <li>Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar]</li> <li>Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]</li> </ul>

**Lecturas :** [Gut13], [Zel10]

<b>Unidad 8: Métodos de Desarrollo (1 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Entornos modernos de programación: <ul style="list-style-type: none"> <li>Búsqueda de código.</li> <li>Programación usando librería de componentes y sus APIs.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Construir y depurar programas que utilizan las bibliotecas estándar disponibles con un lenguaje de programación elegido [Familiarizarse]</li> </ul>

**Lecturas :** [Gut13], [Zel10]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### 10. BIBLIOGRAFÍA BÁSICA

- [Zel10] John Zelle. *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates Inc, 2010.
- [Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python*. MIT Press, 2013.
- [BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview*. Ed. by Pearson. Global Edition. Pearson, 2019. URL: <http://www.pearsonhighered.com/brookshear>.

## 1. CURSO

MA101FCCS. Algebra Linear (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	MA101FCCS. Algebra Linear
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	3 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El álgebra lineal es fundamental en la ciencia de la computación, proporcionando herramientas esenciales para el análisis de algoritmos, gráficos por computadora, aprendizaje automático y muchas otras áreas. Este curso proporciona una base sólida en los conceptos y técnicas del álgebra lineal, con un enfoque en su aplicación en la computación.

## 5. OBJETIVOS

- Comprender los conceptos fundamentales del álgebra lineal, incluyendo espacios vectoriales, matrices, transformaciones lineales y sistemas de ecuaciones lineales.
- Aplicar las técnicas del álgebra lineal para resolver problemas en diversos contextos computacionales.
- Desarrollar habilidades de razonamiento abstracto y pensamiento lógico para abordar problemas matemáticos.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Espacios Vectoriales (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C08</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Definición de espacio vectorial y subespacio.</li> <li>Combinaciones lineales, independencia lineal y bases.</li> <li>Dimensión y rango.</li> </ul>	<ul style="list-style-type: none"> <li>Definir y dar ejemplos de espacios vectoriales y subespacios. [Familiarizarse]</li> <li>Determinar la independencia lineal de un conjunto de vectores. [Usar]</li> <li>Calcular bases y la dimensión de un espacio vectorial. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

<b>Unidad 2: Matrices y Sistemas de Ecuaciones Lineales (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Operaciones con matrices.</li> <li>Eliminación gaussiana y forma escalonada reducida.</li> <li>Solución de sistemas de ecuaciones lineales.</li> <li>Matrices inversas y determinantes.</li> </ul>	<ul style="list-style-type: none"> <li>Realizar operaciones con matrices. [Familiarizarse]</li> <li>Resolver sistemas de ecuaciones lineales usando eliminación gaussiana. [Usar]</li> <li>Calcular la inversa de una matriz y su determinante. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

<b>Unidad 3: Transformaciones Lineales (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Definición y ejemplos de transformaciones lineales.</li> <li>Núcleo e imagen de una transformación lineal.</li> <li>Matrices de transformación.</li> </ul>	<ul style="list-style-type: none"> <li>Definir y dar ejemplos de transformaciones lineales. [Familiarizarse]</li> <li>Calcular el núcleo e imagen de una transformación lineal. [Usar]</li> <li>Representar transformaciones lineales mediante matrices. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

<b>Unidad 4: Autovalores y Autovectores (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Definición y cálculo de autovalores y autovectores.</li> <li>Diagonalización de matrices.</li> <li>Aplicaciones de autovalores y autovectores.</li> </ul>	<ul style="list-style-type: none"> <li>Definir y calcular autovalores y autovectores. [Familiarizarse]</li> <li>Diagonalizar matrices. [Usar]</li> <li>Aplicar autovalores y autovectores en la resolución de problemas. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

<b>Unidad 5: Ortogonalidad y Mínimos Cuadrados (8 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Producto interno y ortogonalidad.</li> <li>• Proyecciones ortogonales.</li> <li>• Método de mínimos cuadrados.</li> </ul>	<ul style="list-style-type: none"> <li>• Definir y calcular el producto interno y ortogonalidad. [Familiarizarse]</li> <li>• Calcular proyecciones ortogonales. [Usar]</li> <li>• Aplicar el método de mínimos cuadrados. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

<b>Unidad 6: Aplicaciones en Computación (8 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Aplicaciones en gráficos por computadora.</li> <li>• Aplicaciones en aprendizaje automático.</li> <li>• Aplicaciones en análisis de algoritmos.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir aplicaciones del álgebra lineal en gráficos por computadora. [Familiarizarse]</li> <li>• Explicar cómo se utiliza el álgebra lineal en aprendizaje automático. [Usar]</li> <li>• Analizar la complejidad de algoritmos utilizando conceptos de álgebra lineal. [Evaluar]</li> </ul>

**Lecturas :** [Str16], [LLM16]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[LLM16] David C Lay, Steven R Lay, and Judi J McDonald. *Linear Algebra and Its Applications*. Pearson, 2016.

[Str16] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016.

## 1. CURSO

MA102FCCS. Cálculo Diferencial (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	MA102FCCS. Cálculo Diferencial
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	5
<b>2.4 Horas</b>	:	4 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El cálculo diferencial es una herramienta fundamental en ciencias de la computación para comprender y modelar el cambio. Este curso introduce los conceptos principales del cálculo diferencial, incluyendo límites, derivadas, aplicaciones de la derivada y optimización.

## 5. OBJETIVOS

- Comprender el concepto de límite y su aplicación al cálculo de derivadas.
- Aplicar las reglas de derivación para calcular derivadas de diversas funciones.
- Utilizar la derivada para resolver problemas de optimización, tasas de cambio y análisis de funciones.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Funciones y Límites (6 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Repaso de funciones.</li> <li>• Definición de límite.</li> <li>• Propiedades de los límites.</li> <li>• Límites que involucran infinito.</li> <li>• Continuidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluar límites gráficamente y numéricamente. [Familiarizarse]</li> <li>• Aplicar las propiedades de los límites para evaluar límites algebraicamente. [Usar]</li> <li>• Determinar la continuidad de una función. [Evaluar]</li> </ul>

Lecturas : [Ste15], [LE14]

<b>Unidad 2: La Derivada (6 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Definición de la derivada.</li> <li>• Interpretación geométrica de la derivada.</li> <li>• Derivadas de funciones polinomiales y exponenciales.</li> <li>• Reglas de derivación: suma, producto, cociente y cadena.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular la derivada de una función utilizando la definición. [Familiarizarse]</li> <li>• Interpretar la derivada como la pendiente de la recta tangente. [Usar]</li> <li>• Aplicar las reglas de derivación para encontrar derivadas de funciones. [Evaluar]</li> </ul>

Lecturas : [Ste15], [LE14]

<b>Unidad 3: Aplicaciones de la Derivada (12 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Razones de cambio relacionadas.</li> <li>• Valores máximos y mínimos.</li> <li>• Teorema del valor medio.</li> <li>• Concavidad y puntos de inflexión.</li> <li>• Optimización.</li> </ul>	<ul style="list-style-type: none"> <li>• Resolver problemas de razones de cambio relacionadas. [Familiarizarse]</li> <li>• Encontrar valores máximos y mínimos de una función. [Usar]</li> <li>• Aplicar el teorema del valor medio. [Evaluar]</li> <li>• Determinar la concavidad y los puntos de inflexión de una función. [Evaluar]</li> <li>• Resolver problemas de optimización. [Evaluar]</li> </ul>

Lecturas : [Ste15], [LE14]

<b>Unidad 4: Funciones Trascendentes (12 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Funciones trigonométricas inversas.</li> <li>• Funciones hiperbólicas.</li> <li>• Derivadas de funciones trigonométricas inversas e hiperbólicas.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluar funciones trigonométricas inversas. [Familiarizarse]</li> <li>• Definir y manipular funciones hiperbólicas. [Usar]</li> <li>• Derivar funciones trigonométricas inversas e hiperbólicas. [Evaluar]</li> </ul>

Lecturas : [Ste15], [LE14]

<b>Unidad 5: Aplicaciones en Computación (12 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Optimización de algoritmos.</li> <li>• Modelado de sistemas dinámicos.</li> <li>• Aprendizaje automático (ej. descenso del gradiente).</li> </ul>	<ul style="list-style-type: none"> <li>• Usar derivadas para optimizar algoritmos. [Familiarizarse]</li> <li>• Modelar sistemas dinámicos utilizando ecuaciones diferenciales. [Usar]</li> <li>• Aplicar el cálculo diferencial en algoritmos de aprendizaje automático. [Evaluar]</li> </ul>

Lecturas : [Ste15]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[LE14] Ron Larson and Bruce H. Edwards. *Calculus*. Cengage Learning, 2014.

[Ste15] James Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 2015.

## 1. CURSO

CH101FCCS. Química I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CH101FCCS. Química I
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	5
<b>2.4 Horas</b>	:	4 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La química proporciona una base para comprender la composición, estructura y propiedades de la materia. Aunque no está directamente relacionada con muchos aspectos de la programación, la química es importante para la ciencia de la computación en áreas como la ciencia de materiales (desarrollo de nuevos materiales para componentes informáticos), la nanotecnología y la bioinformática. Este curso introduce los principios básicos de la química general.

## 5. OBJETIVOS

- Comprender la estructura de la materia a nivel atómico y molecular.
- Aplicar los principios de la estequiometría para realizar cálculos químicos.
- Comprender los diferentes tipos de enlaces químicos y su influencia en las propiedades de las sustancias.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

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

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

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

## 7. TEMAS

<b>Unidad 1: Introducción a la Química (6 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Materia y energía.</li> <li>• El método científico.</li> <li>• Unidades de medida.</li> <li>• Clasificación de la materia.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las propiedades de la materia y la energía. [Familiarizarse]</li> <li>• Aplicar el método científico para resolver problemas químicos. [Usar]</li> <li>• Realizar conversiones de unidades. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17], [CG16]

<b>Unidad 2: Estructura Atómica (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Estructura del átomo.</li> <li>• Número atómico, número másico e isótopos.</li> <li>• Modelo cuántico del átomo.</li> <li>• Configuración electrónica.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la estructura del átomo. [Familiarizarse]</li> <li>• Determinar la configuración electrónica de un átomo. [Usar]</li> <li>• Relacionar la configuración electrónica con las propiedades químicas. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17], [CG16]

<b>Unidad 3: Enlaces Químicos (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Enlace iónico.</li> <li>• Enlace covalente.</li> <li>• Enlace metálico.</li> <li>• Geometría molecular.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir los diferentes tipos de enlaces químicos. [Familiarizarse]</li> <li>• Predecir la geometría molecular de una molécula. [Usar]</li> <li>• Relacionar el tipo de enlace con las propiedades de las sustancias. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17], [CG16]

<b>Unidad 4: Estequiometría (8 horas)</b> <b>Resultados esperados: 1,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Masa molar y mol.</li> <li>• Reacciones químicas y ecuaciones químicas.</li> <li>• Cálculos estequiométricos.</li> <li>• Reactivo limitante y rendimiento porcentual.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular la masa molar de un compuesto. [Familiarizarse]</li> <li>• Balancear ecuaciones químicas. [Usar]</li> <li>• Realizar cálculos estequiométricos para determinar la cantidad de reactivos y productos. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17], [CG16]

<b>Unidad 5: Estados de la Materia (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Gases.</li> <li>• Líquidos.</li> <li>• Sólidos.</li> <li>• Cambios de estado.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las propiedades de los diferentes estados de la materia. [Familiarizarse]</li> <li>• Explicar los cambios de estado y los diagramas de fase. [Usar]</li> <li>• Aplicar las leyes de los gases ideales. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17], [CG16]

<b>Unidad 6: Química y Computación (10 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Modelado molecular.</li> <li>• Simulaciones químicas.</li> <li>• Ciencia de materiales en computación.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir cómo se utiliza la química en el modelado molecular. [Familiarizarse]</li> <li>• Explicar el papel de la química en las simulaciones químicas. [Usar]</li> <li>• Analizar la importancia de la química en la ciencia de materiales para la computación. [Evaluar]</li> </ul>

**Lecturas :** [Bro+17]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[CG16] Raymond Chang and Kenneth A. Goldsby. *Chemistry*. McGraw-Hill Education, 2016.

[Bro+17] Theodore L. Brown et al. *Chemistry: The Central Science*. Pearson, 2017.

## 1. CURSO

FG001FCCS. Electivo Formación General (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FG001FCCS. Electivo Formación General
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Electivo Formación General

## 5. OBJETIVOS

- cursos de Formación General Electivos.

## 6. RESULTADOS DEL ESTUDIANTE

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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)

3) Comunicarse efectivamente en diversos contextos profesionales. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Primera Unidad (48 horas)</b>	
<b>Resultados esperados:</b>	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Cursos Electivo de Formacion General.</li></ul>	<ul style="list-style-type: none"><li>• Desarrollar habilidades y conocimiento de otros cursos fuera de su Formación técnica de la carrera [Usar].</li></ul>
<b>Lecturas :</b> [Rob04]	

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## **10. BIBLIOGRAFÍA BÁSICA**

[Rob04] Stephen Robbins. *Comportamiento Organizacional*. México, Pearson Educación, 2004.

## 1. CURSO

ID101. Inglés I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	ID101. Inglés I
<b>2.2 Semestre</b>	:	1 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso aborda aspectos y técnicas fundamentales de la investigación académica y la redacción con el fin de proporcionar al estudiante principiante e intermedio una base sólida para trabajar en trabajos de ensayo, trabajos finales y proyectos de investigación de pregrado. El objetivo es ofrecer una introducción fácilmente aplicable, pero teóricamente profunda, en el campo de la investigación académica y la escritura, que se puede entender sin literatura adicional.

## 5. OBJETIVOS

- Conocer el idioma Inglés y su estructura gramatical.
- Identificar situaciones y emplear diálogos relacionados a ellas.

## 6. RESULTADOS DEL ESTUDIANTE

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

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

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

## 7. TEMAS

<b>Unidad 1: Planificación y preparación (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Analiza los beneficios para usted de publicar su investigación y sugiere varios enfoques</li> </ul>	<ul style="list-style-type: none"> <li>Elegir la revista adecuada y comprender lo que el editor espera de un artículo en términos de contenido estilo y estructura.</li> <li>Decidir el orden en el que escribir las distintas secciones.</li> <li>Mantener contentos a los Árbitros.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 2: Estructurar una oración: orden de las palabras (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Proporciona reglas para decidir donde colocar varios tipos de palabras dentro de una oración.</li> </ul>	<ul style="list-style-type: none"> <li>Use oraciones cortas, no tome este concepto demasiado literal.</li> <li>Comprender el orden de las palabras.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 3: Estructura de Párrafos (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Cubre como estructurar un párrafo uniendo oraciones en el orden más lógico posible. También sugiere formas de dividir párrafos largos.</li> </ul>	<ul style="list-style-type: none"> <li>Comprender ?'Cómo puedo facilitar que la lectura siga lo que estoy diciendo y comprender claramente los beneficios de mis métodos y hallazgos?</li> <li>Comprender ¿Cómo lograr el mínimo esfuerzo posible de mis lectores?</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 4: Romper oraciones largas (0 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Explique por qué y cuanto tiempo se crean las oraciones, los pros y contras de usar oraciones cortas para sus lectores y los beneficios de usar oraciones cortas para sus lectores y coautores</li> </ul>	<ul style="list-style-type: none"> <li>Comprender como convertir una oración larga en oraciones cortas.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 5: Ser conciso y eliminar la redundancia (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Comienza dándote buenas razones para evitar la redundancia y luego te muestra como ser conciso</li> </ul>	<ul style="list-style-type: none"> <li>No crea que el uso de términos complejos le hará parecer más inteligente</li> <li>Escribe usando la terminología más simple y directiva.</li> <li>Elimine todo lo que no sea esencial: esto permitirá que sus ideas clave se destaque (se vean) más fácilmente.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

<b>Unidad 6: Evitar la ambigüedad, la repetición y el lenguaje vago (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Explique que es un Abstract, que tiempos y estilo elegir y cuáles son los distintos tipos de Abstracts.</li> </ul>	<ul style="list-style-type: none"> <li>Entender como comenzar un resumen.</li> <li>Entender que incluir y como hacer que aumente su impacto.</li> <li>Identificar las trampas típicas de los resúmenes mal escritos.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

<b>Unidad 7: Introducción (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Diferenciar la Introducción del Resumen.</li> <li>Estructura y comienza una Introducción.</li> <li>Evite las trampas típicas.</li> <li>Describe la estructura del resto del artículo.</li> </ul>	<ul style="list-style-type: none"> <li>Uso de los tiempos verbales apropiados.</li> <li>Comprender ¿Cómo puede evaluar la calidad de mi introducción?</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

<b>Unidad 8: Revisión de la literatura (0 horas)</b> <b>Resultados esperados: 3,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Decidir que literatura mencionar..</li> <li>Estructura tu reseña.</li> </ul>	<ul style="list-style-type: none"> <li>Diferenciar su trabajo de la literatura.</li> <li>Resaltar el nivel de innovación en su trabajo junto con sus limitaciones.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

<b>Unidad 9: Métodos (0 horas)</b> <b>Resultados esperados: 3,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Describe los materiales que usaste en tus experimentos y/o los métodos que usaste para llevar a cabo tu investigación, de una manera que sea lo suficientemente detallada para permitir que otros en tu campo sigan fácilmente tu método y si lo deseas, incluso reproduzcan tu trabajo.</li> </ul>		<ul style="list-style-type: none"> <li>Asegúrese de que las descripciones están completas ,también sean lo más concisas posible.</li> <li>Permita que sus lectores sigan fácilmente su descripción.</li> <li>Escribir con extrema claridad, generalmente con no más de dos pasos descritos en una oración y en un orden lógico.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]		

<b>Unidad 10: Resultados (0 horas)</b> <b>Resultados esperados: 3,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
		<ul style="list-style-type: none"> <li>Organice los resultados en una secuencia que resalte las respuestas a los objetivos, hipótesis o preguntas que se plantea al principio del artículo.</li> <li>Informe sus resultados de forma simple y clara.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]		

<b>Unidad 11: Discusión (0 horas)</b> <b>Resultados esperados: 3,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
		<ul style="list-style-type: none"> <li>Estructura de la discusión.</li> <li>Suena convincente y creíble al mismo tiempo.</li> <li>Interprete sus resultados sin repetirlos.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]		

<b>Unidad 12: Conclusiones (0 horas)</b> <b>Resultados esperados: 3,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
		<ul style="list-style-type: none"> <li>Comprender las habilidades clave para saber lo que los árbitros y lectores esperan encontrar en conclusiones.</li> <li>Comprender que no debe repetir exactamente las mismas frases e información de su resumen e introducción y al proporcionar un mensaje claro y de gran impacto para los lectores.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]		

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### **10. BIBLIOGRAFÍA BÁSICA**

- [Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
- [SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.
- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.

## **Capítulo 2**

### **Segundo Semestre**

## 1. CURSO

CS100-CS2023. Introducción a la Ciencia de la Computación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS100-CS2023. Introducción a la Ciencia de la Computación
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso es la base para entender los conceptos fundamentales de pensamiento computacional transversales a cualquier profesión.

El curso presenta, desde un nivel cero, una visión panorámica de: introductoria al pensamiento computacional, almacenamiento de datos, arquitectura de computadores, sistemas operativos, redes e Internet, algoritmos, métodos de ordenamiento, ingeniería de software, bases de datos, estructuras de datos, ingeniería de software, computación gráfica, inteligencia artificial.

Debido a que está diseñado como un curso introductorio a la Ciencia de la Computación, los conceptos son presentados de forma lúdica y utilizando una metodología de Aprendizaje Activo (*Active Learning*). Durante el dictado del curso, se busca siempre una participación activa de la audiencia al estilo de una obra de teatro.

Las áreas de conocimiento relacionadas que se tocan están directamente relacionadas a la Ciencia de la Computación de acuerdo a la *Computing Curricula ACM/IEEE-CS*.

El curso **no requiere** ningún tipo de conocimiento previo en temas de manejo de computadores y puede ser tomado por alumnos de cualquier carrera.

## 5. OBJETIVOS

- Introducir los conceptos fundamentales de Pensamiento Computacional y Ciencia de la Computación a estudiantes de cualquier carrera profesional.
- Desarrollar su capacidad de abstracción.
- Entender como el Pensamiento Computacional se aplica en cada una de sus profesiones.
- Aplicar conceptos avanzados de Ciencia de la Computación de una manera simple en cualquier carrera.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

**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)

## 7. TEMAS

<b>Unidad 1: Pensamiento Computacional. Parte I (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Indicaciones generales del curso.</li> <li>Explicación del sistema de evaluación.</li> <li>Definición de Computación.</li> <li>Computación como un binomio Humano-Computador.</li> <li>Distorsiones en la definición de computación.</li> <li>Computación como la automatización de una abstracción.</li> <li>Computación e Ingeniería: similitudes y diferencias.</li> <li>Resolución de problemas algorítmicos.</li> <li>Dinámica: Entendiendo la ejecución de un algoritmo a velocidad humana.</li> </ul>	<ul style="list-style-type: none"> <li>Aplicar los conceptos fundamentales de computación en situaciones reales. [Usar]</li> <li>Identificar las distorsiones de la Computación en situaciones reales. [Usar]</li> <li>Identificar con claridad al menos 3 contextos de uso de la palabra <i>Engineer</i> en inglés. [Evaluar]</li> <li>Identificar las limitaciones del ser humano para resolver problemas computacionales. [Usar]</li> </ul>
<b>Lecturas :</b> [BB19]	

<b>Unidad 2: Pensamiento Computacional. Parte II (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Numeración binaria vs decimal.</li> <li>Representación de caracteres: la tabla ASCII.</li> <li>Representación interna de colores.</li> <li>Entendiendo una imagen desde adentro.</li> <li>Búsqueda binaria.</li> <li>Complejidad computacional de un algoritmo.</li> </ul>	<ul style="list-style-type: none"> <li>Aplicar diversos sistemas de numeración para problemas reales. [Usar]</li> <li>Entender la representación interna de los caracteres en la tabla ASCII y en UTF-8. [Familiarizarse]</li> <li>Entender la representación de los colores en una imagen. [Familiarizarse]</li> <li>Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima. [Usar]</li> <li>Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos. [Usar]</li> </ul>
<b>Lecturas :</b> [BB19]	

**Unidad 3: Lógica digital y sistemas digitales (4 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Revisión e historia de la Arquitectura de Computadores.</li> <li>Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial.</li> <li>Multiples representaciones / Capas de interpretación (El hardware es solo otra capa)</li> </ul>	<ul style="list-style-type: none"> <li>Describir el avance paulatino de los componentes de la tecnología de computación, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse]</li> <li>Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Familiarizarse]</li> <li>Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Familiarizarse]</li> <li>Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarizarse]</li> <li>Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usar]</li> </ul>

**Lecturas :** [BB19]

**Unidad 4: Representación de programas (2 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Programas que tienen otros programas como entrada tales como interpretadores, compiladores, revisores de tipos y generadores de documentación.</li> <li>Arboles de sintaxis abstracta, para contrastar la sintaxis correcta.</li> <li>Estructuras de datos que representan código para ejecución, traducción o transmisión.</li> </ul>	<ul style="list-style-type: none"> <li>Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse]</li> <li>Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Usar]</li> <li>Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse]</li> <li>Escribir un programa para procesar alguna representación de código para algún propósito, tales como un interprete, una expresión optimizada, o un generador de documentación [Usar]</li> </ul>

**Lecturas :** [BB19]

<b>Unidad 5: Criptografía (2 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Terminología básica de criptografía cubriendo las nociones relacionadas con los diferentes socios (comunicación), canal seguro / inseguro, los atacantes y sus capacidades, cifrado, descifrado, llaves y sus características, firmas.</li> <li>Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos.</li> </ul>	<ul style="list-style-type: none"> <li>Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse]</li> <li>Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [BB19]	

<b>Unidad 6: Organización y Arquitectura del Sistema de Memoria (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Sistemas de Almacenamiento y su Tecnología.</li> <li>Jerarquía de Memoria: importancia de la localización temporal y espacial.</li> <li>Organización y Operaciones de la Memoria Principal.</li> <li>Latencia, ciclos de tiempo, ancho de banda e intercalación.</li> <li>Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento)</li> <li>Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core.</li> <li>Memoria virtual (tabla de página, TLB)</li> <li>Manejo de Errores y confiabilidad.</li> <li>Error de codificación, compresión de datos y la integridad de datos.</li> </ul>	<ul style="list-style-type: none"> <li>Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse]</li> <li>Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse]</li> <li>Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse]</li> <li>Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse]</li> <li>Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [BB19]	

<b>Unidad 7: Visión general de Sistemas Operativos (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Papel y el propósito del sistema operativo.</li> <li>• Funcionalidad de un sistema operativo típico.</li> <li>• Los mecanismos de apoyo modelos cliente-servidor, dispositivos de mano.</li> <li>• Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad)</li> <li>• Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse]</li> <li>• Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Usar]</li> <li>• Describir las funciones de un sistema operativo contemporaneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse]</li> <li>• Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse]</li> <li>• Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]</li> </ul>
<b>Lecturas : [BB19]</b>	

<b>Unidad 8: Introducción a redes (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc)</li> <li>• Técnicas de Switching (por ejemplo, de circuitos, de paquetes)</li> <li>• Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls.</li> <li>• Principios de capas (encapsulación, multiplexación)</li> <li>• Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)</li> </ul>	<ul style="list-style-type: none"> <li>• Articular la organización de la Internet [Familiarizarse]</li> <li>• Listar y definir la terminología de red apropiada [Familiarizarse]</li> <li>• Describir la estructura en capas de una arquitectura típica en red [Familiarizarse]</li> <li>• Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]</li> </ul>
<b>Lecturas : [BB19]</b>	

<b>Unidad 9: Entrega confiable de datos (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Control de errores (técnicas de retransmisión, temporizadores)</li> <li>• El control de flujo (agradecimientos, ventana deslizante)</li> <li>• Problemas de rendimiento (pipelining)</li> <li>• TCP</li> </ul>	<ul style="list-style-type: none"> <li>• Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse]</li> <li>• Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse]</li> <li>• Diseñar e implementar un protocolo confiable simple [Usar]</li> </ul>
<b>Lecturas : [BB19]</b>	

**Unidad 10: Análisis Básico (4 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.</li> <li>• Análisis asintótico de complejidad de cotas superior y esperada.</li> <li>• Definición formal de la Notación Big O.</li> <li>• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.</li> <li>• Medidas empíricas de desempeño.</li> <li>• Compensación entre espacio y tiempo en los algoritmos.</li> <li>• Uso de la notación Big O.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse]</li> <li>• En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar]</li> <li>• Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Usar]</li> <li>• Indique la definición formal de Big O [Familiarizarse]</li> <li>• Lista y contraste de clases estándares de complejidad [Familiarizarse]</li> <li>• Realizar estudios empíricos para validar una hipótesis sobre runtime stemming desde un análisis matemático Ejecute algoritmos con entrada de varios tamaños y compare el desempeño [Evaluar]</li> <li>• Da ejemplos que ilustran las compensaciones entre espacio y tiempo que se dan en los algoritmos [Familiarizarse]</li> <li>• Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar]</li> <li>• Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Familiarizarse]</li> </ul>

**Lecturas :** [BB19]

**Unidad 11: Algoritmos y Estructuras de Datos fundamentales (8 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo.</li> <li>• Algoritmos de búsqueda secuencial y binaria.</li> <li>• Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción)</li> <li>• Algoritmos de ordenamiento con peor caso o caso promedio en <math>O(N \lg N)</math> (Quicksort, Heapsort, Mergesort)</li> <li>• Tablas Hash, incluyendo estrategias para evitar y resolver colisiones.</li> <li>• Árboles de búsqueda binaria: <ul style="list-style-type: none"> <li>– Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles.</li> </ul> </li> <li>• Grafos y algoritmos en grafos: <ul style="list-style-type: none"> <li>– Representación de grafos (ej., lista de adyacencia, matriz de adyacencia)</li> <li>– Recorrido en profundidad y amplitud</li> </ul> </li> <li>• Montículos (Heaps)</li> <li>• Grafos y algoritmos en grafos: <ul style="list-style-type: none"> <li>– Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd)</li> <li>– Árbol de expansión mínima (algoritmos de Prim y Kruskal)</li> </ul> </li> <li>• Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse]</li> <li>• Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar]</li> <li>• Implementar algoritmos numéricos básicos [Usar]</li> <li>• Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar]</li> <li>• Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y <math>O(N \log N)</math> [Usar]</li> <li>• Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse]</li> <li>• Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse]</li> <li>• Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse]</li> <li>• Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse]</li> <li>• Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar]</li> <li>• Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto específico [Evaluar]</li> </ul>

**Lecturas :** [BB19]

**Unidad 12: Sistemas de Bases de Datos (4 horas)****Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Enfoque y Evolución de Sistemas de Bases de Datos.</li><li>• Componentes del Sistema de Bases de Datos.</li><li>• Diseño de las funciones principales de un DBMS.</li><li>• Arquitectura de base de datos e independencia de datos.</li><li>• Uso de un lenguaje de consulta declarativa.</li><li>• Sistemas de apoyo a contenido estructurado y / o corriente.</li><li>• Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce).</li></ul>	<ul style="list-style-type: none"><li>• Describe los enfoques principales para almacenar y procesar grandes volúmenes de datos [Familiarizarse]</li><li>• Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse]</li><li>• Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse]</li><li>• Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Familiarizarse]</li><li>• Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse]</li><li>• Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Familiarizarse]</li><li>• Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse]</li><li>• Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar]</li><li>• Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Familiarizarse]</li></ul>

**Lecturas : [BB19]**

**Unidad 13: Programación orientada a objetos (4 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● Diseño orientado a objetos: <ul style="list-style-type: none"> <li>– Descomposición en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquía de clases para modelamiento</li> </ul> </li> <li>● Definición de las categorías, campos, métodos y constructores.</li> <li>● Las subclases, herencia y método de alteración temporal.</li> <li>● Asignación dinámica: definición de método de llamada.</li> <li>● Subtipificación: <ul style="list-style-type: none"> <li>– Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos.</li> <li>– Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos.</li> <li>– Relación entre subtipos y la herencia.</li> </ul> </li> <li>● Lenguajes orientados a objetos para la encapsulación: <ul style="list-style-type: none"> <li>– privacidad y la visibilidad de miembros de la clase</li> <li>– Interfaces revelan único método de firmas</li> <li>– clases base abstractas</li> </ul> </li> <li>● Uso de colección de clases, iteradores, y otros componentes de la librería estandar.</li> </ul>	<ul style="list-style-type: none"> <li>● Diseñar e implementar una clase [Usar]</li> <li>● Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar]</li> <li>● Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar]</li> <li>● Comparar y contrastar (1) el enfoque procedural/funcional- definiendo una función por cada operación con el cuadro de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Evaluar]</li> <li>● Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse]</li> <li>● Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar]</li> <li>● Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma mas natural por cada lenguaje [Usar]</li> </ul>

**Lecturas :** [BB19]

<b>Unidad 14: Procesos de Software (4 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.</li> <li>• Introducción a modelos del proceso de software (e.g., cascada, incremental, agil): <ul style="list-style-type: none"> <li>– Actividades con ciclos de vida de software.</li> </ul> </li> <li>• Programación a gran escala versus programación individual.</li> <li>• Evaluación de modelos de proceso de software.</li> <li>• Conceptos de calidad de software.</li> <li>• Mejoramiento de procesos.</li> <li>• Modelos de madurez de procesos de software.</li> <li>• Mediciones del proceso de software.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Familiarizarse]</li> <li>• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Familiarizarse]</li> <li>• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Familiarizarse]</li> <li>• Diferenciar entre las fases de desarrollo de software [Familiarizarse]</li> <li>• Describir cómo la programación en grandes equipos difiere de esfuerzos individuales con respecto a la comprensión de una gran base de código, lectura de código, comprensión de las construcciones, y comprensión de contexto de cambios [Familiarizarse]</li> <li>• Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Familiarizarse]</li> <li>• Comparar varios modelos comunes de procesos con respecto a su valor para el desarrollo de las clases particulares de sistemas de software, teniendo en cuenta diferentes aspectos tales como, estabilidad de los requisitos, tamaño y características no funcionales [Usar]</li> <li>• Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Familiarizarse]</li> <li>• Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [BB19]	

<b>Unidad 15: Cuestiones fundamentales (2 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial.</li> <li>• ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> <li>– El Test de Turing</li> <li>– Razonamiento Racional versus No Racional</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Describir el test de Turing y el experimento pensado cuarto chino” (<i>Chinese Room</i>) [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [BB19]	

**Unidad 16: Estrategias de búsquedas básicas (1 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda.</li> <li>• Factored representation (factoring state hacia variables)</li> <li>• Uninformed search (breadth-first, depth-first, depth-first with iterative deepening)</li> <li>• Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*)</li> <li>• El espacio y el tiempo de la eficiencia de búsqueda.</li> <li>• Dos jugadores juegos (introducción a la búsqueda minimax).</li> <li>• Satisfacción de restricciones (backtracking y métodos de búsqueda local).</li> </ul>	<ul style="list-style-type: none"> <li>• Formula el espacio eficiente de un problema para un caso expresado en lenguaje natural (ejm. Inglés) en términos de estados de inicio y final, así como sus operadores [Usar]</li> <li>• Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Familiarizarse]</li> <li>• Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Familiarizarse]</li> <li>• Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar]</li> <li>• Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar]</li> <li>• Evalúa si una heurística dada para un determinado problema es admisible/puede garantizar una solución óptima [Evaluar]</li> <li>• Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Familiarizarse]</li> </ul>

**Lecturas :** [BB19]

**Unidad 17: Aprendizaje Automático Básico (1 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación.</li> <li>• Aprendizaje inductivo</li> <li>• Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión.</li> <li>• El problema exceso de ajuste.</li> <li>• Medición clasificada con exactitud.</li> </ul>	<ul style="list-style-type: none"> <li>• Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Familiarizarse]</li> <li>• Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Familiarizarse]</li> <li>• Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Familiarizarse]</li> </ul>

**Lecturas :** [BB19]

**Unidad 18: Conceptos Fundamentales (2 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual.</li> <li>Digitalización de datos analógicos, la resolución y los límites de la percepción humana, por ejemplo, los píxeles de la pantalla visual, puntos para impresoras láser y muestras de audio</li> <li>El uso de las API estándar para la construcción de interfaces de usuario y visualización de formatos multimedia estándar</li> <li>Formatos estándar, incluyendo formatos sin pérdidas y con pérdidas.</li> <li>Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores.</li> <li>Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes.</li> <li>Animación como una secuencia de imágenes fijas.</li> <li>Almacenamiento doble.</li> </ul>	<ul style="list-style-type: none"> <li>Identificar usos comunes de presentaciones digitales de humanos (por ejemplo, computación gráfica, sonido) [Familiarizarse]</li> <li>Explicar en términos generales cómo las señales analógicas pueden ser representadas por muestras discretas, por ejemplo, cómo las imágenes pueden ser representadas por pixeles [Familiarizarse]</li> <li>Explicar cómo las limitaciones en la percepción humana afectan la selección de la representación digital de señales analógicas [Usar]</li> <li>Describir las diferencias entre técnicas de compresión de imágenes con pérdida y sin pérdida exemplificando cómo se reflejan en formatos de archivos de imágenes conocidos como JPG, PNG, MP3, MP4, y GIF [Familiarizarse]</li> <li>Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse]</li> <li>Describir las ventajas y desventajas entre el almacenamiento de información vs almacenar suficiente información para reproducir la información, como en la diferencia entre el vector y la representación de la trama [Familiarizarse]</li> <li>Describir los procesos básicos de la producción de movimiento continuo a partir de una secuencia de cuadros discretos (algunas veces llamado it flicker fusion) [Familiarizarse]</li> <li>Describir cómo el doble buffer puede eliminar el parpadeo de la animación [Familiarizarse]</li> </ul>

**Lecturas :** [BB19]

**Unidad 19: Rendering Básico (2 horas)**

**Resultados esperados: 1,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica.</li> <li>• Renderizado Fordward and Backward (i.e., <i>ray-casting</i> y rasterización)</li> <li>• Representación poligonal</li> <li>• Radiometría básica, triángulos similares y modelos de proyecciones</li> <li>• Afinamiento y Transformaciones de Sistemas de coordenadas</li> <li>• <i>Ray tracing</i></li> <li>• Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos.</li> <li>• Representación de la ecuación de adelante hacia atrás.</li> <li>• Rasterización triangular simple.</li> <li>• Mapeo de texturas, incluyendo minificación y magnificación (e.g., MIP-mapping trilineal)</li> <li>• Aplicación de la representación de estructuras de datos espaciales.</li> <li>• Muestreo y anti-aliasing.</li> <li>• Gráficos en escena y la canalización de gráficos.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el problema de transporte de la luz y su relación con la integración numérica, es decir, se emite luz, dispersa alrededor de la escena, y es medida por el ojo [Familiarizarse]</li> <li>• Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse]</li> <li>• Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar]</li> <li>• Derivar la perspectiva lineal de triángulos semejantes por conversión de puntos <math>(x,y,z)</math> a puntos <math>(x/z, y/z, 1)</math> [Usar]</li> <li>• Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afín [Usar]</li> <li>• Aplicar sistema de coordenadas de 3-dimensiones y los cambios necesarios para extender las operaciones de transformación 2D para manejar las transformaciones en 3D [Usar]</li> <li>• Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse]</li> <li>• Implementar simples procedimientos que realicen la transformación y las operaciones de recorte de imágenes simples en 2 dimensiones [Usar]</li> <li>• Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar]</li> <li>• Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]</li> </ul>

**Lecturas :** [BB19]

<b>Unidad 20: Clase de cierre: ¿Cómo funciona un buscador como Google? (2 horas)</b> <b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Análisis del problema</li> <li>• El índice no crece de forma lineal con relación al tamaño de la información indexada.</li> <li>• El tiempo de respuesta no depende del tamaño de la “base de datos”.</li> <li>• El tiempo de respuesta no depende del número de ocurrencias encontradas.</li> <li>• Combinando diversas estructuras de datos para llegar a la solución.</li> <li>• Analizando la escalabilidad de la solución.</li> </ul>	<ul style="list-style-type: none"> <li>• Entender los principios bajo los cuales se crea un motor de búsqueda [Usar]</li> <li>• Aplicar correctamente estructuras de datos para dar solución al problema [Usar]</li> <li>• Aplicar los conceptos relacionados a complejidad algorítmica en motor de búsqueda [Usar].</li> </ul>

**Lecturas :** [BB19]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview*. Ed. by PEARSON. Global Edition. Pearson, 2019. URL: <http://www.pearsonhighered.com/brookshear>.

## 1. CURSO

CS112-CS2023. Programación Orientada a Objetos I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS112-CS2023. Programación Orientada a Objetos I
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS111. Introducción a la Programación. (1 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este es el segundo curso en la secuencia de los cursos introductorios a la Ciencia de la Computación en la linea de programación. Ademas este curso incluye un cambio de Lenguaje de Programacion como estrategia para aprendizaje. En este curso se abordan los conceptos fundamentales de Programación Orientada a Objetos profundizando en los conceptos aprendidos en el primer curso tratando de hacer especial enfasis en conocer más conceptos de bajo nivel. Este curso debe dar una base solida para abordar conceptos avanzados en el siguiente curso de esta secuencia.

## 5. OBJETIVOS

- Introducir al alumno a los fundamentos del paradigma de orientación a objetos, permitiendo asimilar los conceptos necesarios para desarrollar sistemas de información.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Visión general de los lenguajes de programación (1 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Breve repaso de los paradigmas de programación.</li> <li>• Comparación entre programación funcional y programación imperativa.</li> <li>• Historia de los lenguajes de programación (énfasis en C y C++).</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Str13], [Jos19], [Dei17]	

<b>Unidad 2: Máquinas virtuales (2 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• El concepto de máquina virtual.</li> <li>• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) .</li> <li>• Lenguajes intermedios.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse]</li> <li>• Diferenciar emulación y el aislamiento [Familiarizarse]</li> <li>• Evaluar virtualización de compensaciones [Evaluar]</li> </ul>
<b>Lecturas :</b> [Str13], [Jos19], [Dei17]	

Unidad 3: Sistemas de tipos básicos (6 horas)	
Resultados esperados: 6	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Sistemas de tipos en lenguajes de programación.</li> <li>• Modelos de declaración (enlace, visibilidad, alcance y tiempo de vida).</li> <li>• Resumen de la verificación de tipos.</li> </ul>	<ul style="list-style-type: none"> <li>• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse]</li> <li>• Para un lenguaje con sistema de tipos estático, describir las operaciones que están prohibidas de forma estática, como pasar el tipo incorrecto de valor a una función o método [Familiarizarse]</li> <li>• Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse]</li> <li>• Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar]</li> <li>• Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Familiarizarse]</li> <li>• Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar]</li> <li>• Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Familiarizarse]</li> <li>• Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar]</li> <li>• Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Familiarizarse]</li> <li>• Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]</li> <li>• Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Familiarizarse]</li> <li>• Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Familiarizarse]</li> </ul>

Lecturas : [Str13], [Jos19], [Dei17]

<b>Unidad 4: Conceptos Fundamentales de Programación (6 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Diseño orientado a objetos: <ul style="list-style-type: none"> <li>– Descomposición en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquía de clases para modelamiento</li> </ul> </li> <li>• Variables y tipos de datos.</li> <li>• Expresiones y operadores.</li> <li>• Sentencias condicionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Analiza y explica el comportamiento de programas simples que involucran estructuras fundamentales de programación variables, expresiones, asignaciones, E/S, estructuras de control, funciones, paso de parámetros, y recursividad [Evaluar]</li> <li>• Identifica y describe el uso de tipos de datos primitivos [Familiarizarse]</li> <li>• Escribe programas que usan tipos de datos primitivos [Usar]</li> <li>• Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar]</li> <li>• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]</li> <li>• Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Evaluar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

<b>Unidad 5: Funciones (6 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Paso de funciones y parámetros.</li> <li>• Paso de parámetros.</li> <li>• Sobrecarga de funciones.</li> <li>• Fundamentos de la recursividad.</li> <li>• Plantillas de funciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Diseña, implementa, prueba, y depura un programa que usa cada una de las siguientes estructuras de datos fundamentales: cálculos básicos, E/S simple, condicional estándar y estructuras iterativas, definición de funciones, y paso de parámetros [Usar]</li> <li>• Entiende y aplica el concepto de paso de parámetros a una función, tanto por valor como por referencia.[Usar]</li> <li>• Identifica y aplica el concepto de sobrecarga de funciones.[Usar]</li> <li>• Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse]</li> <li>• Diseña, implementa y aplica el concepto de plantillas asociado a la necesidad de crear funciones genéricas.[Usar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

**Unidad 6: Arreglos, punteros y gestión de memoria (8 horas)**

**Resultados esperados: 6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Definición de arreglos.</li> <li>Arreglos multidimensionales.</li> <li>Fundamentos de punteros.</li> <li>Gestión de memoria dinámica (new/delete, pila vs. heap).</li> <li>Punteros inteligentes (unique_ptr, shared_ptr, weak_ptr).</li> <li>Conceptos avanzados de punteros (punteros a punteros, punteros a funciones).</li> </ul>	<ul style="list-style-type: none"> <li>Entiende e implementa arreglos unidimensionales. [Familiarizarse]</li> <li>Diseña y aplica el concepto de arreglos multidimensionales. [Usar]</li> <li>Entiende y aplica el concepto de referencias y punteros. [Familiarizarse]</li> <li>Entiende, aplica y evalua la relación entre punteros y arreglos. [Evaluar]</li> <li>Entiende e implementa la gestión dinámica de la memoria. Diferenciando las regiones de memoria: heap y stack. [Evaluar]</li> <li>Diseña, implementa y evalua el concepto de puntero a puntero, puntero a función, entre otros conceptos. [Evaluar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

**Unidad 7: Manejo de punteros con arrays (5 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Arrays como argumentos de una función.</li> <li>Arrays de caracteres y punteros.</li> <li>Punteros y Arrays de 2 dimensiones.</li> <li>Punteros y arrays multidimensionales.</li> </ul>	<ul style="list-style-type: none"> <li>Demostrar el uso de punteros con diferentes tipos de Arrays. [Usar]</li> <li>Demostrar la disposición de un array en la memoria y como se manipula punteros dentro de esos espacios de memoria. [Usar]</li> <li>Demostrar el uso de aritmética de punteros y arrays. [Usar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

**Unidad 8: Punteros y memoria dinámica (5 horas)**
**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Punteros y memoria dinámica - stack vs heap.</li> <li>• Punteros como retorno de una función en C/C++.</li> <li>• Punteros a funciones en C/C++.</li> <li>• Punteros a funciones y callback.</li> <li>• Memory leak en C/C++.</li> </ul>	<ul style="list-style-type: none"> <li>• Mostrar la estructura de la memoria dentro de un programa y comprender cómo es que el compilador dispone elementos en el stack y en el heap.[Usar]</li> <li>• Demostrar el uso de las funciones y operadores de asignación y desasignación de memoria dinámica.[Usar]</li> <li>• Comprender las implicancias de retornar punteros desde funciones. [Usar]</li> <li>• Utilizar punteros a funciones como parámetros. [Usar]</li> <li>• Comprender la implicancia de uso de memoria dinámica y el memory leak. [Usar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

**Unidad 9: Punteros y clases (5 horas)**
**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Punteros a miembros clase - atributos.</li> <li>• Punteros a miembros clase - métodos y llamadas a punteros a métodos.</li> <li>• Punteros a miembros clase - métodos static y llamadas a punteros a métodos static.</li> <li>• Punteros a clases - ejemplo con manejo de lista enlazada.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender el uso de punteros a diferentes elementos de una clase. [Usar]</li> <li>• Comprender el uso de punteros a miembros estáticos de una clase. [Usar]</li> <li>• Introducir en la estructura nodo y su uso en una estructura de datos simple. [Usar]</li> <li>• Introducir a las estructuras de datos, mostrando una implementación simple de listas enlazadas.[Usar]</li> </ul>

**Lecturas :** [Str13], [Jos19], [Dei17]

## Unidad 10: Programación orientada a objetos (8 horas)

Resultados esperados: 6

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Diseño orientado a objetos: <ul style="list-style-type: none"> <li>– Descomposición en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquía de clases para modelamiento</li> </ul> </li> <li>• Lenguajes orientados a objetos para la encapsulación: <ul style="list-style-type: none"> <li>– privacidad y la visibilidad de miembros de la clase</li> <li>– Interfaces revelan único método de firmas</li> <li>– clases base abstractas</li> </ul> </li> <li>• Definición de las categorías, campos, métodos y constructores.</li> <li>• Las subclases, herencia y método de alteración temporal.</li> <li>• Subtipificación: <ul style="list-style-type: none"> <li>– Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos.</li> <li>– Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos.</li> <li>– Relación entre subtipos y la herencia.</li> </ul> </li> <li>• Uso de colección de clases, iteradores, y otros componentes de la librería estandar.</li> <li>• Asignación dinámica: definición de método de llamada.</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar e implementar una clase [Usar]</li> <li>• Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar]</li> <li>• Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar]</li> <li>• Comparar y contrastar (1) el enfoque procedural/funcional- definiendo una función por cada operación con el cuadro de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Evaluar]</li> <li>• Explicar la relación entre la herencia orientada a objetos (codigo compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse]</li> <li>• Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar]</li> <li>• Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma mas natural por cada lenguaje [Usar]</li> </ul>

Lecturas : [Str13], [Jos19], [Dei17]

## Unidad 11: Plantillas y STL (6 horas)

Resultados esperados: 6

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Plantillas de clases.</li> <li>• Conceptos básicos de la Biblioteca Estándar de Plantillas (STL) incluyendo: vector, list, stack, queue.</li> </ul>	<ul style="list-style-type: none"> <li>• Entiende los conceptos de plantillas en clases. [Familiarizarse]</li> <li>• Implementa y crea nuevos tipos de datos genéricos. [Usar]</li> <li>• Entiende las estructuras básicas de la STL. [Familiarizarse]</li> <li>• Usa las estructuras de datos básicas como: pila, cola, lista, vector contenidos en la STL. [Usar]</li> </ul>

Lecturas : [Str13], [Jos19], [Dei17]

<b>Unidad 12: Sobrecarga de operadores (4 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Definición de sobrecarga de operadores.</li> </ul>	<ul style="list-style-type: none"> <li>Entiende los conceptos de sobrecarga de operadores. [Familiarizarse]</li> <li>Implementa la sobrecarga de operadores permitidos en el lenguaje de programación. [Usar]</li> </ul>

Lecturas : [Str13], [Jos19], [Dei17]

<b>Unidad 13: Manejo de archivos (4 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Entrada y salida de archivos (I/O).</li> </ul>	<ul style="list-style-type: none"> <li>Entiende los conceptos de manipulación de archivos. [Familiarizarse]</li> <li>Crea programas de lectura y escrita en archivos. [Usar]</li> </ul>

Lecturas : [Str13], [Jos19], [Dei17]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013.

[Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017.

[Jos19] Nicolai M. Josuttis. *C++17 - The Complete Guide*. 1st. 2019.

## 1. CURSO

CS1D1-2-CS2023. Estructuras Discretas (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS1D1-2-CS2023. Estructuras Discretas
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Las estructuras discretas proporcionan los fundamentos teóricos necesarios para la computación. Dichos fundamentos no son sólo útiles para desarrollar la computación desde un punto de vista teórico como sucede en el curso de teoría de la computación, sino que también son útiles para la práctica de la computación; en particular se aplica en áreas como verificación, criptografía, métodos formales, etc.

## 5. OBJETIVOS

- Aplicar adecuadamente conceptos de la matemática finita (conjuntos, relaciones, funciones) para representar datos de problemas reales.
- Modelar situaciones reales descritas en lenguaje natural, usando lógica proposicional y lógica de predicados.
- Aplicar el método de demostración más adecuado para determinar la veracidad de un enunciado.
- Construir argumentos matemáticos correctos.
- Interpretar las soluciones matemáticas para un problema y determinar su fiabilidad, ventajas y desventajas.
- Expresar el funcionamiento de un circuito electrónico simple usando conceptos del Álgebra de Boole.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Usage)

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

## 7. TEMAS

<b>Unidad 1: Lógica básica (14 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Lógica proposicional.</li> <li>• Conectores lógicos.</li> <li>• Tablas de verdad.</li> <li>• Forma normal (conjuntiva y disyuntiva)</li> <li>• Validación de fórmula bien formada.</li> <li>• Reglas de inferencia proposicional (conceptos de modus ponens y modus tollens)</li> <li>• Lógica de predicados: <ul style="list-style-type: none"> <li>– Cuantificación universal y existencial</li> </ul> </li> <li>• Limitaciones de la lógica proposicional y de predicados (ej. problemas de expresividad)</li> </ul>	<ul style="list-style-type: none"> <li>• Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Usar]</li> <li>• Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar]</li> <li>• Usar reglas de inferencia para construir demostraciones en lógica proposicional y de predicados [Usar]</li> <li>• Describir como la lógica simbólica puede ser usada para modelar situaciones o aplicaciones de la vida real, incluidos aquellos planteados en el contexto computacional como análisis de software (ejm. programas correctores ), consulta de base de datos y algoritmos [Familiarizarse]</li> <li>• Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar]</li> <li>• Describir las fortalezas y limitaciones de la lógica proposicional y de predicados [Usar]</li> </ul>

**Lecturas :** [Ros07], [Gri03]

<b>Unidad 2: Técnicas de demostración (14 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Nociones de implicancia, equivalencia, conversión, inversa, contrapositivo, negación, y contradicción</li> <li>• Estructura de pruebas matemáticas.</li> <li>• Demostración directa.</li> <li>• Refutar por contraejemplo.</li> <li>• Demostración por contradicción.</li> <li>• Inducción sobre números naturales.</li> <li>• Inducción estructural.</li> <li>• Inducción leve y fuerte (Ej. Primer y Segundo principio de la inducción)</li> <li>• Definiciones matemáticas recursivas.</li> <li>• Conjuntos bien ordenados.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar la técnica de demostración utilizada en una demostración dada [Evaluar]</li> <li>• Describir la estructura básica de cada técnica de demostración (demostración directa, demostración por contradicción e inducción) descritas en esta unidad [Usar]</li> <li>• Aplicar las técnicas de demostración (demostración directa, demostración por contradicción e inducción) correctamente en la construcción de un argumento sólido [Usar]</li> <li>• Determinar qué tipo de demostración es la mejor para un problema dado [Evaluar]</li> <li>• Explicar el paralelismo entre ideas matemáticas y/o inducción estructural para la recursión y definir estructuras recursivamente [Familiarizarse]</li> <li>• Explicar la relación entre inducción fuerte y débil y dar ejemplos del apropiado uso de cada uno [Evaluar]</li> <li>• Enunciar el principio del buen-orden y su relación con la inducción matemática [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Ros07], [Epp10], [Sch12]	

<b>Unidad 3: Funciones, relaciones y conjuntos (22 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Conjuntos: <ul style="list-style-type: none"> <li>– Diagramas de Venn</li> <li>– Unión, intersección, complemento</li> <li>– Producto Cartesiano</li> <li>– Potencia de conjuntos</li> <li>– Cardinalidad de Conjuntos finitos</li> </ul> </li> <li>• Relaciones: <ul style="list-style-type: none"> <li>– Reflexividad, simetría, transitividad</li> <li>– Relaciones equivalentes, ordenes parciales</li> </ul> </li> <li>• Funciones: <ul style="list-style-type: none"> <li>– Suryecciones, inyecciones, biyecciones</li> <li>– Inversas</li> <li>– Composición</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Explicar con ejemplos la terminología básica de funciones, relaciones y conjuntos [Evaluar]</li> <li>• Realizar las operaciones asociadas con conjuntos, funciones y relaciones [Evaluar]</li> <li>• Relacionar ejemplos prácticos para conjuntos funciones o modelos de relación apropiados e interpretar la asociación de operaciones y terminología en contexto [Evaluar]</li> </ul>
<b>Lecturas :</b> [Gri03], [Ros07]	

<b>Unidad 4: Fundamentos de Lógica Digital (10 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Órdenes Parciales y Conjuntos Parcialmente Ordenados.</li> <li>Elementos extremos de un Conjunto Parcialmente Ordenado.</li> <li>Retículas: Tipos y propiedades.</li> <li>Álgebras Booleanas</li> <li>Funciones y expresiones Booleanas</li> <li>Representación de Funciones Booleanas: Forma Normal Disyuntiva y Conjuntiva</li> <li>Puertas Lógicas</li> <li>Minimización de funciones booleanas.</li> </ul>	<ul style="list-style-type: none"> <li>Explicar la importancia del Álgebra de Boole como unificación de la Teoría de Conjuntos y la Lógica Proposicional [Familiarizarse].</li> <li>Demostrar enunciados usando el concepto de retícula y sus propiedades [Evaluar].</li> <li>Explicar la relación entre retícula y conjunto parcialmente ordenado [Familiarizarse].</li> <li>Demostrar para una terna formada por un conjunto y dos operaciones internas, si cumple las propiedades de una Álgebra de Boole [Evaluar].</li> <li>Representar una función booleana en sus formas canónicas [Usar].</li> <li>Representar una función booleana como un circuito booleano usando puertas lógicas [Usar].</li> <li>Minimizar una función booleana [Usar].</li> </ul>
<b>Lecturas :</b> [Ros07], [Gri03]	

<b>Unidad 5: Fundamentos de conteo (10 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Técnicas de Conteo: <ul style="list-style-type: none"> <li>Conteo y cardinalidad de un conjunto</li> <li>Regla de la suma y producto</li> <li>Principio de inclusión-exclusión</li> <li>Progresión geométrica y aritmética</li> </ul> </li> <li>Principio de las casillas.</li> <li>Permutaciones y combinaciones: <ul style="list-style-type: none"> <li>Definiciones básicas</li> <li>Identidad de Pascal</li> <li>Teorema del binomio</li> </ul> </li> <li>Resolviendo relaciones de recurrencia: <ul style="list-style-type: none"> <li>Un ejemplo de una relación de recurrencia simple, como los números de Fibonacci</li> <li>Otras ejemplos, mostrando una variedad de soluciones</li> </ul> </li> <li>Aritmetica modular basica</li> </ul>	<ul style="list-style-type: none"> <li>Aplicar argumentos de conteo, incluyendo las reglas del producto y de la suma, principio de inclusión-exclusión y progresiones aritméticas/geométricas [Familiarizarse]</li> <li>Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse]</li> <li>Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse]</li> <li>Mapear aplicaciones del mundo real a formalismos de conteo adecuados, como el determinar el número de formas de acomodar a un conjunto de personas alrededor de una mesa, sujeto a restricciones en la disposición de los asientos, o en el número de maneras de determinar ciertas manos en juegos de cartas (ejm. una casa llena) [Familiarizarse]</li> <li>Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse]</li> <li>Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse]</li> <li>Realizar cálculos que involucran aritmética modular [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Gri97]	

<b>Unidad 6: Árboles y Grafos (10 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>● Árboles. <ul style="list-style-type: none"> <li>– Propiedades</li> <li>– Estrategias de recorrido</li> </ul> </li> <li>● Grafos no dirigidos</li> <li>● Grafos dirigidos</li> <li>● Grafos ponderados</li> <li>● Árboles de expansión/bosques.</li> <li>● Isomorfismo en grafos.</li> </ul>	<ul style="list-style-type: none"> <li>● Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Familiarizarse]</li> <li>● Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse]</li> <li>● Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Familiarizarse]</li> <li>● Demuestrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse]</li> <li>● Explicar como construir un árbol de expansión de un grafo [Familiarizarse]</li> <li>● Determinar si dos grafos son isomorfos [Familiarizarse]</li> </ul>

**Lecturas :** [Joh99]

<b>Unidad 7: Probabilidad Discreta (10 horas)</b> <b>Resultados esperados: 1,6,AG-C08,AG-C10,AG-C12</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Espacio de probabilidad finita, eventos.</li> <li>• Axiomas de Probabilidad y medidas de probabilidad.</li> <li>• Probabilidad condicional, Teorema de Bayes.</li> <li>• Independencia.</li> <li>• Variables enteras aleatorias (Bernoulli, binomial).</li> <li>• Esperado, Linearidad del esperado.</li> <li>• Varianza.</li> <li>• Independencia Condicional.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular las probabilidades de eventos y el valor esperado de variables aleatorias para problemas elementales como en los juegos de azar [Familiarizarse]</li> <li>• Distinguir entre eventos dependientes e independientes [Familiarizarse]</li> <li>• Identificar un caso de la distribución binomial y calcular la probabilidad usando dicha distribución [Familiarizarse]</li> <li>• Aplicar el teorema de Bayes para determinar las probabilidades condicionales en un problema [Familiarizarse]</li> <li>• Aplicar herramientas de probabilidades para resolver problemas como el análisis de caso promedio en algoritmos o en el análisis de hash [Familiarizarse]</li> <li>• Calcular la varianza para una distribución de probabilidad dada [Familiarizarse]</li> <li>• Explicar como los eventos que son independientes pueden ser condicionalmente dependientes (y vice versa) Identificar ejemplos del mundo real para estos casos [Familiarizarse]</li> </ul>

**Lecturas :** [Mic98]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.
- [Mic98] Elias Micha. *Matemáticas Discretas*. Limusa, 1998.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. 2007.
- [Epp10] Susanna S. Epp. *Discrete Mathematics with Applications*. 4 ed. 2010.
- [Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. 3 ed. 2012.

## 1. CURSO

MA103FCCS. Cálculo Integral (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	MA103FCCS. Cálculo Integral
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	5
<b>2.4 Horas</b>	:	4 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BMA102. Cálculo Diferencial. (1 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El cálculo integral es esencial en la ciencia de la computación 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, aplicaciones y su relación con el cálculo diferencial.

## 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, incluyendo aplicaciones en computación.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: La Integral Indefinida (6 horas)</b> <b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Antiderivadas y la integral indefinida.</li> <li>• Reglas básicas de integración.</li> <li>• Integración por sustitución.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular antiderivadas de funciones básicas. [Familiarizarse]</li> <li>• Aplicar las reglas básicas de integración. [Usar]</li> <li>• Resolver integrales indefinidas utilizando la técnica de sustitución. [Evaluar]</li> </ul>

**Lecturas :** [Ste15], [LE14]

<b>Unidad 2: La Integral Definida (6 horas)</b> <b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Sumas de Riemann y la integral definida.</li> <li>• El Teorema Fundamental del Cálculo.</li> <li>• Cálculo de áreas.</li> </ul>	<ul style="list-style-type: none"> <li>• Aproximar integrales definidas utilizando sumas de Riemann. [Familiarizarse]</li> <li>• Aplicar el Teorema Fundamental del Cálculo para evaluar integrales definidas. [Usar]</li> <li>• Calcular áreas bajo curvas utilizando integrales definidas. [Evaluar]</li> </ul>

**Lecturas :** [Ste15], [LE14]

<b>Unidad 3: Técnicas de Integración (12 horas)</b> <b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Integración por partes.</li> <li>• Integración de funciones trigonométricas.</li> <li>• Integración por fracciones parciales.</li> <li>• Integrales impropias.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar la técnica de integración por partes. [Familiarizarse]</li> <li>• Integrar funciones trigonométricas utilizando identidades y técnicas de sustitución. [Usar]</li> <li>• Resolver integrales utilizando la técnica de fracciones parciales. [Evaluar]</li> <li>• Evaluar integrales impropias. [Evaluar]</li> </ul>

**Lecturas :** [Ste15], [LE14]

<b>Unidad 4: Aplicaciones de la Integral Definida (12 horas)</b> <b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Cálculo de áreas entre curvas.</li> <li>• Cálculo de volúmenes de sólidos de revolución.</li> <li>• Longitud de arco.</li> <li>• Trabajo, valor promedio y centroides.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular el área entre dos curvas utilizando integrales definidas. [Familiarizarse]</li> <li>• Calcular el volumen de sólidos de revolución utilizando diferentes métodos. [Usar]</li> <li>• Calcular la longitud de arco de una curva. [Evaluar]</li> <li>• Aplicar integrales para calcular trabajo, valor promedio y centroides. [Evaluar]</li> </ul>

**Lecturas :** [Ste15], [LE14]

<b>Unidad 5: Aplicaciones en Computación (12 horas)</b> <b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Análisis de algoritmos (ej. cálculo de complejidad temporal).</li> <li>• Procesamiento de señales e imágenes (ej. transformadas integrales).</li> <li>• Probabilidad y estadística (ej. funciones de densidad de probabilidad).</li> </ul>	<ul style="list-style-type: none"> <li>• Utilizar integrales para analizar la complejidad temporal de algoritmos. [Familiarizarse]</li> <li>• Aplicar transformadas integrales en el procesamiento de señales e imágenes. [Usar]</li> <li>• Utilizar integrales en el contexto de funciones de densidad de probabilidad. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Ste15]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[LE14] Ron Larson and Bruce H. Edwards. *Calculus*. Cengage Learning, 2014.

[Ste15] James Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 2015.

## 1. CURSO

FI101FCCS. Física I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FI101FCCS. Física I
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	5
<b>2.4 Horas</b>	:	4 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La física es fundamental para comprender el mundo físico que las computadoras modelan, simulan e interactúan. Este curso establece las bases de la mecánica clásica, proporcionando el marco conceptual para entender movimiento, fuerzas, energía y momentum. Estos principios son directamente aplicables en áreas de la ciencia de la computación como desarrollo de videojuegos, simulaciones físicas realistas, robótica, animación por computadora y sistemas de control.

## 5. OBJETIVOS

- Comprender y aplicar las leyes fundamentales de la mecánica clásica en una y dos dimensiones.
- Desarrollar la capacidad de modelar problemas físicos de manera matemática y resolverlos analíticamente.
- Relacionar los conceptos físicos con sus aplicaciones en el ámbito de la computación.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Usage)

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

## 7. TEMAS

<b>Unidad 1: Cinemática en una y dos Dimensiones (10 horas)</b> <b>Resultados esperados: 1,AG-C07,AG-C08</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Movimiento rectilíneo: posición, desplazamiento, velocidad y aceleración.</li> <li>• Movimiento con aceleración constante (caída libre).</li> <li>• Movimiento en dos dimensiones: movimiento de proyectiles.</li> <li>• Movimiento circular uniforme y no uniforme.</li> <li>• Cantidades vectoriales en cinemática.</li> </ul>	<ul style="list-style-type: none"> <li>• Diferenciar entre posición, desplazamiento, velocidad y aceleración de manera conceptual y matemática. [Familiarizarse]</li> <li>• Resolver problemas de movimiento en una y dos dimensiones, incluyendo proyectiles y movimiento circular. [Usar]</li> <li>• Modelar cinemáticamente situaciones físicas simples. [Evaluar]</li> </ul>
<b>Lecturas :</b> <a href="#">[YoungFreedman2018]</a> , <a href="#">[SerwayJewett2018]</a>	

<b>Unidad 2: Dinámica: Leyes de Newton y Aplicaciones (12 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Concepto de fuerza, masa y inercia.</li> <li>• Las tres Leyes de Newton del movimiento.</li> <li>• Diagramas de cuerpo libre.</li> <li>• Aplicaciones de las Leyes de Newton: planos inclinados, poleas, sistemas conectados.</li> <li>• Fuerzas de fricción estática y cinética.</li> <li>• Fuerzas de resistencia y velocidad terminal.</li> </ul>	<ul style="list-style-type: none"> <li>• Enunciar las Leyes de Newton y explicar su significado físico. [Familiarizarse]</li> <li>• Construir diagramas de cuerpo libre y aplicar las Leyes de Newton para resolver problemas de dinámica. [Usar]</li> <li>• Analizar sistemas con múltiples cuerpos y fuerzas de fricción. [Evaluar]</li> </ul>
<b>Lecturas :</b> <a href="#">[YoungFreedman2018]</a> , <a href="#">[SerwayJewett2018]</a>	

<b>Unidad 3: Trabajo y Energía (10 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Trabajo realizado por una fuerza constante y variable.</li> <li>• Energía cinética y el Teorema del Trabajo-Energía.</li> <li>• Potencia.</li> <li>• Fuerzas conservativas y no conservativas.</li> <li>• Energía potencial gravitacional y elástica.</li> <li>• Conservación de la energía mecánica.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular el trabajo realizado por diferentes fuerzas. [Familiarizarse]</li> <li>• Aplicar el teorema del trabajo-energía para resolver problemas de dinámica. [Usar]</li> <li>• Aplicar el principio de conservación de la energía mecánica en sistemas conservativos. [Evaluar]</li> </ul>
<b>Lecturas :</b> <a href="#">[YoungFreedman2018]</a> , <a href="#">[SerwayJewett2018]</a>	

<b>Unidad 4: Sistemas de Partículas y Conservación del Momento Lineal (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Centro de masa y su movimiento.</li> <li>Momento lineal (cantidad de movimiento) e impulso.</li> <li>Conservación del momento lineal.</li> <li>Colisiones en una y dos dimensiones: elásticas e inelásticas.</li> </ul>	<ul style="list-style-type: none"> <li>Calcular la posición y velocidad del centro de masa de un sistema de partículas. [Familiarizarse]</li> <li>Aplicar el principio de conservación del momento lineal para analizar colisiones y explosiones. [Usar]</li> <li>Diferenciar entre colisiones elásticas e inelásticas y aplicar las leyes de conservación apropiadas. [Evaluar]</li> </ul>

Lecturas : [YoungFreedman2018], [SerwayJewett2018]

<b>Unidad 5: Dinámica Rotacional (10 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Cinemática rotacional: variables angulares.</li> <li>Torque y equilibrio estático.</li> <li>Momento de inercia y la segunda ley de Newton para la rotación.</li> <li>Energía cinética rotacional.</li> <li>Rodadura sin deslizamiento.</li> <li>Momento angular y su conservación (conceptos introductorios).</li> </ul>	<ul style="list-style-type: none"> <li>Relacionar variables lineales y angulares. [Familiarizarse]</li> <li>Calcular torques y momentos de inercia para configuraciones simples. [Usar]</li> <li>Aplicar las leyes de la dinámica rotacional para resolver problemas que involucren rotación y rodadura. [Evaluar]</li> </ul>

Lecturas : [YoungFreedman2018], [SerwayJewett2018]

<b>Unidad 6: Aplicaciones en Computación (6 horas)</b> <b>Resultados esperados: 1,AG-C07,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Simulaciones físicas: integración numérica (método de Euler) para movimiento.</li> <li>Motores de física en videojuegos: colisiones, gravedad y movimiento de proyectiles.</li> <li>Robótica: cinemática de brazos robóticos simples.</li> <li>Animación por computadora: trayectorias y movimientos realistas.</li> </ul>	<ul style="list-style-type: none"> <li>Describir cómo los principios de la cinemática y dinámica se implementan en simulaciones por computadora. [Familiarizarse]</li> <li>Explicar el concepto básico de un motor de física en el desarrollo de videojuegos. [Usar]</li> <li>Modelar un problema físico simple (ej. lanzamiento de proyectil) de manera algorítmica. [Evaluar]</li> </ul>

Lecturas : [Bou02], [NS05]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Bou02] David M. Bourg. *Physics for Game Developers*. Excelente para ver aplicaciones directas de la física mecánica en la programación de videojuegos. O'Reilly Media, 2002.
- [NS05] Noam Nisan and Shimon Schocken. *The Elements of Computing Systems: Building a Modern Computer from First Principles*. Ayuda a entender el viaje desde la física hasta la lógica de alto nivel. MIT Press, 2005.

## 1. CURSO

ID102. Inglés II (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	ID102. Inglés II
<b>2.2 Semestre</b>	:	2 <sup>do</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BEI101. Inglés I. (1 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida de las personas. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el Inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento, como parte de su formación integral.

## 5. OBJETIVOS

- Desarrollar la capacidad de hablar fluídamente el idioma.
- Incrementar el vocabulario y el manejo de expresiones simples.

## 6. RESULTADOS DEL ESTUDIANTE

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

## 7. TEMAS

<b>Unidad 1: How long ago? (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Pasado Simple</li> <li>• Oraciones Negativas con ago.</li> <li>• Conjunciones</li> <li>• Expresiones de Tiempo en pasado</li> <li>• Relaciones y símbolos fonéticos</li> <li>• Expresiones para dar la fecha</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la octava unidad, cada uno de los alumnos, comprendiendo la gramática del tiempo pasado es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de fechas y números en orden.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 2: Food you like! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Sustantivos Contables y No Contables</li> <li>• Expresiones con Would like y I'd like</li> <li>• Cuantificadores</li> <li>• Comidas alrededor del mundo</li> <li>• Pedidos formales</li> <li>• Cartas formales</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar pedidos y hacer ofrecimientos, los utilizan en situaciones varias. Expresar situaciones y estados relacionados con cantidades. Explica y aplica vocabulario de comidas y bebidas.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 3: The world of work (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Adjetivos</li> <li>• Oraciones con Adjetivos Comparativos.</li> <li>• Oraciones con Adjetivos Superlativos</li> <li>• Ciudades y el campo</li> <li>• Indicaciones de dirección</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la décima unidad, los alumnos habiendo reconocido las características de los adjetivos, utilizan éstos para hacer comparaciones de diversos tipos. Describen personas y lugares y dan indicaciones de dirección. Utilizarán conjunciones para unir ideas tipo.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 4: Looking good! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Presente Continuo</li> <li>• Oraciones Afirmativas, Negativas y Preguntas</li> <li>• Uso de Whose</li> <li>• Pronombres Posesivos</li> <li>• Ropa y colores</li> <li>• Expresiones a usar en tiendas de ropa</li> <li>• Símbolos fonéticos.</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la décimo primera unidad, los alumnos habiendo identificado la idea de expresar ideas de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente Progresivo. Expresan ideas de posesión con respecto a la ropa y los colores.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 5: Life is an adventure! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Uso de going to</li> <li>• Oraciones en Tiempo Futuro</li> <li>• Expresiones de Cantidad.</li> <li>• Verbos de acción</li> <li>• Vocabulario del clima</li> <li>• Expresiones de Sugerencia</li> <li>• Escribir una postal</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar infinitivos de propósito. Adquirirán vocabulario para describir el clima. Se presentará expresiones para hacer y pedir sugerencias.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 6: You're pretty smart! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Formas de Preguntas</li> <li>• Adverbios y Adjetivos</li> <li>• Vocabulario descripción de sentimientos</li> <li>• Expresiones para viajes en tren</li> <li>• Redacción de historias cortas</li> <li>• Lecturas.</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo tercera unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas diversas, realizarán trabajos aplicativos en contextos adecuados. Enfatizan la diferencia entre adjetivos y adverbios. Describen sentimientos. Utilizan expresiones para coger un tren. Asumen la idea de sufijos y prefijos.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 7: Have you ever? (0 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Presente Perfecto</li> <li>• Expresiones con never, ever y yet</li> <li>• Vocabulario verbos en Participio pasado</li> <li>• Expresiones para viajes en avión</li> <li>• Redacción de cartas de agradecimiento</li> <li>• Lecturas</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del Presente Perfecto experimentan la necesidad de poder expresar este tipo de tiempo en acciones. Realizarán prácticas en contextos adecuados. Enfatizan la diferencia entre pasado simple y presente perfecto. Describen acciones con never, ever y yet. Utilizan expresiones para utilizar en un aeropuerto.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 1 Student Book*. Editorial Oxford, 2002.

[Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.

## Capítulo 3

### Tercer Semestre

## 1. CURSO

CS113-CS2023. Programación Orientada a Objetos II (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS113-CS2023. Programación Orientada a Objetos II
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS112. Programación Orientada a Objetos I. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este es el tercer curso de la secuencia introductoria en Ciencia de la Computación. El curso profundiza en el dominio avanzado de la Programación Orientada a Objetos (POO) en C++, enfocándose en el desarrollo de sistemas de alto rendimiento. Los temas principales incluyen:

### Conceptos Avanzados Fundamentales:

- Metaprogramación con plantillas (TMP) y *Substitution Failure Is Not An Error* (SFINAE)
- Semántica de movimiento (*move semantics*), *perfect forwarding* y optimización RAI (Resource Acquisition Is Initialization)
- Problemas comunes de herencia múltiple y patrones de herencia virtual

### Programación Concurrente y de Sistemas:

- `std::thread`, `async/await` y sincronización de hilos
- Programación *lock-free* con tipos atómicos
- Patrones de diseño POO seguros en entornos multihilo (ej. variantes de *singleton*)

### Paradigmas Modernos de C++:

- CRTP (*Curiously Recurring Template Pattern*)
- Plantillas de expresiones para código crítico en rendimiento
- Introspección en tiempo de compilación con `constexpr` y `type traits`

### Aplicaciones Prácticas:

- Interfaz entre C++ y otros lenguajes (FFI - *Foreign Function Interface*)
- Benchmarking y perfilado de código con uso intensivo de plantillas

Prepara a los estudiantes para desarrollo de motores de juego, computación de alto rendimiento (HPC) y sistemas embebidos, áreas donde C++ es dominante.

## 5. OBJETIVOS

- Introducir al alumno en los conceptos relacionados con Punteros en C y C++, permitiendo asimilar esta técnica de programación, la misma que es necesaria para desarrollar algoritmos y estructuras de datos eficientes.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: STL Avanzado (8 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Contenedores asociativos (std::set, std::map, std::unordered_set, std::unordered_map).</li><li>Adaptadores (std::stack, std::queue, std::priority_queue).</li><li>Algoritmos avanzados de la STL.</li><li>Functores y predicados.</li></ul>	<ul style="list-style-type: none"><li>Comprender el uso de contenedores asociativos. [Usar]</li><li>Implementar programas que utilicen adaptadores de la STL. [Usar]</li><li>Aplicar algoritmos avanzados de la STL. [Usar]</li><li>Utilizar functores y predicados con la STL. [Usar]</li></ul>
<b>Lecturas :</b> [Str13], [MJo19], [VJG17], [Dei17]	

<b>Unidad 2: Plantillas Avanzadas (7 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Metaprogramación con plantillas.</li><li>SFINAE (<i>Substitution Failure Is Not An Error</i>).</li><li>Perfect Forwarding.</li></ul>	<ul style="list-style-type: none"><li>Aplicar la metaprogramación con plantillas para resolver problemas complejos. [Usar]</li><li>Comprender y utilizar SFINAE para la selección de plantillas. [Usar]</li><li>Utilizar Perfect Forwarding para el paso eficiente de argumentos. [Usar]</li></ul>
<b>Lecturas :</b> [Str13], [MJo19], [VJG17], [Dei17]	

<b>Unidad 3: Expresiones Lambda y Funciones de Orden Superior (6 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Expresiones lambda (sintaxis, captura de variables).</li> <li>• Uso de lambdas con la STL.</li> <li>• Funciones de orden superior (ejemplos con la STL).</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender la sintaxis y el uso de expresiones lambda. [Usar]</li> <li>• Utilizar expresiones lambda con algoritmos de la STL. [Usar]</li> <li>• Aplicar funciones de orden superior con la STL. [Usar]</li> </ul>

**Lecturas :** [Str13], [MJo19], [VJG17], [Dei17]

<b>Unidad 4: Semántica de Movimiento y Referencias Rvalue (5 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Lvalues y Rvalues.</li> <li>• Referencias Rvalue.</li> <li>• Semántica de Movimiento.</li> <li>• Constructores de movimiento y operadores de asignación de movimiento.</li> <li>• Perfect Forwarding (repaso).</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la semántica de movimiento y su propósito en C++. [Familiarizarse]</li> <li>• Definir y usar referencias rvalue. [Usar]</li> <li>• Analizar las implicaciones de rendimiento del uso de la semántica de movimiento. [Evaluar]</li> <li>• Implementar constructores de movimiento y operadores de asignación de movimiento para clases personalizadas. [Usar]</li> <li>• Aplicar la semántica de movimiento para optimizar la gestión de recursos en programas C++. [Usar]</li> </ul>

**Lecturas :** [Str13], [MJo19], [VJG17], [Dei17]

<b>Unidad 5: Patrones de Diseño (Creacionales y Estructurales) (6 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Singleton, Factory, Builder.</li> <li>• Adapter, Decorator, Facade.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender y aplicar los patrones de diseño creacionales: Singleton, Factory, Builder. [Usar]</li> <li>• Comprender y aplicar los patrones de diseño estructurales: Adapter, Decorator, Facade. [Usar]</li> </ul>

**Lecturas :** [Str13], [MJo19], [VJG17], [Dei17]

<b>Unidad 6: Functores (3 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Definición de functores.</li> <li>• Functores y templates.</li> <li>• Paso de functores a funciones usando parámetros.</li> <li>• Paso de functores a funciones usando templates.</li> <li>• Paso de functores a clases usando parámetros.</li> <li>• Paso de functores a clases usando templates.</li> <li>• Ejemplos y aplicaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducción a los functores. [Usar]</li> <li>• Uso de functores como parámetros a funciones y clases. [Usar]</li> <li>• Uso de functores en funciones y clases a través de templates. [Usar]</li> </ul>
<b>Lecturas :</b> [Str13], [MJo19], [VJG17], [Dei17]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Str13] Bjarne Stroustrup. *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013.
- [Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017.
- [VJG17] David Vandervoorde, Nicolai Josuttis, and Douglas Gregor. *C++ Templates: The Complete Guide*. 2nd. Addison-Wesley, 2017.
- [MJo19] Nicolai M.Josuttis. *C++17-The Complete Guide*. 1st. 2019.

## 1. CURSO

CS2B1-CS2023. Desarrollo Basado en Plataformas (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS2B1-CS2023. Desarrollo Basado en Plataformas
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS112. Programación Orientada a Objetos I. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El mundo ha cambiado debido al uso de la web y tecnologías relacionadas, el acceso rápido, oportuno y personalizado de la información, a través de la tecnología web, ubicuo y pervasiva; han cambiado la forma de ¿cómo hacemos las cosas?, ¿cómo pensamos? y ¿cómo la industria se desarrolla?.

Las tecnologías web, ubicuo y pervasivo se basan en el desarrollo de servicios web, aplicaciones web y aplicaciones móviles, las cuales son necesarias entender la arquitectura, el diseño, y la implementación de servicios web, aplicaciones web y aplicaciones móviles.

## 5. OBJETIVOS

- Que el alumno sea capaz de diseño e implementación de servicios, aplicaciones web utilizando herramientas y lenguajes como HTML, CSS, JavaScript (incluyendo AJAX) , back-end scripting y una base de datos, a un nivel intermedio.
- Que el alumno sea capaz de desarrollar aplicaciones móviles, administrar servidores web en sistemas basados en UNIX y aplicar técnicas de seguridad en la web a un nivel intermedio.

## 6. RESULTADOS DEL ESTUDIANTE

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

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

## 7. TEMAS

<b>Unidad 1: Introducción (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial)</li> <li>Programación a través de APIs específicos.</li> <li>Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5)</li> <li>Programación bajo restricciones de plataforma.</li> </ul>	<ul style="list-style-type: none"> <li>Describir cómo el desarrollo basado en plataforma difiere de la programación de propósito general [Familiarizarse]</li> <li>Listar las características de lenguajes de plataforma [Familiarizarse]</li> <li>Escribir y ejecutar un programa simple basado en plataforma [Familiarizarse]</li> <li>Listar las ventajas y desventajas de la programación con restricciones de plataforma [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Fie00], [Fie21], [Gro09], [ADC13]	

<b>Unidad 2: Plataformas web (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Lenguajes de programación web (e.g., HTML5, Javascript, PHP, CSS)</li> <li>Restricciones de las plataformas web: Client-Server, Stateless-Stateful, Caché, Uniform Interface, Layered System, Code on Demand, ReST.</li> <li>Restricción de plataformas web.</li> <li>Software como servicio.</li> <li>Estándares web.</li> </ul>	<ul style="list-style-type: none"> <li>Diseñar e implementar una aplicación web sencilla [Familiarizarse]</li> <li>Describir las limitaciones que la web pone a los desarrolladores [Familiarizarse]</li> <li>Comparar y contrastar la programación web con la programación de propósito general [Familiarizarse]</li> <li>Describir las diferencias entre software como un servicio y productos de software tradicionales [Familiarizarse]</li> <li>Discutir cómo los estándares de web impactan el desarrollo de software [Familiarizarse]</li> <li>Revise una aplicación web existente con un estándar web actual [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Fie00], [Fie21]	

**Unidad 3: Desarrollo de servicios y aplicaciones web (25 horas)**
**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Describir, identificar y depurar problemas relacionados con el desarrollo de aplicaciones web.</li> <li>Diseño y desarrollo de aplicaciones web interactivas usando HTML5 y Python.</li> <li>Utilice MySQL para la gestión de datos y manipular MySQL con Python.</li> <li>Diseño y desarrollo de aplicaciones web asíncronos utilizando técnicas Ajax.</li> <li>Uso del lado del cliente dinámico lenguaje de script Javascript y del lado del servidor lenguaje de scripting python con Ajax.</li> <li>Aplicar las tecnologías XML / JSON para la gestión de datos.</li> <li>Utilizar los servicios, APIs Web, Ajax y aplicar los patrones de diseño para el desarrollo de aplicaciones web.</li> </ul>	<ul style="list-style-type: none"> <li>Del lado del servidor lenguaje de scripting python: variables, tipos de datos, operaciones, cadenas, funciones, sentencias de control, matrices, archivos y el acceso a directorios, mantener el estado. [Usar]</li> <li>Enfoque de programación web usando python incrustado. [Usar]</li> <li>El acceso y la manipulación de MySQL. [Usar]</li> <li>El enfoque de desarrollo de aplicaciones web Ajax. [Usar]</li> <li>DOM y CSS utilizan en JavaScript. [Usar]</li> <li>Tecnologías de actualización de contenido asíncrono. [Usar]</li> <li>Objetos XMLHttpRequest utilizar para comunicarse entre clientes y servidores. [Usar]</li> <li>XML y JSON. [Usar]</li> <li>XSLT y XPath como mecanismos para transformar documentos XML. [Usar]</li> <li>Servicios web y APIs (especialmente Google Maps). [Usar]</li> <li>Marcos Ajax para el desarrollo de aplicaciones web contemporánea. [Usar]</li> <li>Los patrones de diseño utilizados en aplicaciones web. [Usar]</li> </ul>

**Lecturas :** [FR11]

**Unidad 4: Plataformas móviles (5 horas)**
**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Lenguajes de Programación para Móviles.</li> <li>Principios de diseño: Segregación de Interfaces, Responsabilidad Única, Separación de Responsabilidades, Inversión de Dependencias.</li> <li>Desafíos con mobilidad y comunicación inalámbrica.</li> <li>Aplicaciones Location-aware.</li> <li>Rendimiento / Compensación de Potencia.</li> <li>Restricciones de las Plataformas Móviles.</li> <li>Tecnologías Emergentes.</li> </ul>	<ul style="list-style-type: none"> <li>Diseñar e implementar una aplicación móvil para una plataforma móvil dada [Familiarizarse]</li> <li>Discutir las limitaciones que las plataformas móviles ponen a los desarrolladores [Familiarizarse]</li> <li>Discutir los principios de diseño que guian la construcción de aplicaciones móviles [Familiarizarse]</li> <li>Discutir el rendimiento vs perdida de potencia [Familiarizarse]</li> <li>Compare y contraste la programación móvil con la programación de propósito general [Familiarizarse]</li> </ul>

**Lecturas :** [Mar17]

## Unidad 5: Aplicaciones Móviles para dispositivos Android (25 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• The Android Platform</li> <li>• The Android Development Environment</li> <li>• Application Fundamentals</li> <li>• The Activity Class</li> <li>• The Intent Class</li> <li>• Permissions</li> <li>• The Fragment Class</li> <li>• User Interface Classes</li> <li>• User Notifications</li> <li>• The BroadcastReceiver Class</li> <li>• Threads, AsyncTask &amp; Handlers</li> <li>• Alarms</li> <li>• Networking (http class)</li> <li>• Multi-touch &amp; Gestures</li> <li>• Sensors</li> <li>• Location &amp; Maps</li> </ul>	<ul style="list-style-type: none"> <li>• Los estudiantes identifican software necesario y lo instalan en sus ordenadores personales. Los estudiantes realizan varias tareas para familiarizarse con la plataforma Android y Ambiente para el Desarrollo. [Usar]</li> <li>• Los estudiantes construyen aplicaciones que trazan los métodos de devolución de llamada de ciclo de vida emitidas por la plataforma Android y que demuestran el comportamiento de Android cuando los cambios de configuración de dispositivos (por ejemplo, cuando el dispositivo se mueve de vertical a horizontal y viceversa). [Usar]</li> <li>• Los estudiantes construyen aplicaciones que requieren iniciar múltiples actividades a través de ambos métodos estándar y personalizados. [Usar]</li> <li>• Los estudiantes construyen aplicaciones que requieren permisos estándar y personalizados. [Usar]</li> <li>• Los estudiantes construyen una aplicación que utiliza una única base de código, sino que crea diferentes interfaces de usuario dependiendo del tamaño de la pantalla de un dispositivo. [Usar]</li> <li>• Los estudiantes construyen un gestor de listas de tareas pendientes utilizando los elementos de la interfaz de usuario discutidos en clase. La aplicación permite a los usuarios crear nuevos elementos y para mostrarlos en un ListView. [Usar]</li> <li>• Los estudiantes construyen una aplicación que utiliza la información de ubicación para recoger latitud, longitud de los lugares que visitan. [Usar]</li> </ul>

Lecturas : [ADC13]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Fie00] Roy Thomas Fielding. “Fielding dissertation: Chapter 5: Representational state transfer (rest)”. In: [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm) (2000).

- 
- [Gro09] R. Grove. *Web Based Application Development*. Jones & Bartlett Learning, 2009.
- [FR11] Eric Freeman and Elisabeth Robson. *Head first HTML5 programming: building web apps with JavaScript*. O'Reilly Media, Inc., 2011.
- [ADC13] J. Annuzzi, L. Darcey, and S. Conder. *Introduction to Android Application Development: Android Essentials*. Developer's Library. Pearson Education, 2013.
- [Mar17] Robert C Martin. *Clean architecture: a craftsman's guide to software structure and design*. Prentice Hall Press, 2017.
- [Fie21] Roy T. Fielding. “REST APIs Must Be Hypertext-Driven: Revisiting the Architectural Constraints”. In: *IEEE Software* 38.3 (2021). Artículo del creador de REST actualizando sus principios para aplicaciones modernas., pp. 27–32.

## 1. CURSO

AI161. IA Aplicada (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	AI161. IA Aplicada
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS111. Introducción a la Programación. (1 <sup>er</sup> 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

- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Usage)

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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

<b>Unidad 1: Fundamentos de IA y Panorama Global (16 horas)</b>	
<b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• ¿Qué es la IA? Definiciones, historia y panorama global actual.</li><li>• Tipos de sistemas de IA: chatbots, generadores de imágenes, asistentes de investigación entre regiones.</li><li>• Ecosistemas occidentales de IA: OpenAI, Anthropic, Google, Microsoft.</li><li>• Ecosistemas chinos de IA: DeepSeek, Kimi, ERNIE, Zhipu AI, Baidu.</li><li>• Capacidades y limitaciones de la IA: análisis comparativo entre plataformas.</li><li>• Alfabetización digital en la era de la IA: pensamiento crítico sobre diversos resultados de IA.</li></ul>	<ul style="list-style-type: none"><li>• Aplicar conocimientos fundamentales de IA para identificar tipos de sistemas y sus capacidades [Usar].</li><li>• Analizar sistemas inteligentes globales y sus características distintivas [Evaluar].</li><li>• Evaluar críticamente los resultados de IA en diferentes contextos culturales [Evaluar].</li></ul>
<b>Lecturas :</b> [Ng24], [MM24], [UNE23], [Ins24b]	

<b>Unidad 2: Interacción Efectiva con IA e Ingeniería de Prompts Multiplataforma (20 horas)</b>	
<b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Principios de escritura efectiva de prompts: claridad, contexto, restricciones entre plataformas.</li><li>• Patrones de prompts: persona, plantilla, cadena de pensamiento, few-shot para diferentes sistemas de IA.</li><li>• Herramientas occidentales: técnicas específicas para ChatGPT, Claude, Gemini, Copilot.</li><li>• Herramientas chinas: características específicas y mejores prácticas para DeepSeek, Kimi, ERNIE.</li><li>• Estrategias multiplataforma: aprovechar diferentes fortalezas de IA para tareas complejas.</li><li>• Refinamiento iterativo: cómo mejorar prompts basándose en resultados de diversos sistemas.</li><li>• Sesiones prácticas: talleres comparativos de escritura de prompts entre plataformas.</li></ul>	<ul style="list-style-type: none"><li>• Diseñar soluciones de prompts efectivas para diversos sistemas de IA [Evaluar].</li><li>• Desarrollar flujos de trabajo multiplataforma que integren diferentes herramientas de IA [Usar].</li><li>• Implementar estrategias de refinamiento iterativo para optimizar resultados de IA [Usar].</li></ul>
<b>Lecturas :</b> [Whi+23], [Ins24a], [Mic24], [Dee24]	

<b>Unidad 3: Aplicaciones de IA entre Disciplinas y Plataformas (16 horas)</b>	
<b>Resultados esperados: 2,6,AG-C09,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• IA para investigación: revisión de literatura, análisis de datos usando múltiples asistentes de IA.</li> <li>• IA para escritura: documentos académicos, informes con verificación multiplataforma.</li> <li>• IA para resolución de problemas: cálculos científicos, diseño de ingeniería con herramientas especializadas.</li> <li>• IA para creatividad: lluvia de ideas, desarrollo de conceptos entre contextos culturales.</li> <li>• Fortalezas específicas por plataforma: cuándo usar herramientas de IA occidentales vs. chinas.</li> <li>• Talleres disciplinares: aplicaciones personalizadas usando diversos ecosistemas de IA.</li> <li>• Estudios de caso: aplicaciones en el mundo real en investigación científica entre regiones.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar fundamentos de IA para resolver problemas específicos de cada disciplina [Usar].</li> <li>• Diseñar soluciones que integren múltiples sistemas inteligentes para tareas complejas [Evaluar].</li> <li>• Desarrollar aplicaciones prácticas usando herramientas de IA occidentales y chinas [Usar].</li> <li>• Implementar flujos de trabajo que aprovechen fortalezas específicas de cada plataforma [Evaluar].</li> </ul>

**Lecturas :** [Mol23], [Goo24], [Clo24], [Ope24]

<b>Unidad 4: Ética Global en IA y Uso Responsable (12 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Comprensión de sesgos en IA: perspectivas culturales occidentales y orientales.</li> <li>• Marcos éticos: comparando la Ley de IA de la UE, regulaciones chinas y estándares globales.</li> <li>• Integridad académica: citación apropiada y uso de IA en trabajos académicos entre plataformas.</li> <li>• Privacidad y seguridad de datos: diferencias regionales en el manejo de datos de IA.</li> <li>• Sensibilidad cultural: navegando resultados de IA en contextos globales.</li> <li>• Impacto ambiental: consideraciones de sostenibilidad de diferentes sistemas de IA.</li> <li>• Desarrollo de pautas personales para el uso ético de IA en entornos internacionales.</li> <li>• Estudios de caso: dilemas éticos en el despliegue de IA entre regiones.</li> </ul>	<ul style="list-style-type: none"> <li>• Demostrar responsabilidad profesional en el uso de tecnologías de IA [Evaluar].</li> <li>• Aplicar principios éticos profesionales en contextos globales de IA [Usar].</li> <li>• Desarrollar pautas éticas para mitigar sesgos en sistemas de IA [Evaluar].</li> <li>• Evaluar implicaciones éticas del despliegue de IA en diferentes culturas [Usar].</li> </ul>

**Lecturas :** [UNE23], [Uni24], [BG+21], [Cyd24], [Lab23]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### 10. BIBLIOGRAFÍA BÁSICA

- [BG+21] Emily M. Bender, Timnit Gebru, et al. “Sobre los Peligros de los Loros Estocásticos: ¿Pueden los Modelos de Lenguaje Ser Demasiado Grandes?” In: *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (2021). Artículo fundamental sobre ética en IA con relevancia continua.
- [Lab23] Tencent AI Lab. *Prácticas de IA Responsable en la Industria Tecnológica China*. Perspectiva industrial sobre ética en IA de empresa tecnológica china líder. 2023. URL: <https://ai.tencent.com/>.
- [Mol23] Ethan Mollick. “ChatGPT y Cómo la IA Disrumppe Industrias”. In: *Harvard Business Review* (2023). Análisis del impacto práctico de la IA en sectores globales.
- [UNE23] UNESCO. *IA y Educación: Guía para Responsables de Políticas*. Guía más reciente sobre alfabetización en IA e implementación ética en educación. 2023. URL: <https://unesdoc.unesco.org/>.
- [Whi+23] Jules White et al. *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. Investigación revisada por pares sobre patrones de ingeniería de prompts. 2023. URL: <https://arxiv.org/abs/2302.11382>.
- [Clo24] Alibaba Cloud. *Ética en IA y Mejores Prácticas en Contexto Chino*. Perspectiva china sobre ética en IA y guías de implementación. 2024. URL: <https://www.alibabacloud.com/>.
- [Cyd24] Chinese Academy of Cyberspace Studies. *Gobernanza y Ética de IA en China*. Perspectiva oficial china sobre gobernanza de IA y estándares éticos. 2024. URL: <http://www.cac.gov.cn/>.
- [Dee24] DeepSeek. *Documentación del Modelo DeepSeek y Mejores Prácticas*. Documentación oficial para modelos de IA DeepSeek y guías de uso. 2024. URL: <https://platform.deepseek.com/>.
- [Goo24] Google. *Introducción a la IA Responsable*. Marco actualizado de Google para desarrollo y uso responsable de IA. 2024. URL: <https://cloud.google.com/learn/responsible-ai>.
- [Ins24a] Prompt Engineering Institute. *Guía de Ingeniería de Prompts*. Recurso integral de ingeniería de prompts actualizado continuamente. 2024. URL: <https://www.promptingguide.ai/>.
- [Ins24b] Tencent Research Institute. *Desarrollo de IA en China: Estado Actual y Tendencias Futuras*. Análisis integral del desarrollo del ecosistema chino de IA. 2024. URL: <https://tri.tencent.com/>.
- [Mic24] Microsoft. *Elaboración de Prompts para Sistemas de IA*. Guía oficial de Microsoft para ingeniería de prompts actualizada para 2024. 2024. URL: <https://learn.microsoft.com/en-us/training/prompt-engineering>.
- [MM24] Ethan Mollick and Lilach Mollick. *Co-Intelligence: Living and Working with AI*. Guía práctica para la colaboración humano-IA. Penguin Random House, 2024.
- [Ng24] Andrew Ng. *AI for Everyone*. Curso en línea actualizado que cubre fundamentos de IA para audiencias no técnicas. 2024. URL: <https://wwwdeeplearning.ai/courses/ai-for-everyone/>.
- [Ope24] OpenAI. *Mejores Prácticas para Ingeniería de Prompts*. Guías oficiales actualizadas de ingeniería de prompts de OpenAI. 2024. URL: <https://platform.openai.com/docs/guides/prompt-engineering>.
- [Uni24] European Union. *Ley de IA: Marco Regulatorio*. Marco integral de regulación de IA efectivo en 2024. 2024. URL: <https://digital-strategy.ec.europa.eu/en/policies/ai-act>.

## 1. CURSO

MA104FCCS. Cálculo Diferencial e Integral Avanzado (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	MA104FCCS. Cálculo Diferencial e Integral Avanzado
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	5
<b>2.4 Horas</b>	:	4 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BMA102. Cálculo Diferencial. (1 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El cálculo diferencial e integral avanzado es fundamental para la comprensión de fenómenos multidimensionales en ingeniería y ciencias de la computación. Este curso profundiza en los conceptos del cálculo vectorial, incluyendo funciones vectoriales, cálculo multivariable, integrales múltiples y teoremas fundamentales. Estas herramientas son esenciales para modelar sistemas complejos, optimizar funciones multivariadas y resolver problemas en gráficos por computadora, visión artificial y simulaciones físicas.

## 5. OBJETIVOS

- Comprender y manipular funciones vectoriales de una y varias variables.
- Aplicar el cálculo diferencial e integral a problemas en múltiples dimensiones.
- Utilizar los teoremas de Green, Gauss y Stokes en aplicaciones prácticas.
- Modelar y resolver problemas de ingeniería usando integrales de línea y superficie.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**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. (Familiarity)

## 7. TEMAS

Unidad 1: Funciones Vectoriales de una Variable Real (14 horas)	
Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Definición de función vectorial de variable real.</li> <li>Límites, continuidad y derivada de funciones vectoriales.</li> <li>Vector tangente, normal y binormal.</li> <li>Curvatura, torsión y fórmulas de Frenet.</li> <li>Longitud de arco y parametrización.</li> <li>Plano osculador y triedro móvil.</li> </ul>	<ul style="list-style-type: none"> <li>Calcular derivadas e integrales de funciones vectoriales. [Familiarizarse]</li> <li>Determinar vectores tangente, normal y binormal para una curva. [Usar]</li> <li>Calcular curvatura y torsión usando fórmulas de Frenet. [Evaluar]</li> <li>Parametrizar curvas usando longitud de arco. [Evaluar]</li> </ul>
Lecturas : [Apostol1997], [MarsdenTromba]	

Unidad 2: Funciones Reales de una Variable Vectorial (18 horas)	
Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Funciones de varias variables: gráficas y curvas de nivel.</li> <li>Límites y continuidad en múltiples variables.</li> <li>Derivadas parciales y direcionales.</li> <li>Vector gradiente y diferencial.</li> <li>Plano tangente y recta normal.</li> <li>Teorema de la función implícita.</li> <li>Máximos y mínimos multivariados.</li> <li>Teorema de Taylor para varias variables.</li> </ul>	<ul style="list-style-type: none"> <li>Calcular derivadas parciales y direcionales. [Familiarizarse]</li> <li>Determinar planos tangentes y aproximaciones lineales. [Usar]</li> <li>Optimizar funciones de varias variables. [Evaluar]</li> <li>Aplicar el teorema de la función implícita. [Evaluar]</li> </ul>
Lecturas : [Apostol1997], [MarsdenTromba]	

Unidad 3: Funciones Vectoriales de Variables Vectoriales (12 horas)	
Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Transformaciones de <math>\mathbb{R}^n</math> a <math>\mathbb{R}^m</math>.</li> <li>Derivada y diferencial de transformaciones.</li> <li>Matriz Jacobiana y Jacobiano.</li> <li>Teorema de la función inversa.</li> <li>Teorema de la función implícita.</li> <li>Transformaciones en coordenadas polares, cilíndricas y esféricas.</li> </ul>	<ul style="list-style-type: none"> <li>Calcular la matriz Jacobiana de una transformación. [Familiarizarse]</li> <li>Aplicar el teorema de la función inversa. [Usar]</li> <li>Realizar cambios de variable usando transformaciones coordenadas. [Evaluar]</li> <li>Utilizar el teorema de la función implícita en contextos multivariados. [Evaluar]</li> </ul>
Lecturas : [Apostol1997], [MarsdenTromba]	

<b>Unidad 4: Integrales Múltiples (18 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Integrales dobles sobre rectángulos y regiones generales.</li> <li>• Integrales iteradas y cambio de variables.</li> <li>• Integrales dobles en coordenadas polares.</li> <li>• Integrales triples y aplicaciones.</li> <li>• Integrales triples en coordenadas cilíndricas y esféricas.</li> <li>• Cálculo de áreas, volúmenes y centros de masa.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluar integrales dobles y triples. [Familiarizarse]</li> <li>• Aplicar cambios de variable en integrales múltiples. [Usar]</li> <li>• Calcular volúmenes y áreas usando integrales múltiples. [Evaluar]</li> <li>• Utilizar coordenadas cilíndricas y esféricas en integrales triples. [Evaluar]</li> </ul>

Lecturas : [Apostol1997], [MarsdenTromba]

<b>Unidad 5: Integrales de Línea (10 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Integral de línea de primer y segundo tipo.</li> <li>• Campos vectoriales y trabajo.</li> <li>• Teorema de Green.</li> <li>• Independencia del camino.</li> <li>• Campos conservativos y función potencial.</li> <li>• Teoremas fundamentales del cálculo para integrales de línea.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular integrales de línea sobre curvas. [Familiarizarse]</li> <li>• Aplicar el teorema de Green en el plano. [Usar]</li> <li>• Determinar si un campo vectorial es conservativo. [Evaluar]</li> <li>• Construir funciones potenciales para campos conservativos. [Evaluar]</li> </ul>

Lecturas : [Apostol1997], [MarsdenTromba]

<b>Unidad 6: Integrales de Superficie (12 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C08,AG-C11</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Superficies paramétricas y área de superficie.</li> <li>• Integral de superficie de primer tipo (campos escalares).</li> <li>• Integral de superficie de segundo tipo (campos vectoriales).</li> <li>• Teorema de Stokes.</li> <li>• Teorema de Gauss (Divergencia).</li> <li>• Rotacional y divergencia de campos vectoriales.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular integrales de superficie. [Familiarizarse]</li> <li>• Aplicar el teorema de Stokes a superficies orientadas. [Usar]</li> <li>• Utilizar el teorema de la divergencia en problemas 3D. [Evaluar]</li> <li>• Calcular rotacional y divergencia de campos vectoriales. [Evaluar]</li> </ul>

Lecturas : [Apostol1997], [MarsdenTromba]

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## **10. BIBLIOGRAFÍA BÁSICA**

## 1. CURSO

ST251FCCS. Estadística y Probabilidades (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	ST251FCCS. Estadística y Probabilidades
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BMA102. Cálculo Diferencial. (1 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El cálculo de probabilidades y la estadística son fundamentales en la ciencia de la computación para el análisis de algoritmos, el modelado de sistemas, la toma de decisiones en situaciones de incertidumbre y el análisis de datos. Este curso integra teoría de probabilidades con métodos estadísticos, enfocándose en aplicaciones computacionales como análisis de algoritmos, machine learning, modelado de sistemas y ciencia de datos.

## 5. OBJETIVOS

- Comprender los fundamentos del cálculo de probabilidades y la estadística inferencial.
- Aplicar distribuciones de probabilidad y métodos estadísticos para resolver problemas en computación.
- Desarrollar habilidades para modelar sistemas estocásticos y realizar inferencia sobre datos.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)
- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

## 7. TEMAS

**Unidad 1: Fundamentos de Probabilidad (6 horas)****Resultados esperados: 1,6,AG-C07**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Espacios de probabilidad y axiomas</li> <li>• Operaciones con eventos: unión, intersección, complemento</li> <li>• Técnicas de conteo: permutaciones y combinaciones</li> <li>• Probabilidad condicional e independencia</li> <li>• Teorema de Bayes y aplicaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Definir espacios muestrales y aplicar axiomas de probabilidad. [Familiarizarse]</li> <li>• Utilizar técnicas de conteo para calcular probabilidades. [Usar]</li> <li>• Aplicar el teorema de Bayes en problemas de clasificación. [Evaluar]</li> </ul>

**Lecturas :** [Ros14], [Dev16]**Unidad 2: Variables Aleatorias Discretas (6 horas)****Resultados esperados: 1,6,AG-C07**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Variables aleatorias discretas</li> <li>• Función de masa de probabilidad (PMF) y distribución acumulativa (CDF)</li> <li>• Esperanza, varianza y momentos</li> <li>• Distribuciones: Bernoulli, Binomial, Geométrica, Poisson</li> <li>• Aplicaciones en modelado de sistemas computacionales</li> </ul>	<ul style="list-style-type: none"> <li>• Definir y caracterizar variables aleatorias discretas. [Familiarizarse]</li> <li>• Calcular esperanza y varianza para diferentes distribuciones. [Usar]</li> <li>• Aplicar distribuciones discretas en modelado de redes y sistemas. [Evaluar]</li> </ul>

**Lecturas :** [Ros14], [Dev16]**Unidad 3: Variables Aleatorias Continuas (6 horas)****Resultados esperados: 1,6,AG-C07**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Variables aleatorias continuas</li> <li>• Función de densidad de probabilidad (PDF) y CDF</li> <li>• Transformaciones de variables aleatorias</li> <li>• Distribuciones: Uniforme, Exponencial, Normal</li> <li>• Aplicaciones en simulación de sistemas y colas</li> </ul>	<ul style="list-style-type: none"> <li>• Diferenciar entre variables discretas y continuas. [Familiarizarse]</li> <li>• Calcular probabilidades usando funciones de densidad. [Usar]</li> <li>• Modelar tiempos de servicio y llegada usando distribuciones continuas. [Evaluar]</li> </ul>

**Lecturas :** [Ros14], [Dev16]

<b>Unidad 4: Distribuciones Multivariadas y Dependencia (6 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Distribuciones conjuntas y marginales</li> <li>• Distribuciones condicionales</li> <li>• Covarianza y correlación</li> <li>• Independencia de variables aleatorias</li> <li>• Aplicaciones en análisis de datos multidimensionales</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular distribuciones conjuntas y marginales. [Familiarizarse]</li> <li>• Medir dependencia mediante covarianza y correlación. [Usar]</li> <li>• Analizar relaciones entre variables en conjuntos de datos. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Ros14], [Dev16]	
<b>Unidad 5: Teoremas Límite y Aproximaciones (6 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Ley de los Grandes Números</li> <li>• Teorema del Límite Central</li> <li>• Momentos y funciones generadoras</li> <li>• Desigualdades probabilísticas (Chernoff, Markov, Chebyshev)</li> <li>• Aplicaciones en big data y análisis de algoritmos</li> </ul>	<ul style="list-style-type: none"> <li>• Enunciar e interpretar teoremas límite. [Familiarizarse]</li> <li>• Aplicar el TLC en aproximaciones de distribuciones. [Usar]</li> <li>• Utilizar desigualdades en análisis de cotas para algoritmos. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Ros14], [Dev16]	
<b>Unidad 6: Inferencia Bayesiana (6 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Actualización bayesiana con priors discretos/continuos</li> <li>• Estimación Máxima a Posteriori (MAP)</li> <li>• Intervalos creíbles bayesianos</li> <li>• Priors conjugados</li> <li>• Aplicaciones en clasificadores Naive Bayes y ML</li> </ul>	<ul style="list-style-type: none"> <li>• Diferenciar entre enfoque frecuentista y bayesiano. [Familiarizarse]</li> <li>• Realizar actualización bayesiana de creencias. [Usar]</li> <li>• Implementar estimación MAP en problemas de machine learning. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Ros14], [Dev16]	
<b>Unidad 7: Inferencia Estadística (6 horas)</b>	
<b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Estimación de parámetros: MLE (Máxima Verosimilitud)</li> <li>• Pruebas de hipótesis: z-test, t-test, chi-cuadrado</li> <li>• Intervalos de confianza</li> <li>• Bootstrapping y métodos de remuestreo</li> <li>• Aplicaciones en validación de modelos y data science</li> </ul>	<ul style="list-style-type: none"> <li>• Estimar parámetros usando métodos de máxima verosimilitud. [Familiarizarse]</li> <li>• Realizar pruebas de hipótesis para validar supuestos. [Usar]</li> <li>• Construir intervalos de confianza y aplicar bootstrapping. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Dev16]	

<b>Unidad 8: Regresión y Aplicaciones Avanzadas (6 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Regresión lineal simple</li> <li>• Ajuste por mínimos cuadrados</li> <li>• Procesos de Poisson</li> <li>• Simulación Monte Carlo</li> <li>• Introducción a cadenas de Markov</li> <li>• Aplicaciones en predicción y modelado de sistemas</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar modelos de regresión lineal. [Familiarizarse]</li> <li>• Utilizar simulación Monte Carlo para resolver problemas complejos. [Usar]</li> <li>• Aplicar procesos estocásticos en modelado de sistemas reales. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Dev16]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Ros14] Sheldon M. Ross. *A First Course in Probability*. Pearson, 2014.

[Dev16] Jay L. Devore. *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 2016.

## **1. CURSO**

ID103. Inglés III (Obligatorio)

## **2. INFORMACIÓN GENERAL**

<b>2.1 Curso</b>	:	ID103. Inglés III
<b>2.2 Semestre</b>	:	3 <sup>er</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BEI102. Inglés II. (2 <sup>do</sup> Sem)

## **3. PROFESORES**

Atención previa coordinación con el profesor

## **4. INTRODUCCIÓN AL CURSO**

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

## **5. OBJETIVOS**

- Formar en el alumno de capacidad de comprender y retener una conversación.
- Brindar técnicas de ilación de ideas.

## **6. RESULTADOS DEL ESTUDIANTE**

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

## **7. TEMAS**

<b>Unidad 1: Getting to know you! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempos Presente, Pasado y Futuro.</li> <li>• Oraciones Interrogativas con Wh-.</li> <li>• Palabras con más de un significado.</li> <li>• Partes de la oración.</li> <li>• Expresiones para tiempo libre.</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los tiempos presente, pasado y futuro es capaz de expresar una mayor cantidad de acciones en forma de oraciones. Además es capaz de expresar ideas en forma de preguntas. Asume la idea de palabras con más de un significado. Utiliza expresiones sociales en situaciones de entretenimiento.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 2: The way we live! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempo Presente Simple.</li> <li>• Tiempo Presente Continuo.</li> <li>• Colocaciones.</li> <li>• Vocabulario de países del mundo.</li> <li>• Expresiones de enojo.</li> <li>• Conectores.</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas del mismo y las aplican adecuadamente. Describen países acucliosamente. Asumen expresiones para demostrar interés. Utilizan conectores para unir ideas varias.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 3: It all went wrong! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempo Pasado Simple.</li> <li>• Tiempo Pasado Continuo.</li> <li>• Verbos Irregulares.</li> <li>• Expresiones de Tiempo.</li> <li>• Conectores de tiempo.</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la tercera unidad, los alumnos habiendo reconocido las características de los tiempos en pasado los utilizan adecuadamente. Utilizan prefijos y sufijos para crear y reconocer nuevas palabras. Describen tiempo en forma amplia. Utilizarán conjunciones para unir ideas tipo.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 4: Lets go shopping! (0 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Expresiones de Cantidad Indefinida</li> <li>• Oraciones Afirmativas, Negativas y Preguntas</li> <li>• Uso de Artículos</li> <li>• Precios de productos</li> <li>• Llenado de formatos y encuestas</li> <li>• Expresiones para ir de compras</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la cuarta primera unidad, los alumnos habiendo identificado la idea de cantidad expresan diversas situaciones que la involucran. Reconocen y aplican artículos a sustantivos. Asumen la idea de ir de compras con la ayuda de expresiones. Expresan precios e ideas de dinero. Llenan formatos varios. Expresan actitudes.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 5: What do you want to do? (0 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Patrones Verbales I.</li> <li>• Intenciones Futuras.</li> <li>• Verbos de Percepción.</li> <li>• Vocabulario de sentimientos.</li> <li>• Expresiones de Planes y Ambiciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la quinta unidad, los alumnos, a partir de la comprensión de la idea de patrones verbales elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar intenciones futuras. Adquirirán vocabulario para describir sentimientos. Se presentará expresiones para describir planes y ambiciones.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 6: The best in the world! (0 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Whats it like?.</li> <li>• Adjetivos.</li> <li>• Comparativos y Superlativos.</li> <li>• Sinónimos y Antónimos.</li> <li>• Indicaciones de Dirección.</li> <li>• Lecturas.</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos del uso de adjetivos, estructuran oraciones con diversas formas de los mismos en contextos adecuados. Enfatizan la diferencia entre tipos de ciudades y pueblos y estilos de vida. Utilizan expresiones indicación de direcciones.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 7: Fame! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Presente Perfecto y Pasado Simple</li> <li>• Expresiones for, ever, since</li> <li>• Adverbios</li> <li>• Expresiones que vienen en pares</li> <li>• Respuestas cortas</li> <li>• Celebridades</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la séptima unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo presente perfecto y lo diferencian del pasado simple. Enfatizan la diferencia entre formas de adjetivos. Describen ideas de la música. Utilizan expresiones para dar respuestas cortas. Asumen la idea de dar explicaciones extra de los elementos de una oración.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

[Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.

## Capítulo 4

# Cuarto Semestre

## 1. CURSO

CS210-CS2023. Algoritmos y Estructuras de Datos (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS210-CS2023. Algoritmos y Estructuras de Datos
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS113. Programación Orientada a Objetos II. (3 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El fundamento teórico de todas las ramas de la informática descansa sobre los algoritmos y estructuras de datos, este curso brindará a los participantes una introducción a estos temas, formando así una base que servirá para los siguientes cursos en la carrera.

## 5. OBJETIVOS

- Hacer que el alumno entienda la importancia de los algoritmos para la solución de problemas.
- Introducir al alumno hacia el campo de la aplicación de las estructuras de datos.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Grafos (12 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Concepto de Grafos.</li> <li>• Grafos Dirigidos y Grafos no Dirigidos.</li> <li>• Utilización de los Grafos.</li> <li>• Medida de la Eficiencia. En tiempo y espacio.</li> <li>• Matrices de Adyacencia.</li> <li>• Matrices de Adyacencia etiquetada.</li> <li>• Listas de Adyacencia.</li> <li>• Implementación de Grafos usando Matrices de Adyacencia.</li> <li>• Implementación de Grafos usando Listas de Adyacencia.</li> <li>• Inserción, Búsqueda y Eliminación de nodos y aristas.</li> <li>• Algoritmos de búsqueda en grafos.</li> </ul>	<ul style="list-style-type: none"> <li>• Adquirir destreza para realizar una implementación correcta. [Usar]</li> <li>• Desarrollar los conocimientos para decidir cuando es mejor usar una técnica de implementación que otra. [Usar]</li> </ul>
<b>Lecturas :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

<b>Unidad 2: Matrices Esparzas (8 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Conceptos Iniciales.</li> <li>• Matrices poco densas</li> <li>• Medida de la Eficiencia en Tiempo y en Espacio</li> <li>• Creación de la matriz esparza estática vs Dinámicas.</li> <li>• Métodos de inserción, búsqueda y eliminación</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender el uso y implementación de matrices esparzas. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

<b>Unidad 3: Árboles Equilibrados (16 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Árboles AVL.</li> <li>• Medida de la Eficiencia.</li> <li>• Rotaciones Simples y Compuestas</li> <li>• Inserción, Eliminación y Búsqueda.</li> <li>• Árboles B , B+ B* y Patricia.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender las funciones básicas de estas estructuras complejas con el fin de adquirir la capacidad para su implementación. [Evaluar]</li> </ul>
<b>Lecturas :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## **10. BIBLIOGRAFÍA BÁSICA**

- [Knu97] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. 3rd. Addison-Wesley Professional, 1997.
- [Knu98] Donald E. Knuth. *The art of computer programming, volume 3:Sorting and searching*. 2nd. Addison-Wesley Professional, 1998.
- [Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms*. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.
- [Fag+14] José Fager et al. *Estructura de datos*. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.

## 1. CURSO

CS211-CS2023. Teoría de la Computación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS211-CS2023. Teoría de la Computación
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS1D1. Estructuras Discretas. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso hace énfasis en los lenguajes formales, modelos de computación y computabilidad, además de incluir fundamentos de la complejidad computacional y de los problemas NP completos.

## 5. OBJETIVOS

- Que el alumno aprenda los conceptos fundamentales de la teoría de lenguajes formales.

## 6. RESULTADOS DEL ESTUDIANTE

- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

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

## 7. TEMAS

**Unidad 1: Autómatas y Lenguajes (20 horas)**

**Resultados esperados: 1**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Conjuntos y Lenguajes: <ul style="list-style-type: none"> <li>– Lenguajes Regulares.</li> <li>– Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs)</li> <li>– Autómata finito no determinístico (Nondeterministic Finite Automata NFAs)</li> <li>– Equivalencia de DFAs y NFAs.</li> <li>– Revisión de expresiones regulares; su equivalencia con autómatas finitos.</li> <li>– Propiedades de cierre.</li> <li>– Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos.</li> </ul> </li> <li>• Gramáticas libres de contexto.</li> <li>• Lenguajes libres de contexto: <ul style="list-style-type: none"> <li>– Autómatas de pila (Push-down automata (PDAs))</li> <li>– Relación entre PDA y gramáticas libres de contexto.</li> <li>– Propiedades de los lenguajes libres de contexto.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar]</li> <li>• Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar]</li> <li>• Discute el concepto de máquina de estado finito [Evaluar]</li> <li>• Diseñe una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar]</li> <li>• Genere una expresión regular para representar un lenguaje específico [Evaluar]</li> <li>• Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar]</li> </ul>

**Lecturas :** [Sip12], [HU08], [Bro93]

<b>Unidad 2: Teoría de la Computabilidad (20 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Problema de la parada.</li> <li>• Introducción a las clases P y NP y al problema P vs. NP.</li> <li>• Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos.</li> <li>• Máquinas de Turing, o un modelo formal equivalente de computación universal.</li> <li>• Máquinas de Turing no determinísticas.</li> <li>• Jerarquía de Chomsky.</li> <li>• La tesis de Church-Turing.</li> <li>• Computabilidad.</li> <li>• Teorema de Rice.</li> <li>• Ejemplos de funciones no computables.</li> <li>• Implicaciones de la no-computabilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique porque el problema de la parada no tiene solución algorítmica [Evaluar]</li> <li>• Define las clases P y NP [Evaluar]</li> <li>• Explique el significado de NP-Compleitud [Evaluar]</li> <li>• Explica la tesis de Church-Turing y su importancia [Familiarizarse]</li> <li>• Explica el teorema de Rice y su importancia [Familiarizarse]</li> <li>• Da ejemplos de funciones no computables [Familiarizarse]</li> <li>• Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él</li> </ul>

**Lecturas :** [Sip12], [Kel95]

<b>Unidad 3: Teoría de la Complejidad (20 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Revisión de las clases P y NP; introducir espacio P y EXP.</li> <li>• Jerarquía polinomial.</li> <li>• NP completitud (Teorema de Cook).</li> <li>• Problemas NP completos clásicos.</li> <li>• Técnicas de reducción.</li> </ul>	<ul style="list-style-type: none"> <li>• Define la clase P-Space y su relación con la clase EXP [Evaluar]</li> <li>• Define la clase P-Space y su relación con la clase EXP [Evaluar]</li> <li>• Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar]</li> <li>• Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar]</li> <li>• Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar]</li> </ul>

**Lecturas :** [Sip12], [Kel95], [HU08]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### 10. BIBLIOGRAFÍA BÁSICA

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [Kel95] Dean Kelley. *Teoría de Autómatas y Lenguajes Formales*. Prentice Hall, 1995.
- [HU08] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Educacion, 2008.
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation (third edition)*. Publisher: Cengage Learning, 2012.

## 1. CURSO

CS221-CS2023. Arquitectura de Computadores (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS221-CS2023. Arquitectura de Computadores
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS1D1. Estructuras Discretas. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Es necesario que el profesional en Ciencia de la Computación tenga sólido conocimiento de la organización y funcionamiento de los diversos sistema de cómputo actuales en los cuales gira el entorno de programación. Con ello también sabrá establecer los alcances y límites de las aplicaciones que se desarrollen de acuerdo a la plataforma siendo usada.

Se tratarán los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

## 5. OBJETIVOS

- Este curso tiene como propósito ofrecer al estudiante una base sólida de la evolución de las arquitecturas de computadores y los factores que influenciaron en el diseño de los elementos de *hardware* y *software* en sistemas de computación actuales.
- Garantizar la comprensión de cómo es el *hardware* en sí y cómo interactúan *hardware* y *software* en un sistema de cómputo actual.
- Tratar los siguientes temas: componentes de lógica digital básicos en un sistema de computación, diseño de conjuntos de instrucciones, microarquitectura del procesador y ejecución en *pipelining*, organización de la memoria: caché y memoria virtual, protección y compartición, sistema I/O e interrupciones, arquitecturas super escalares y ejecución fuera de orden, computadoras vectoriales, arquitecturas para *multithreading*, multiprocesadores simétricos, modelo de memoria y sincronización, sistemas integrados y computadores en paralelo.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

**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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Lógica digital y sistemas digitales (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Revisión e historia de la Arquitectura de Computadores.</li><li>Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial.</li><li>Multiples representaciones / Capas de interpretación (El hardware es solo otra capa)</li><li>Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales.</li><li>Registrar transferencia notación / Hardware lenguage descriptivo (Verilog/VHDL)</li><li>Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/poder)</li></ul>	<ul style="list-style-type: none"><li>Describir el avance paulatino de los componentes de la tecnología de computación, desde los tubos de vacío hasta VLSI, desde las arquitecturas mainframe a las arquitecturas en escala warehouse [Familiarizarse]</li><li>Comprender que la tendencia de las arquitecturas modernas de computadores es hacia núcleos múltiples y que el paralelismo es inherente en todos los sistemas de hardware [Familiarizarse]</li><li>Explicar las implicancias de los límites de potencia para mejoras adicionales en el rendimiento de los procesadores y también en el aprovechamiento del paralelismo [Familiarizarse]</li><li>Relacionar las varias representaciones equivalentes de la funcionalidad de un computador, incluyendo expresiones y puertas lógicas, y ser capaces de utilizar expresiones matemáticas para describir las funciones de circuitos combinacionales y secuenciales sencillos [Familiarizarse]</li><li>Diseñar los componentes básicos de construcción de un computador: unidad aritmético lógica (a nivel de puertas lógicas), unidad central de procesamiento (a nivel de registros de transferencia), memoria (a nivel de registros de transferencia) [Usar]</li><li>Usar herramientas CAD para capturar, sistetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Usar]</li><li>Evaluar el comportamiento de un diagrama de tiempos y funcional de un procesador simple implementado a nivel de circuitos lógicos [Evaluar]</li></ul>
<b>Lecturas :</b> [PH14], [Sta14], [Par05]	

<b>Unidad 2: Organización de la Máquina a Nivel Ensamblador (8 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Organización Básica de la Máquina de Von Neumann.</li> <li>• Unidad de Control.</li> <li>• Paquetes de instrucciones y tipos (manipulación de información, control, I/O)</li> <li>• Assembler / Programación en Lenguaje de Máquina.</li> <li>• Formato de instrucciones.</li> <li>• Modos de direccionamiento.</li> <li>• Llamada a subrutinas y mecanismos de retorno.</li> <li>• I/O e Interrupciones.</li> <li>• Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.</li> <li>• Multiprocesadores de memoria compartida / organización multinúcleo.</li> <li>• Introducción a SIMD vs. MIMD y Taxonomía de Flynn.</li> </ul>	
<b>Lecturas :</b> [PH14], [Sta14], [Par05]	

<b>Unidad 3: Organización funcional (8 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Implementación de rutas de datos simples, incluyendo la canalización de instrucciones, detección de riesgos y la resolución.</li> <li>• Control de unidades: Realización Cableada vs Realización Microprogramada.</li> <li>• Instrucción (Pipelining)</li> <li>• Introducción al paralelismo al nivel de instrucción (PNI)</li> </ul>	
<b>Lecturas :</b> [PH14], [Sta14], [Par05]	

<b>Unidad 4: Mejoras de rendimiento (8 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Arquitectura superescalar.</li> <li>• Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden.</li> <li>• Prefetching.</li> <li>• Procesadores vectoriales y GPU's</li> <li>• Soporte de hardware para multiprocesamiento.</li> <li>• Escalabilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las arquitecturas superescalares y sus ventajas [Familiarizarse]</li> <li>• Explicar el concepto de predicción de bifurcaciones y su utilidad [Familiarizarse]</li> <li>• Caracterizar los costos y beneficios de la precarga prefetching [Familiarizarse]</li> <li>• Explicar la ejecución especulativa e identifique las condiciones que la justifican [Familiarizarse]</li> <li>• Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el maximo beneficio de estas [Familiarizarse]</li> <li>• Describir la importancia de la escalabilidad en el rendimiento [Familiarizarse]</li> </ul>

**Lecturas :** [Par05], [Par02], [PH14], [Don06], [Joh91]

<b>Unidad 5: Organización y Arquitectura del Sistema de Memoria (8 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Sistemas de Almacenamiento y su Tecnología.</li> <li>• Jerarquía de Memoria: importancia de la localización temporal y espacial.</li> <li>• Organización y Operaciones de la Memoria Principal.</li> <li>• Latencia, ciclos de tiempo, ancho de banda e intercalación.</li> <li>• Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento)</li> <li>• Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core.</li> <li>• Memoria virtual (tabla de página, TLB)</li> <li>• Manejo de Errores y confiabilidad.</li> <li>• Error de codificación, compresión de datos y la integridad de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnetico) y su relación costo beneficio [Familiarizarse]</li> <li>• Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse]</li> <li>• Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse]</li> <li>• Describa los principios de la administración de memoria [Familiarizarse]</li> <li>• Explique el funcionamiento de un sistema con gestión de memoria virtual [Familiarizarse]</li> <li>• Calcule el tiempo de acceso promedio a memoria bajo varias configuraciones de caché y memoria y para diversas combinaciones de instrucciones y referencias a datos [Usar]</li> </ul>

**Lecturas :** [PH14], [Sta14], [Par05]

<b>Unidad 6: Interfaz y comunicación (8 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O.</li> <li>• Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado.</li> <li>• Almacenamiento externo, organización física y discos.</li> <li>• Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA).</li> <li>• Introducción a Redes: comunicación de redes como otra capa de acceso remoto.</li> <li>• Soporte Multimedia.</li> <li>• Arquitecturas RAID.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse]</li> <li>• Identificar diversos tipos de buses en un sistema computacional [Familiarizarse]</li> <li>• Describir el acceso a datos desde una unidad de disco magnético [Familiarizarse]</li> <li>• Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones commutadas versus ruteadas [Familiarizarse]</li> <li>• Identificar las interfaces entre capas necesarios para el acceso y presentación multimedia, desde la captura de la imagen en almacenamiento remoto, a través del transporte por una red de comunicaciones, hasta la puesta en la memoria local y la presentación final en una pantalla gráfica [Familiarizarse]</li> <li>• Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]</li> </ul>

**Lecturas :** [PH14], [Sta14], [Par05]

<b>Unidad 7: Multiprocesamiento y arquitecturas alternativas (8 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Ley potencial.</li> <li>• Ejemplos de juego de instrucciones y arquitecturas SIMD y MIMD.</li> <li>• Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar)</li> <li>• Sistemas de memoria de multiprocesador compartido y consistencia de memoria.</li> <li>• Coherencia de cache multiprocesador.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Familiarizarse]</li> <li>• Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse]</li> <li>• Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Familiarizarse]</li> <li>• Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse]</li> <li>• Describir las diferencias entre conectores eléctricos en paralelo backplane, interconexión memoria procesador y memoria remota vía red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Familiarizarse]</li> </ul>

**Lecturas :** [PH14], [Sta14], [Par05]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Joh91] M. Johnson. *Superscalar microprocessor design*. Prentice Hall series in innovative technology. Prentice Hall, 1991.
- [Par02] Behrooz Parhami. *Introduction to parallel processing: algorithms and architectures*. Plenum series in computer science. Plenum Press, 2002.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005.
- [Don06] J. Dongarra. “Trends in high performance computing: a historical overview and examination of future developments”. In: *Circuits and Devices Magazine, IEEE* 22.1 (2006), pp. 22–27. DOI: 10.1109/MCD.2006.1598076.
- [PH14] David A. Patterson and John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 5th ed. San Mateo, CA: Morgan Kaufman, 2014.
- [Sta14] William Stallings. *Computer Organization and Architecture: Designing for Performance*. 9th. Upper Saddle River, NJ: Prentice Hall, 2014.

## 1. CURSO

CS271-CS2023. Bases de Datos I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS271-CS2023. Bases de Datos I
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
		• CS112. Programación Orientada a Objetos I. (2 <sup>do</sup> Sem)
<b>2.8 Prerrequisitos</b>	:	• CS1D1. Estructuras Discretas. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La gestión de la información (IM) juega un rol principal en casi todas las áreas donde los computadores son usados. Esta área incluye la captura, digitalización, representación, organización, transformación y presentación de información; algoritmos para mejorar la eficiencia y efectividad del acceso y actualización de información almacenada, modelamiento de datos y abstracción, y técnicas de almacenamiento de archivos físicos.

Este también abarca la seguridad de la información, privacidad, integridad y protección en un ambiente compartido. Los estudiantes necesitan ser capaces de desarrollar modelos de datos conceptuales y físicos, determinar qué métodos de (IM) y técnicas son apropiados para un problema dado, y ser capaces de seleccionar e implementar una apropiada solución de IM que refleje todas las restricciones aplicables, incluyendo escalabilidad y usabilidad.

## 5. OBJETIVOS

- Que el alumno aprenda a representar información en una base de datos priorizando la eficiencia en la recuperación de la misma
- Que el alumno aprenda los conceptos fundamentales de gestión de bases de datos. Esto incluye aspectos de diseño de bases de datos, lenguajes de bases de datos y realización de bases de datos
- Discutir el modelo de bases de datos con base en el álgebra relacional, cálculo relacional y en el estudio de sentencias SQL.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

### Unidad 1: Sistemas de Bases de Datos (14 horas)

Resultados esperados: 2,6

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Enfoque y Evolución de Sistemas de Bases de Datos.</li><li>• Componentes del Sistema de Bases de Datos.</li><li>• Diseño de las funciones principales de un DBMS.</li><li>• Arquitectura de base de datos e independencia de datos.</li><li>• Uso de un lenguaje de consulta declarativa.</li><li>• Sistemas de apoyo a contenido estructurado y / o corriente.</li><li>• Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce).</li></ul>	<ul style="list-style-type: none"><li>• Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar]</li><li>• Describe los componentes de un sistema de bases datos y da ejemplos de su uso [Usar]</li><li>• Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Usar]</li><li>• Describe los componentes de un sistema de bases datos y da ejemplos de su uso [Usar]</li><li>• Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Usar]</li><li>• Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar]</li><li>• Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar]</li><li>• Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Usar]</li><li>• Describe los enfoques principales para almacenar y procesar grandes volúmenes de datos [Usar]</li></ul>

Lecturas : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]

<b>Unidad 2: Modelado de datos (14 horas)</b> <b>Resultados esperados: 2,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Modelado de datos</li> <li>• Modelos conceptuales (e.g., entidad-relación, diagramas UML)</li> <li>• Modelos de hoja de cálculo</li> <li>• Modelos Relacionales.</li> <li>• Modelos orientados a objetos.</li> <li>• Modelos de datos semi-estructurados (expresados usando DTD o XML Schema, por ejemplo)</li> </ul>	<ul style="list-style-type: none"> <li>• Compare y contrasta modelos apropiados de datos, incluyendo estructuras sus estructuras internas, para diversos tipos de datos [Usar]</li> <li>• Describe los conceptos en notación de modelos (ejm. Diagramas Entidad-Relación o UML) y cómo deben de ser usados [Usar]</li> <li>• Define la terminología fundamental a ser usada en un modelo relacional de datos [Usar]</li> <li>• Describe los principios básicos del modelo relacional de datos [Usar]</li> <li>• Aplica los conceptos de modelado y la notación de un modelo relacional de datos [Usar]</li> <li>• Describe los principios básicos del modelo relacional de datos [Usar]</li> <li>• Describe los principios básicos del modelo relacional de datos [Usar]</li> <li>• Da una semi estructura equivalente (ejm. en DTD o Esquema XML) para un esquema relacional dado [Usar]</li> </ul>	

Lecturas : [SW04], [EN04], [KS02]

<b>Unidad 3: Indexación (4 horas)</b> <b>Resultados esperados: 2,6</b>	<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• El impacto de indices en el rendimiento de consultas.</li> <li>• La estructura basica de un indice.</li> <li>• Mantener un buffer de datos en memoria.</li> <li>• Creando indices con SQL.</li> <li>• Indexando texto.</li> <li>• Indexando la web (e.g., web crawling)</li> </ul>	<ul style="list-style-type: none"> <li>• Generar un archivo índice para una colección de recursos [Usar]</li> <li>• Explicar la función de un índice invertido en la localización de un documento en una colección [Usar]</li> <li>• Explicar cómo rechazar y detener palabras que afectan a la indexación [Usar]</li> <li>• Identificar los índices adecuados para determinado el esquema relacional y el conjunto de consultas [Usar]</li> <li>• Estimar el tiempo para recuperar información, cuando son usados los índices comparado con cuando no son usados [Usar]</li> <li>• Describir los desafíos claves en el rastreo web, por ejemplo, la detección de documentos duplicados, la determinación de la frontera de rastreo [Usar]</li> </ul>	

Lecturas : [WM01], [RG03], [ER15], [CJ11], [KS02]

**Unidad 4: Bases de Datos Relacionales (14 horas)**

**Resultados esperados: 2,6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Mapeo de esquemas conceptuales a esquemas relacionales.</li> <li>• Entidad y integridad referencial.</li> <li>• Algebra relacional y calculo relacional.</li> <li>• Diseño de bases de datos relacionales.</li> <li>• Dependencia funcional.</li> <li>• Descomposición de un esquema.</li> <li>• Llaves candidatas, SuperLlaves y cierre de un conjunto de atributos.</li> <li>• Formas Normales (BCNF)</li> <li>• Dependencias multi-valoradas (4NF)</li> <li>• Uniendo dependencias (PJNF, 5NF)</li> <li>• Teoría de la representación.</li> </ul>	<ul style="list-style-type: none"> <li>• Prepara un esquema relacional de un modelo conceptual desarrollado usando el modelo entidad-relación [Usar]</li> <li>• Explica y demuestra los conceptos de restricciones de integridad de la entidad e integridad referencial (incluyendo la definición del concepto de clave foránea) [Usar]</li> <li>• Demuestra el uso de las operaciones de álgebra relacional de la teoría matemática de conjuntos (unión, intersección, diferencia, y producto Cartesiano) y de las operaciones de álgebra relacional desarrolladas específicamente para las bases de datos relacionales (selección (restringida), proyección, unión y división) [Usar]</li> <li>• Escribe consultas en cálculo relacional de tuplas [Usar]</li> <li>• Escribe consultas en cálculo relacional de tuplas [Usar]</li> <li>• Determina la dependencia funcional entre dos o más atributos que son subconjunto de una relación [Usar]</li> <li>• Conecta restricciones expresadas como clave primaria y foránea, con dependencias funcionales [Usar]</li> <li>• Calcula la cerradura de un conjunto de atributos dado dependencias funcionales [Usar]</li> <li>• Determina si un conjunto de atributos forma una superclave y/o una clave candidata de una relación dada dependencias funcionales [Usar]</li> <li>• Evalua una descomposición propuesta, a fin de determinar si tiene una unión sin pérdidas o preservación de dependencias [Usar]</li> <li>• Describe las propiedades de la FNBC, FNUP (forma normal unión de proyecto), 5FN [Usar]</li> <li>• Explica el impacto de la normalización en la eficacia de las operaciones de una base de datos especialmente en la optimización de consultas [Usar]</li> <li>• Describe que es una dependencia de multi valor y cual es el tipo de restricciones que especifica [Usar]</li> </ul>

**Lecturas :** [WM01], [RG03], [ER15], [CJ11], [KS02]

<b>Unidad 5: Lenguajes de Consulta (12 horas)</b>	
<b>Resultados esperados: 2,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Visión general de lenguajes de base de datos.</li> <li>SQL (definición de datos, formulación de consultas, sublenguaje update, restricciones, integridad)</li> <li>Selecciones</li> <li>Proyecciones</li> <li>Select-project-join</li> <li>Agregaciones y agrupaciones.</li> <li>Subconsultas.</li> <li>Entornos QBE de cuarta generación.</li> <li>Diferentes maneras de invocar las consultas no procedimentales en lenguajes convencionales.</li> <li>Introducción a otros lenguajes importantes de consulta (por ejemplo, XPATH, SPARQL)</li> <li>Procedimientos almacenados.</li> </ul>	<ul style="list-style-type: none"> <li>Crear un esquema relacional de bases de datos en SQL que incorpora restricciones clave y restricciones de integridad de entidad e integridad referencial [Usar]</li> <li>Usar SQL para crear tablas y devuelve (SELECT) la información de una base de datos [Usar]</li> <li>Evaluando un conjunto de estrategias de procesamiento de consultas y selecciona la estrategia óptima [Usar]</li> <li>Crear una consulta no-procedimental al llenar plantillas de relacines para construir un ejemplo del resultado de una consulta requerida [Usar]</li> <li>Adicionar consultas orientadas a objetos en un lenguaje stand-alone como C++ o Java (ejm. SELECT ColMethod() FROM Objeto) [Usar]</li> <li>Escribe un procedimiento almacenado que trata con parámetros y con algo de flujo de control de tal forma que tenga funcionalidad [Usar]</li> </ul>

**Lecturas :** [Die01], [EN04], [Cel05], [KS02]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.
- [WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.
- [KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.
- [RG03] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. 3rd. McGraw-Hill, 2003.
- [EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.
- [RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.
- [SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.
- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.
- [CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.
- [ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases*. 2nd. O'Reilly Media, 2015.

## 1. CURSO

CS401. Metodología de la Investigación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS401. Metodología de la Investigación
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	1 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS100. Introducción a la Ciencia de la Computación. (2 <sup>do</sup> 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 le hará entrega de bibliografía para analizar y a partir de la misma, y de fuentes bibliográficas adicionales (investigadas por el alumno), el alumno deberá ser capaz de construir un artículo del tipo survey del tema asignado.

## 5. OBJETIVOS

- Que el alumno aprenda como se inicia una investigación científica en el área de computación.
- Que el alumno conozca las principales fuentes para obtener bibliografía relevante para trabajos de investigación en el área de computación: Researchindex, IEEE-CS<sup>1</sup>, ACM<sup>2</sup>.
- Que el alumno sea capaz de analizar las propuestas existentes sobre un determinado tópico y relacionarlos de forma coherente en una revisión bibliográfica.
- Que el alumno pueda redactar documentos técnicos en computación utilizando L<sup>A</sup>T<sub>E</sub>X.
- Que el alumno sea capaz de reproducir los resultados ya existentes en un determinado tópico a través de la experimentación.
- Los entregables de este curso son:

**Avance parcial:** Dominio del tema del artículo y bibliografía preliminar en formato de artículo L<sup>A</sup>T<sub>E</sub>X.

**Final:** Entendimiento del artículo del tipo survey, documento concluído donde se contenga, opcionalmente, los resultados experimentales de la(s) técnica(s) estudiada(s).

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

7) Desarrollar principios investigación en el área de computación con niveles de competitividad internacional. (Assessment)

<sup>1</sup><http://www.computer.org>

<sup>2</sup><http://www.acm.org>

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

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

**4)** Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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

<b>Unidad 1: Iniciación científica en el área de computación (48 horas)</b>	
<b>Resultados esperados: 1,3,4,7,AG-C02,AG-C04,AG-C10</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Búsqueda bibliográfica en computación.</li><li>• Redacción de artículos técnicos en computación.</li></ul>	<ul style="list-style-type: none"><li>• Aprender a hacer una investigación correcta en el área de computación[Usar]</li><li>• Conocer las fuentes de bibliografía adecuada para esta área[Usar]</li><li>• Saber redactar un documento de acuerdo con las características que las conferencias de esta área exigen[Usar]</li></ul>
<b>Lecturas :</b> [IEE08], [ACM08], [Cit08]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[ACM08] ACM. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.

## 1. CURSO

MA106FCCS. Métodos Numéricos (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	MA106FCCS. Métodos Numéricos
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BMA103. Cálculo Integral. (2 <sup>do</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Los métodos numéricos son esenciales en la ciencia de la computación para aproximar soluciones a problemas matemáticos que no pueden ser resueltos analíticamente. Este curso proporciona una introducción a los métodos numéricos más comunes, incluyendo la resolución de ecuaciones, interpolación, integración numérica y la solución de ecuaciones diferenciales.

## 5. OBJETIVOS

- Comprender la importancia de los métodos numéricos en la resolución de problemas computacionales.
- Aplicar diferentes métodos numéricos para aproximar soluciones a problemas matemáticos.
- Analizar la precisión y eficiencia de los métodos numéricos utilizados.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)
- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

## 7. TEMAS

<b>Unidad 1: Introducción a los Métodos Numéricos (4 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Representación de números en la computadora.</li> <li>• Errores de redondeo y truncamiento.</li> <li>• Propagación de errores.</li> <li>• Análisis de estabilidad y convergencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar cómo se representan los números en una computadora y las limitaciones de esta representación. [Familiarizarse]</li> <li>• Diferenciar entre errores de redondeo y truncamiento. [Usar]</li> <li>• Analizar cómo se propagan los errores en los cálculos numéricos. [Evaluar]</li> </ul>

**Lecturas :** [CC15], [BF10]

<b>Unidad 2: Solución de Ecuaciones No Lineales (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Método de bisección.</li> <li>• Método de Newton-Raphson.</li> <li>• Método de la secante.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar el método de bisección para encontrar raíces de ecuaciones. [Familiarizarse]</li> <li>• Utilizar el método de Newton-Raphson para aproximar soluciones. [Usar]</li> <li>• Implementar el método de la secante para resolver ecuaciones no lineales. [Evaluar]</li> </ul>

**Lecturas :** [CC15], [BF10]

<b>Unidad 3: Interpolación y Aproximación Polinomial (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Interpolación polinomial de Lagrange.</li> <li>• Interpolación de Newton.</li> <li>• Splines.</li> </ul>	<ul style="list-style-type: none"> <li>• Construir polinomios interpolantes de Lagrange. [Familiarizarse]</li> <li>• Aplicar la interpolación de Newton. [Usar]</li> <li>• Utilizar splines para aproximar funciones. [Evaluar]</li> </ul>

**Lecturas :** [CC15], [BF10]

<b>Unidad 4: Integración Numérica (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Regla del trapecio.</li> <li>• Regla de Simpson.</li> <li>• Cuadratura gaussiana.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar la regla del trapecio para aproximar integrales. [Familiarizarse]</li> <li>• Utilizar la regla de Simpson para calcular integrales numéricamente. [Usar]</li> <li>• Aplicar la cuadratura gaussiana para la integración numérica. [Evaluar]</li> </ul>

**Lecturas :** [CC15], [BF10]

<b>Unidad 5: Solución Numérica de Ecuaciones Diferenciales Ordinarias (8 horas)</b> <b>Resultados esperados: 1,6,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Método de Euler.</li> <li>• Métodos de Runge-Kutta.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar el método de Euler para aproximar soluciones de EDOs. [Familiarizarse]</li> <li>• Implementar métodos de Runge-Kutta para resolver EDOs numéricamente. [Usar]</li> </ul>

Lecturas : [CC15], [BF10]

<b>Unidad 6: Aplicaciones en Computación (12 horas)</b> <b>Resultados esperados: 1,6,AG-C07,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Simulación de sistemas físicos.</li> <li>• Modelado científico.</li> <li>• Aprendizaje automático (ej. optimización de modelos).</li> </ul>	<ul style="list-style-type: none"> <li>• Utilizar métodos numéricos para simular sistemas físicos. [Familiarizarse]</li> <li>• Aplicar métodos numéricos en el modelado científico. [Usar]</li> <li>• Implementar métodos numéricos en algoritmos de aprendizaje automático. [Evaluar]</li> </ul>

Lecturas : [CC15]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[BF10] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. Cengage Learning, 2010.

[CC15] Steven C. Chapra and Raymond P. Canale. *Numerical Methods for Engineers*. McGraw-Hill Education, 2015.

## 1. CURSO

ID104. Inglés IV (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	ID104. Inglés IV
<b>2.2 Semestre</b>	:	4 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BEI201. Inglés III. (3 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde este no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

## 5. OBJETIVOS

- Incrementar el nivel de conversación en diferentes temas, en los alumnos. Así como la capacidad de escribir y leer documentación de todo tipo.
- Llevar al alumno a una expresión más intensa en el dominio del idioma.

## 6. RESULTADOS DEL ESTUDIANTE

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

## 7. TEMAS

<b>Unidad 1: Do and don't! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Auxiliares de Modo should, must y have got to.</li> <li>• Oraciones afirmativas, negativas e interrogativas con modals.</li> <li>• Términos para cartas formales.</li> <li>• Partes de las respuestas cortas.</li> <li>• Expresiones para ocupaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la octava unidad, cada una de las alumnas, comprendiendo la gramática de los auxiliares should y must es capaz de expresar una mayor cantidad de acciones en forma obligación y sugerencia. Además es capaz de expresar ideas describiendo ocupaciones. Asume la necesidad de escribir cartas formales.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 2: Going places! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempo Presente Simple y Futuro con Will</li> <li>• Primer Condicional</li> <li>• Colocaciones</li> <li>• Vocabulario de preposiciones de lugar y de tiempo</li> <li>• Expresiones de conexión de ideas</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la novena unidad, los alumnos habiendo identificado la forma de expresar presente reconocen la diferencia entre las formas de futuro y las aplican adecuadamente. Describen condiciones acuciosamente. Asumen expresiones para demostrar ubicación de lugar. Utilizan expresiones de tiempo y conectores para unir ideas varias.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

<b>Unidad 3: Scared to death! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Patrones de Verbos Infinitivos y gerundios</li> <li>• What + Infinitivo</li> <li>• Something + infinitive</li> <li>• Expresiones de sentimientos</li> <li>• Exclamaciones de sorpresa</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la décimo unidad los alumnos serán capaces de reconocer y utilizar patrones tiempos en pasado los utilizan adecuadamente. Utilizan expresiones de exclamación. Y describen sentimientos. Utilizarán conjunciones para unir ideas tipo.</li> </ul>
<b>Lecturas :</b> [SJ02], [Cam06], [Mac99]	

**Unidad 4: Things that changed the world! (0 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Voz Pasiva</li> <li>• Oraciones Afirmativas, Negativas y Preguntas</li> <li>• Uso de Participios, verbos y sustantivos que van unidos</li> <li>• Señales. Signos y notas</li> <li>• Resúmenes</li> <li>• Expresiones para indicar prohibición</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la décimo primera primera unidad, los alumnos habiendo identificado la idea de acciones pasivas describen acciones adecuadamente en diversas situaciones que la involucran. Reconocen y aplican participios. Asumen la idea de respetar signos y señales públicas. Expresan ideas de hábitos . Hacén resúmenes.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

**Unidad 5: Dreams and reality! (0 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Segundo Condicional</li> <li>• Auxiliar de Modo Might</li> <li>• Verbos de Frase</li> <li>• Vocabulario de expresiones sociales</li> <li>• Adverbios</li> <li>• Expresiones para dar consejo</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo segunda unidad, los alumnos, a partir de la comprensión de la idea de Conicionales y de expresar posibilidad elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de frases verbales (verbos de 2 palabras). Adquirirán vocabulario para describir expresiones sociales.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

**Unidad 6: Making a living! (0 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Present Perfect Continuous</li> <li>• Present Continuous</li> <li>• Ocupaciones</li> <li>• Formación de palabras</li> <li>• Adverbios</li> <li>• Expresiones de uso en el teléfono.</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo tercera unidad estructuran oraciones con acciones que incluyen presente y pasado en contextos adecuados. Enfatizan la diferencia entre tipos de ocupaciones. Utilizan expresiones adecuadas para conversaciones telefónicas</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

<b>Unidad 7: All you need is love! (0 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Pasado Perfecto y Pasado Simple</li> <li>• Expresiones de Reporte</li> <li>• Expresiones de palabras en contextos diferentes</li> <li>• Despedidas cortas y formales</li> <li>• Historias de amor</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la décimo cuarta unidad, los alumnos habiendo conocido los fundamentos de la estructuración del tiempo pasado perfecto, lo diferencian del pasado simple. Enfatizan la diferencia entre palabras en contextos diferentes. Describen ideas de despedidas. Utilizan expresiones para escribir historias de amor. Asumen la idea de dar y hacer entrevistas.</li> </ul>

**Lecturas :** [SJ02], [Cam06], [Mac99]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

[Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.

## **Capítulo 5**

# **Quinto Semestre**

## 1. CURSO

CS212-CS2023. Análisis y Diseño de Algoritmos (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS212-CS2023. Análisis y Diseño de Algoritmos
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	<ul style="list-style-type: none"><li>• CS210. Algoritmos y Estructuras de Datos. (4<sup>to</sup> Sem)</li><li>• CS211. Teoría de la Computación. (4<sup>to</sup> Sem)</li></ul>

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Un algoritmo es, esencialmente, un conjunto bien definido de reglas o instrucciones que permitan resolver un problema computacional. El estudio teórico del desempeño de los algoritmos y los recursos utilizados por estos, generalmente tiempo y espacio, nos permite evaluar si un algoritmo es adecuado para un resolver un problema específico, compararlo con otros algoritmos para el mismo problema o incluso delimitar la frontera entre lo viable y lo imposible.

Esta materia es tan importante que incluso Donald E. Knuth definió a Ciencia de la Computación como el estudio de algoritmos.

En este curso serán presentadas las técnicas más comunes utilizadas en el análisis y diseño de algoritmos eficientes, con el propósito de aprender los principios fundamentales del diseño, implementación y análisis de algoritmos para la solución de problemas computacionales.

## 5. OBJETIVOS

- Desarrollar la capacidad para evaluar la complejidad y calidad de algoritmos propuestos para un determinado problema.
- Estudiar los algoritmos más representativos, introductorios de las clases más importantes de problemas tratados en computación.
- Desarrollar la capacidad de resolución de problemas algorítmicos utilizando los principios fundamentales de diseño de algoritmos aprendidos.
- Ser capaz de responder a las siguientes preguntas cuando le sea presentado un nuevo algoritmo: ¿Cuán buen desempeño tiene?, ¿Existe una mejor forma de resolver el problema?

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

- 2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Análisis Básico (10 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo.</li><li>• Análisis asintótico de complejidad de cotas superior y esperada.</li><li>• Definición formal de la Notación Big O.</li><li>• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.</li><li>• Uso de la notación Big O.</li><li>• Relaciones recurrentes.</li><li>• Análisis de algoritmos iterativos y recursivos.</li><li>• Algunas versiones del Teorema Maestro.</li></ul>	<ul style="list-style-type: none"><li>• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Evaluar]</li><li>• En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar]</li><li>• Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Evaluar]</li><li>• Indique la definición formal de Big O [Evaluar]</li><li>• Lista y contraste de clases estándares de complejidad [Evaluar]</li><li>• Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Evaluar]</li><li>• Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Evaluar]</li><li>• Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Evaluar]</li><li>• Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Evaluar]</li><li>• Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Evaluar]</li></ul>
<b>Lecturas :</b> [KT05], [DPV06], [RS09], [SF13], [Knu97]	

**Unidad 2: Estrategias Algorítmicas (30 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Algoritmos de fuerza bruta.</li> <li>• Algoritmos voraces.</li> <li>• Divide y vencerás.</li> <li>• Programación Dinámica.</li> </ul>	<ul style="list-style-type: none"> <li>• Para cada una de las estrategias (fuerza bruta, algoritmo goloso, divide y vencerás, recursividad en reversa y programación dinámica), identifica un ejemplo práctico en el cual se pueda aplicar [Evaluar]</li> <li>• Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar]</li> <li>• Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar]</li> <li>• Usa programación dinámica para resolver un problema determinado [Evaluar]</li> <li>• Determina el enfoque algorítmico adecuado para un problema [Evaluar]</li> </ul>

**Lecturas :** [KT05], [DPV06], [RS09], [Als99]

### Unidad 3: Algoritmos y Estructuras de Datos fundamentales (10 horas)

#### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo.</li> <li>Algoritmos de búsqueda secuencial y binaria.</li> <li>Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción)</li> <li>Algoritmos de ordenamiento con peor caso o caso promedio en <math>O(N \lg N)</math> (Quicksort, Heapsort, Mergesort)</li> <li>Grafos y algoritmos en grafos: <ul style="list-style-type: none"> <li>Representación de grafos (ej., lista de adyacencia, matriz de adyacencia)</li> <li>Recorrido en profundidad y amplitud</li> </ul> </li> <li>Montículos (Heaps)</li> <li>Grafos y algoritmos en grafos: <ul style="list-style-type: none"> <li>Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd)</li> <li>Árbol de expansión mínima (algoritmos de Prim y Kruskal)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Implementar algoritmos numéricos básicos [Evaluar]</li> <li>Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar]</li> <li>Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y <math>O(N \log N)</math> [Evaluar]</li> <li>Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Usar]</li> <li>Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse]</li> <li>Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar]</li> <li>Demostrar habilidad para evaluar algoritmos, para seleccionar de un rango de posibles opciones, para proveer una justificación por esa selección, y para implementar el algoritmo en un contexto específico [Evaluar]</li> <li>Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Evaluar]</li> <li>Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar]</li> </ul>

Lecturas : [KT05], [DPV06], [RS09], [SW11], [GT09]

### Unidad 4: Computabilidad y complejidad básica de autómatas (2 horas)

#### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Introducción a las clases P y NP y al problema P vs. NP.</li> <li>Introducción y ejemplos de problemas NP-Completos y a clases NP-Completo.</li> </ul>	<ul style="list-style-type: none"> <li>Define las clases P y NP [Familiarizarse]</li> <li>Explique el significado de NP-Complejidad [Familiarizarse]</li> </ul>

Lecturas : [KT05], [DPV06], [RS09]

<b>Unidad 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (8 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Grafos (ej. Ordenamiento Topológico, encontrando componentes puertamente conectados)</li> <li>• Algoritmos Teórico-Numéricos (Aritmética Modular, Prueba del Número Primo, Factorización Entera)</li> <li>• Algoritmos aleatorios.</li> <li>• Análisis amortizado.</li> <li>• Análisis Probabilístico.</li> </ul>	<ul style="list-style-type: none"> <li>• Entender el mapeamiento de problemas del mundo real a soluciones algorítmicas (ejemplo, problemas de grafos, programas lineares,etc) [Familiarizarse]</li> <li>• Seleccionar y aplicar técnicas de algoritmos avanzadas (ejemplo, randomización, aproximación) para resolver problemas reales [Usar]</li> <li>• Seleccionar y aplicar técnicas avanzadas de análisis (ejemplo, amortizado, probabilístico,etc) para algoritmos [Usar]</li> </ul>
<b>Lecturas :</b> [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983.
- [Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms*. Computer Science Press, 1992.
- [Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1*. Third Edition. Addison-Wesley, 1997. URL: <http://www-cs-faculty.stanford/~knuth/taocp.html>.
- [Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scientific, 1999.
- [KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005.
- [DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006.
- [GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 2009.
- [RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009.
- [SW11] R. Sedgewick and K. Wayne. *Algorithms*. Pearson Education, 2011.
- [SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013.

## 1. CURSO

CS261-CS2023. Inteligencia Artificial (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS261-CS2023. Inteligencia Artificial
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	ST251. Estadística y Probabilidades. (3 <sup>er</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La investigación en Inteligencia Artificial ha conducido al desarrollo de numerosas tópicas relevantes, dirigidas a la automatización de la inteligencia humana, dando una visión panorámica de diferentes algoritmos que simulan los diferentes aspectos del comportamiento y la inteligencia del ser humano.

## 5. OBJETIVOS

- Evaluar las posibilidades de simulación de la inteligencia, para lo cual se estudiarán las técnicas de modelización del conocimiento.
- Construir una noción de inteligencia que soporte después las tareas de su simulación.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**1)** Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Cuestiones fundamentales (2 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>● Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial.</li> <li>● ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> <li>– El Test de Turing</li> <li>– Razonamiento Racional versus No Racional</li> </ul> </li> <li>● Características del Problema: <ul style="list-style-type: none"> <li>– Observable completamente versus observable parcialmente</li> <li>– Individual versus multi-agente</li> <li>– Determinístico versus estocástico</li> <li>– Estático versus dinámico</li> <li>– Discreto versus continuo</li> </ul> </li> <li>● Naturaleza de agentes: <ul style="list-style-type: none"> <li>– Autónomo versus semi-autónomo</li> <li>– Reflexivo, basado en objetivos, y basado en utilidad</li> <li>– La importancia en percepción e interacciones con el entorno</li> </ul> </li> <li>● Cuestiones filosóficas y éticas.</li> </ul>	<ul style="list-style-type: none"> <li>● Describir el test de Turing y el experimento pensado cuarto chino” (<i>Chinese Room</i>) [Usar]</li> <li>● Determinando las características de un problema dado que la Inteligencia Artificial deberían resolver [Usar]</li> </ul>

**Lecturas :** [De 06], [Pon+14]

<b>Unidad 2: Agentes (2 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>● Definición de Agentes</li> <li>● Arquitectura de agentes (Ej. reactivo, en capa, cognitivo)</li> <li>● Teoría de agentes</li> <li>● Racionalidad, teoría de juegos: <ul style="list-style-type: none"> <li>– Agentes de decisión teórica</li> <li>– Procesos de decisión de Markov (MDP)</li> </ul> </li> <li>● Agentes de Software, asistentes personales, y acceso a información: <ul style="list-style-type: none"> <li>– Agentes colaborativos</li> <li>– Agentes de recolección de información</li> <li>– Agentes creíbles (carácter sintético, modelamiento de emociones en agentes)</li> </ul> </li> <li>● Agentes de aprendizaje</li> <li>● Sistemas Multi-agente <ul style="list-style-type: none"> <li>– Agentes Colaborativos</li> <li>– Equipos de Agentes</li> <li>– Agentes Competitivos (ej., subastas, votaciones)</li> <li>– Sistemas de enjambre y modelos biológicamente inspirados</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Lista las características que definen un agente inteligente [Usar]</li> <li>● Describe y contrasta las arquitecturas de agente estándares [Usar]</li> <li>● Describe las aplicaciones de teoría de agentes para dominios como agentes de software, asistentes personales, y agentes creíbles [Usar]</li> <li>● Describe los paradigmas primarios usados por agentes de aprendizaje [Usar]</li> <li>● Demuestra mediante ejemplos adecuados como los sistemas multi-agente soportan interacción entre agentes [Usar]</li> </ul>

**Lecturas :** [Nil01], [RN03], [Pon+14]

### Unidad 3: Estrategias de búsquedas básicas (2 horas)

#### Resultados esperados:

Temas	Objetivos de Aprendizaje (Learning Outcomes)
<ul style="list-style-type: none"> <li>• Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda.</li> <li>• Factored representation (factoring state hacia variables)</li> <li>• Uninformed search (breadth-first, depth-first, depth-first with iterative deepening)</li> <li>• Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*)</li> <li>• El espacio y el tiempo de la eficiencia de búsqueda.</li> <li>• Dos jugadores juegos (introducción a la búsqueda minimax).</li> <li>• Satisfacción de restricciones (backtracking y métodos de búsqueda local).</li> </ul>	<ul style="list-style-type: none"> <li>• Formula el espacio eficiente de un problema para un caso expresado en lenguaje natural (ejm. Inglés) en términos de estados de inicio y final, así como sus operadores [Usar]</li> <li>• Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Usar]</li> <li>• Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Usar]</li> <li>• Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar]</li> </ul>

Lecturas : [Nil01], [Pon+14]

### Unidad 4: Búsqueda Avanzada (18 horas)

#### Resultados esperados:

Temas	Objetivos de Aprendizaje (Learning Outcomes)
<ul style="list-style-type: none"> <li>• Búsqueda estocástica: <ul style="list-style-type: none"> <li>– Simulated annealing</li> <li>– Algoritmos genéticos</li> <li>– Búsqueda de árbol Monte-Carlo</li> </ul> </li> <li>• Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda.</li> <li>• Implementación de búsqueda A *, búsqueda en haz.</li> <li>• Búsqueda Minimax, poda alfa-beta.</li> <li>• Búsqueda Expectimax (MDP-Solving) y los nodos de azar.</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar e implementar una solución a un problema con algoritmo genético [Usar]</li> <li>• Diseñar e implementar un esquema de recocido simulado (<i>simulated annealing</i>) para evitar mínimos locales en un problema [Usar]</li> <li>• Diseñar e implementar una búsqueda A* y búsqueda en haz (<i>beam search</i>) para solucionar un problema [Usar]</li> <li>• Aplicar búsqueda minimax con poda alfa-beta para simplificar el espacio de búsqueda en un juego con dos jugadores [Usar]</li> <li>• Comparar y contrastar los algoritmos genéticos con técnicas clásicas de búsqueda [Usar]</li> <li>• Comparar y contrastar la aplicabilidad de varias heurísticas de búsqueda, para un determinado problema [Usar]</li> </ul>

Lecturas : [Gol89], [Nil01], [RN03], [Pon+14]

<b>Unidad 5: Razonamiento Bajo Incertidumbre (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Revisión de Probabilidad Básica</li> <li>• Variables aleatorias y distribuciones de probabilidad: <ul style="list-style-type: none"> <li>– Axiomas de probabilidad</li> <li>– Inferencia probabilística</li> <li>– Regla de Bayes</li> </ul> </li> <li>• Independencia Condicional</li> <li>• Representaciones del conocimiento: <ul style="list-style-type: none"> <li>– Redes bayesianas <ul style="list-style-type: none"> <li>* Inferencia exacta y su complejidad</li> <li>* Métodos de Muestreo aleatorio (Monte Carlo) (p.e. Muestreo de Gibbs)</li> </ul> </li> <li>– Redes Markov</li> <li>– Modelos de probabilidad relacional</li> <li>– Modelos ocultos de Markov</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar la regla de Bayes para determinar el cumplimiento de una hipótesis [Usar]</li> <li>• Explicar cómo al tener independencia condicional permite una gran eficiencia en sistemas probabilísticos [Usar]</li> <li>• Identificar ejemplos de representación de conocimiento para razonamiento bajo incertidumbre [Usar]</li> <li>• Indicar la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Usar]</li> </ul>

**Lecturas :** [KF09], [RN03]

<b>Unidad 6: Aprendizaje Automático Básico (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación.</li> <li>• Aprendizaje inductivo</li> <li>• Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión.</li> <li>• El problema exceso de ajuste.</li> <li>• Medición clasificada con exactitud.</li> </ul>	<ul style="list-style-type: none"> <li>• Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Usar]</li> <li>• Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Usar]</li> <li>• Explicar la diferencia entre aprendizaje inductivo y deductivo [Usar]</li> <li>• Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Usar]</li> <li>• Aplicar un algoritmo de aprendizaje estadístico simple como el Clasificador Naïve Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]</li> </ul>

**Lecturas :** [Mit98], [RN03], [Pon+14]

**Unidad 7: Aprendizaje de máquina avanzado (20 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina</li> <li>• Aprendizaje general basado en estadística, estimación de parámetros (máxima probabilidad)</li> <li>• Programación lógica inductiva (<i>Inductive logic programming ILP</i>)</li> <li>• Aprendizaje supervisado <ul style="list-style-type: none"> <li>– Aprendizaje basado en árboles de decisión</li> <li>– Aprendizaje basado en redes neuronales</li> <li>– Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>)</li> </ul> </li> <li>• Aprendizaje y <i>clustering</i> no supervisado <ul style="list-style-type: none"> <li>– EM</li> <li>– K-means</li> <li>– Mapas auto-organizados</li> </ul> </li> <li>• Aprendizaje semi-supervisado.</li> <li>• Aprendizaje de modelos gráficos</li> <li>• Evaluación del desempeño (tal como cross-validation, área bajo la curva ROC)</li> <li>• Aplicación de algoritmos Machine Learning para Minería de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explica las diferencias entre los tres estilos de aprendizaje: supervisado, por refuerzo y no supervisado [Usar]</li> <li>• Implementa algoritmos simples para el aprendizaje supervisado, aprendizaje por refuerzo, y aprendizaje no supervisado [Usar]</li> <li>• Determina cuál de los tres estilos de aprendizaje es el apropiado para el dominio de un problema en particular [Usar]</li> <li>• Compara y contrasta cada una de las siguientes técnicas, dando ejemplo de cuando una estrategia es la mejor: árboles de decisión, redes neuronales, y redes bayesianas [Usar]</li> <li>• Evalúa el rendimiento de un sistema de aprendizaje simple en un conjunto de datos reales [Usar]</li> <li>• Describe el estado del arte en la teoría del aprendizaje, incluyendo sus logros y limitantes [Usar]</li> <li>• Explica el problema del sobreajuste, conjuntamente con técnicas para determinar y manejar el problema [Usar]</li> </ul>

**Lecturas :** [RN03], [KF09], [Mur12]

## Unidad 8: Procesamiento del Lenguaje Natural (12 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Gramáticas determinísticas y estocásticas</li> <li>• Algoritmos de parseo <ul style="list-style-type: none"> <li>– Gramáticas libres de contexto (CFGs) y cuadros de parseo (e.g. Cocke-Younger-Kasami CYK)</li> <li>– CFGs probabilísticos y ponderados CYK</li> </ul> </li> <li>• Representación del significado / Semántica <ul style="list-style-type: none"> <li>– Representación de conocimiento basado en lógica</li> <li>– Roles semánticos</li> <li>– Representaciones temporales</li> <li>– Creencias, deseos e intenciones</li> </ul> </li> <li>• Métodos basados en el corpus</li> <li>• N-gramas y Modelos ocultos de Markov (HMMs)</li> <li>• Suavizado y back-off</li> <li>• Ejemplos de uso: POS etiquetado y morfología</li> <li>• Recuperación de la información: <ul style="list-style-type: none"> <li>– Modelo de espacio vectorial <ul style="list-style-type: none"> <li>* TF &amp; IDF</li> </ul> </li> <li>– Precisión y cobertura</li> </ul> </li> <li>• Extracción de información</li> <li>• Traducción de lenguaje</li> <li>• Clasificación y categorización de texto: <ul style="list-style-type: none"> <li>– Modelo de bolsa de palabras</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Define y contrasta gramáticas de tipo estocásticas y determinísticas, dando ejemplos y demostrando como adecuar cada una de ellas [Usar]</li> <li>• Simula, aplica, o implementa algoritmos clásicos y estocásticos para el parseo de un lenguaje natural [Usar]</li> <li>• Identifica los retos de la representación del significado [Usar]</li> <li>• Lista las ventajas de usar corpus estándares. Identifica ejemplos de corpus actuales para una variedad de tareas de PLN [Usar]</li> <li>• Identifica técnicas para la recuperación de la información, traducción de lenguajes, y clasificación de textos [Usar]</li> </ul>

Lecturas : [Nil01], [RN03], [Pon+14]

Unidad 9: Visión y percepción por computador (12 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Visión Computacional <ul style="list-style-type: none"> <li>Adquisición de imágenes, representación, procesamiento y propiedades</li> <li>Representación de formas, reconocimiento y segmentación de objetos</li> <li>Ánálisis de movimiento</li> </ul> </li> <li>Modularidad en reconocimiento.</li> <li>Enfoques de reconocimiento de patrones <ul style="list-style-type: none"> <li>Algoritmos de clasificación y medidas de calidad de la clasificación.</li> <li>Técnicas estadísticas.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar]</li> <li>Listar al menos tres aproximaciones de segmentación de imágenes, tales como algoritmos de límites (thresholding), basado en el borde y basado en regiones, junto con sus características definitorias, fortalezas y debilidades [Usar]</li> <li>Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar]</li> <li>Proporcionar al menos dos ejemplos de transformación de una fuente de datos de un dominio sensorial a otro, ejemplo, datos táctiles interpretados como imágenes en 2d de una sola banda [Usar]</li> <li>Implementar un algoritmo para la extracción de características en información real, ejemplo, un detector de bordes o esquinas para imágenes o vectores de coeficientes de Fourier describiendo una pequeña porción de señal de audio [Usar]</li> <li>Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Usar]</li> <li>Evaluar el desempeño de la función de extracción subyacente, en relación con al menos una aproximación alternativa posible (ya sea implementado o no) en su contribución a la tarea de clasificación (8) anterior [Usar]</li> </ul>

Lecturas : [Nil01], [RN03], [Pon+14]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

- 
- [Mit98] M. Mitchell. *An introduction to genetic algorithms*. The MIT press, 1998.
  - [Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión*. McGraw-Hill, 2001.
  - [RN03] Stuart Russell and Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice Hall, 2003.
  - [De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. CRC Press, 2006.
  - [KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
  - [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
  - [Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

## 1. CURSO

CS272-CS2023. Bases de Datos II (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS272-CS2023. Bases de Datos II
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS271. Bases de Datos I. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La Gestión de la Información (*IM-Information Management*) juega un rol principal en casi todas las áreas donde los computadores son usados. Esta área incluye la captura, digitalización, representación, organización, transformación y presentación de información; algoritmos para mejorar la eficiencia y efectividad del acceso y actualización de información almacenada, modelamiento de datos y abstracción, y técnicas de almacenamiento de archivos físicos.

Este también abarca la seguridad de la información, privacidad, integridad y protección en un ambiente compartido. Los estudiantes necesitan ser capaces de desarrollar modelos de datos conceptuales y físicos, determinar que métodos de IM y técnicas son apropiados para un problema dado, y ser capaces de seleccionar e implementar una apropiada solución de IM que refleje todas las restricciones aplicables, incluyendo escalabilidad y usabilidad.

## 5. OBJETIVOS

- Hacer que el alumno entienda las diferentes aplicaciones que tienen las bases de datos, en las diversas áreas de conocimiento.
- Mostrar las formas adecuadas de almacenamiento de información basada en sus diversos enfoques y su posterior recuperación de información.

## 6. RESULTADOS DEL ESTUDIANTE

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

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

## 7. TEMAS

**Unidad 1: Diseño Físico de Bases de Datos (10 horas)**

**Resultados esperados: 6**

<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Almacenamiento y estructura de archivos.</li> <li>Archivos indexados.</li> <li>Archivos Hash.</li> <li>Archivos de Firma.</li> <li>Árboles B.</li> <li>Archivos con índice denso.</li> <li>Archivos con registros de tamaño variable.</li> <li>Eficiencia y Afinación de Bases de Datos.</li> </ul>	<ul style="list-style-type: none"> <li>Explica los conceptos de registro, tipos de registro, y archivos, así como las diversas técnicas para colocar registros de archivos en un disco [Usar]</li> <li>Da ejemplos de la aplicación de índices primario, secundario y de agrupamiento [Usar]</li> <li>Distingue entre un índice no denso y uno denso [Usar]</li> <li>Implementa índices de multinivel dinámicos usando árboles-B [Usar]</li> <li>Explica la teoría y la aplicación de técnicas de hash internas y externas [Usar]</li> <li>Usa técnicas de hasp para facilitar la expansión de archivos dinámicos [Usar]</li> <li>Describe las relaciones entre hashing, compresión, y búsquedas eficientes en bases de datos [Usar]</li> <li>Evalúa el costo y beneficio de diversos esquemas de hashing [Usar]</li> <li>Explica como el diseño físico de una base de datos afecta la eficiencia de las transacciones en ésta [Usar]</li> </ul>

**Lecturas :** [Bur04], [Cel05]

**Unidad 2: Procesamiento de Transacciones (12 horas)**

**Resultados esperados: 6**

<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Transacciones.</li> <li>Fallo y recuperación.</li> <li>Control concurrente.</li> <li>Interacción de gestión de transacciones con el almacenamiento, especialmente en almacenamiento.</li> </ul>	<ul style="list-style-type: none"> <li>Crear una transacción mediante la incorporación de SQL en un programa de aplicación [Usar]</li> <li>Explicar el concepto de confirmaciones implícitas [Usar]</li> <li>Describir los problemas específicos para la ejecución de una transacción eficiente [Usar]</li> <li>Explicar cuando y porqué se necesita un <i>rollback</i>, y cómo registrar todo asegura un <i>rollback</i> adecuado [Usar]</li> <li>Explicar el efecto de diferentes niveles de aislamiento sobre los mecanismos de control de concurrencia [Usar]</li> <li>Elejir el nivel de aislamiento adecuado para la aplicación de un protocolo de transacción especificado [Usar]</li> <li>Identificar los límites apropiados de la transacción en programas de aplicación [Usar]</li> </ul>

**Lecturas :** [Phi97], [Ram04]

**Unidad 3: Almacenamiento y Recuperación de Información (10 horas)**

**Resultados esperados: 6**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Documentos, publicación electrónica, markup, y lenguajes markup.</li> <li>• Tries, archivos invertidos, Árboles PAT, archivos de firma, indexación.</li> <li>• Análisis Morfológico, stemming, frases, stop lists.</li> <li>• Distribuciones de frecuencia de términos, incertidumbre, fuzzificación (fuzzyness), ponderación.</li> <li>• Espacio vectorial, probabilidad, lógica, y modelos avanzados.</li> <li>• Necesidad de Información , Relevancia, evaluación, efectividad.</li> <li>• Thesauri, ontologías, clasificación y categorización, metadata.</li> <li>• Información bibliográfica, bibliometría, citaciones.</li> <li>• Enrutamiento y filtrado.</li> <li>• Búsqueda multimedia.</li> <li>• Información de resumen y visualización.</li> <li>• Búsqueda por facetas (por ejemplo, el uso de citas, palabras clave, esquemas de clasificación).</li> <li>• Librerías digitales.</li> <li>• Digitalización, almacenamiento, intercambio, objetos digitales, composición y paquetes.</li> <li>• Metadata y catalogación.</li> <li>• Nombramiento, repositorios, archivos</li> <li>• Archivamiento y preservación, integridad</li> <li>• Espacios (Conceptual, geográfico, 2/3D, Realidad virtual)</li> <li>• Arquitecturas (agentes, autobuses, envolturas / mediadores), de interoperabilidad.</li> <li>• Servicios (búsqueda, de unión, de navegación, y así sucesivamente).</li> <li>• Gestión de derechos de propiedad intelectual, la privacidad y la protección (marcas de agua).</li> </ul>	<ul style="list-style-type: none"> <li>• Explica los conceptos básicos de almacenamiento y recuperación de la información [Usar]</li> <li>• Describe qué temas son específicos para una recuperación de la información eficiente [Usar]</li> <li>• Da aplicaciones de estrategias alternativas de búsqueda y explica por qué una estrategia en particular es apropiada para una aplicación [Usar]</li> <li>• Diseña e implementa un sistema de almacenamiento y recuperación de la información o librería digital de tamaño pequeño a mediano [Usar]</li> <li>• Describe algunas de las soluciones técnicas a los problemas relacionados al archivamiento y preservación de la información en una librería digital [Usar]</li> </ul>

**Lecturas :** [Pet98], [Ram04]

<b>Unidad 4: Bases de Datos Distribuidas (36 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• DBMS Distribuidas <ul style="list-style-type: none"> <li>– Almacenamiento de datos distribuido</li> <li>– Procesamiento de consultas distribuido</li> <li>– Modelo de transacciones distribuidas</li> <li>– Soluciones homogéneas y heterogéneas</li> <li>– Bases de datos distribuidas cliente-servidor</li> </ul> </li> <li>• Parallel DBMS <ul style="list-style-type: none"> <li>– Arquitecturas paralelas DBMS: memoria compartida, disco compartido, nada compartido;</li> <li>– Aceleración y ampliación, por ejemplo, el uso del modelo de procesamiento MapReduce</li> <li>– Replicación de información y modelos de consistencia débil</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Explicar las técnicas usadas para la fragmentación de datos, replicación, y la asignación durante el proceso de diseño de base de datos distribuida [Usar]</li> <li>• Evaluar estrategias simples para la ejecución de una consulta distribuida para seleccionar una estrategia que minimice la cantidad de transferencia de datos [Usar]</li> <li>• Explicar como el protocolo de dos fases de <i>commit</i> es usado para resolver problemas de transacciones que acceden a bases de datos almacenadas en múltiples nodos [Usar]</li> <li>• Describir el control concurrente distribuido basados en técnicas de copia distinguibles y el método de votación. [Usar]</li> <li>• Describir los tres niveles del software en el modelo cliente servidor [Usar]</li> </ul>

**Lecturas :** [M T99]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Phi97] Eric Newcomer Philip A. Bernstein. *Principles of Transaction Processing, First Edition*. Morgan Kaufmann, 1997.
- [Pet98] Julita Vassileva Peter Brusilovsky Alfred Kobsa. *Adaptive Hypertext and Hypermedia, First Edition*. Springer, 1998.
- [M T99] Patrick Valduriez M. Tamer Ozsu. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.
- [Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.
- [Ram04] Shamkant B. Navathe Ramez Elmasri. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.
- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

## 1. CURSO

CS291-CS2023. Ingeniería de Software I (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS291-CS2023. Ingeniería de Software I
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
		• CS113. Programación Orientada a Objetos II. (3 <sup>er</sup> Sem)
<b>2.8 Prerrequisitos</b>	:	• CS271. Bases de Datos I. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La ingeniería de software moderna requiere profesionales capaces de diseñar sistemas robustos y mantenibles en entornos cloud-native y ágiles. Este curso desarrolla competencias en arquitectura limpia, patrones de diseño, testing profesional y prácticas DevSecOps, preparando estudiantes para los desafíos de la industria actual.

## 5. OBJETIVOS

- Desarrollar competencias en diseño de arquitecturas de software escalables y mantenibles aplicando principios SOLID y patrones de diseño establecidos.
- Dominar herramientas profesionales de desarrollo incluyendo Python tipado, pytest avanzado, Git colaborativo y automatización con Makefiles.
- Implementar metodologías ágiles con métricas de calidad, gestión de proyectos y mejora continua en equipos de desarrollo.
- Aplicar principios de seguridad en el diseño de software y prácticas de testing que aseguren la confiabilidad de sistemas críticos.
- Producir documentación técnica profesional utilizando estándares internacionales como C4 Model y UML para comunicar decisiones de diseño.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Control de versiones y desarrollo en equipo (git) (8 horas)</b>	
<b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Flujos: <i>trunk-based</i> (recomendado) vs. Git-flow.</li><li><code>rebase</code> interactivo, <code>cherry-pick</code>, <code>bisect</code>, <code>reflog</code>.</li><li>Mensajes tipo <i>Conventional Commits</i> y Pull Requests profesionales.</li><li>Git hooks + GitHub Actions básicas (linters + tests).</li><li>Resolución de conflictos complejos en código y archivos de configuración.</li><li>Meta-objetivo: al menos 50 PRs revisados y recibidos durante el curso.</li></ul>	<ul style="list-style-type: none"><li>Evaluar y seleccionar entre flujos de trabajo modernos (<i>trunk-based</i> vs. Git-flow) según las características del proyecto y del equipo.</li><li>Aplicar comandos avanzados de Git (<code>rebase</code>, <code>cherry-pick</code>, <code>bisect</code>) para mantener un historial limpio, incorporar cambios selectivos y depurar eficientemente la introducción de errores.</li><li>Crear mensajes de commit siguiendo el estándar Conventional Commits y redactar Pull Requests profesionales que faciliten la revisión de código.</li><li>Configurar y utilizar hooks de Git y pipelines básicas de GitHub Actions para automatizar la verificación de calidad (linters) y la ejecución de tests.</li><li>Resolver conflictos complejos en merges o rebases, tanto en código como en archivos de configuración, analizando las diferencias y tomando decisiones integradoras.</li><li>Demostrar competencia en un entorno de desarrollo colaborativo mediante la revisión activa de código y la incorporación de feedback en sus propios trabajos (meta-objetivo de 50 PRs).</li></ul>
<b>Lecturas :</b> [CS23], [TH24], [Fow19], [Mic23]	

<b>Unidad 2: Linux y Bash para desarrolladores (8 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Procesos, señales, permisos, <code>pipes</code> y redirecciones.</li> <li><code>grep</code>, <code>awk</code>, <code>sed</code>, <code>find</code>, <code>rg</code>, <code>fd</code> sobre código y logs reales.</li> <li>Bash scripting robusto: <code>set -euo pipefail</code>, funciones, <code>getopts</code>, <code>trap</code>.</li> <li>SSH + claves, configuración en <code>ssh-config</code>, <code>agent forwarding</code>.</li> <li>Introducción a <code>Makefile</code> como herramienta de orquestación local.</li> </ul>	<ul style="list-style-type: none"> <li>Gestionar procesos del sistema, permisos de archivos y flujos de datos utilizando eficientemente pipes, redirecciones y señales.</li> <li>Construir pipelines de procesamiento de texto para analizar código y logs en tiempo real, utilizando herramientas como <code>grep</code>, <code>awk</code>, <code>sed</code> y sus alternativas modernas (<code>rg</code>, <code>fd</code>).</li> <li>Diseñar scripts Bash robustos y mantenibles que incluyan manejo estricto de errores, parámetros nombrados y limpieza de recursos mediante <code>trap</code>.</li> <li>Configurar y utilizar conexiones SSH seguras con autenticación por claves, multiplexación y agent forwarding para un desarrollo remoto eficiente.</li> <li>Emplear <code>Makefile</code> para orquestar tareas de desarrollo complejas, automatizando flujos de construcción, testing y despliegue local.</li> </ul>

**Lecturas :** [Sho19], [RN24], [Ker21], [Fou23]

<b>Unidad 3: Procesos de Software (4 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.</li> <li>Introducción a modelos del proceso de software (e.g., cascada, incremental, agil): <ul style="list-style-type: none"> <li>Actividades con ciclos de vida de software.</li> </ul> </li> <li>Programación a gran escala versus programación individual.</li> <li>Evaluación de modelos de proceso de software.</li> <li>Conceptos de calidad de software.</li> <li>Mejoramiento de procesos.</li> <li>Modelos de madurez de procesos de software.</li> <li>Mediciones del proceso de software.</li> </ul>	

**Lecturas :** [PM19], [Som20], [Mar17]

<b>Unidad 4: Ingeniería de Requisitos (6 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios.</li> <li>• Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad.</li> <li>• Requisitos de software elicitation.</li> <li>• Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación.</li> <li>• Requisitos no funcionales y su relación con la calidad del software.</li> <li>• Evaluación y uso de especificaciones de requisitos.</li> <li>• Requisitos de las técnicas de modelado de análisis.</li> <li>• La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema.</li> <li>• Prototipos.</li> <li>• Conceptos básicos de la especificación formal de requisitos.</li> <li>• Especificación de requisitos.</li> <li>• Validación de requisitos.</li> <li>• Rastreo de requisitos.</li> </ul>	

**Lecturas :** [WB13], [Coh04], [Robertson2012]

<b>Unidad 5: Diseño de Software (16 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar.</li> <li>Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio.</li> <li>Modelos estructurales y de comportamiento de los diseños de software.</li> <li>Diseño de patrones.</li> <li>Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.</li> <li>Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).</li> <li>El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standar widget set)</li> <li>Diseños de refactorización utilizando patrones de diseño</li> <li>Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.</li> <li>Medición y análisis de la calidad de un diseño.</li> <li>Compensaciones entre diferentes aspectos de la calidad.</li> <li>Aplicaciones en frameworks.</li> <li>Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo.</li> <li>Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design). <ul style="list-style-type: none"> <li>– Principio de privilegios mínimos</li> <li>– Principio de falla segura por defecto</li> <li>– Principio de aceptabilidad psicológica</li> </ul> </li> </ul>	

<b>Unidad 6: Principios de Diseño Seguro (4 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Menor privilegio y aislamiento.</li> <li>• Valores predeterminados a prueba de fallos.</li> <li>• Diseño abierto.</li> <li>• La seguridad de extremo a extremo.</li> <li>• La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas)</li> <li>• Diseño de seguridad.</li> <li>• Las tensiones entre la seguridad y otros objetivos de diseño.</li> <li>• Mediación completa.</li> <li>• El uso de componentes de seguridad vetados.</li> <li>• Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque)</li> <li>• Seguridad utilizable.</li> <li>• Componibilidad de seguridad.</li> <li>• Prevención, detección y disuasión.</li> </ul>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>

**Lecturas :** [HL06], [Mar17], [Sho14]

**Unidad 7: Verificación y Validación de Software (10 horas)****Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>● Verificación y validación de conceptos.</li><li>● Inspecciones, revisiones, auditorias.</li><li>● Tipos de pruebas, incluyendo la interfa- s humano computador, usabilidad, confiabilidad, seguridad, desempeño para la especificación.</li><li>● Fundamentos de testeo:<ul style="list-style-type: none"><li>– Pruebas de Unit, integración, validación y de Sistema</li><li>– Creación de plan de pruebas y generación de casos de test</li><li>– Técnicas de test de caja negra y caja blanca</li><li>– Test de regresión y automatización de pruebas</li></ul></li><li>● Seguimiento de defectos.</li><li>● Limitaciones de testeo en dominios particulares, tales como sistemas paralelos o críticos en cuanto a seguridad.</li><li>● Enfoques estáticos y enfoques dinámicos para la verificación.</li><li>● Desarrollo basado en pruebas.</li><li>● Plan de Validación, documentación para validación.</li><li>● Pruebas Orientadas a Objetos, Sistema de Pruebas.</li><li>● Verificación y validación de artefactos no codificados (documentación, archivos de ayuda, materiales de entrenamiento)</li><li>● Logeo fallido, error crítico y apoyo técnico para dichas actividades.</li><li>● Estimación fallida y terminación de las pruebas que incluye la envíos por defecto.</li></ul>	

**Lecturas :** [Bec02], [Mes07], [FP09]

<b>Unidad 8: Métodos de Desarrollo (4 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>● Comprensión de programas.</li> <li>● Exactitud del programa. <ul style="list-style-type: none"> <li>– Tipos de error (sintaxis, logica, tiempo de ejecucion)</li> <li>– El concepto de la especificacion</li> <li>– Programación defensiva (ejem., codificación segura, manejo de excepciones)</li> <li>– Revision de codigo</li> <li>– Fundamentos de <i>testing</i> y generación de casos de prueba</li> <li>– El rol y el uso de contratos, incluyendo pre- y post- condiciones</li> <li>– Pruebas unitarias</li> </ul> </li> <li>● Refactorización simple.</li> <li>● Entornos modernos de programación: <ul style="list-style-type: none"> <li>– Búsqueda de código.</li> <li>– Programación usando librería de componentes y sus APIs.</li> </ul> </li> <li>● Estrategias de depuración.</li> <li>● Documentación y estilo de programas.</li> </ul>	

**Lecturas :** [Fow18], [Mar17], [Eva03]

<b>Unidad 9: Algoritmos y Diseño (6 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> <li>– Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones)</li> </ul> </li> <li>• Rol de los algoritmos en el proceso de solución de problemas</li> <li>• Estrategias de solución de problemas <ul style="list-style-type: none"> <li>– Funciones matemáticas iterativas y recursivas</li> <li>– Recorrido iterativo y recursivo en estructura de datos</li> <li>– Estrategias Divide y Conquistar</li> </ul> </li> <li>• Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> <li>– Abstracción</li> <li>– Descomposición de Programa</li> <li>– Encapsulamiento y camuflaje de información</li> <li>– Separación de comportamiento y aplicación</li> </ul> </li> </ul>	

**Lecturas :** [McC04], [Mar17], [SW11]

<b>Unidad 10: Fundamentos y Conceptos en Seguridad (6 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• CIA (Confidencialidad, Integridad, Disponibilidad)</li> <li>• Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque .</li> <li>• Autenticación y autorización, control de acceso (vs. obligatoria discrecional)</li> <li>• Concepto de la confianza y la honradez .</li> <li>• Ética (revelación responsable)</li> </ul>	

**Lecturas :** [Smith2019], [Sho14], [HL06]

**Unidad 11: Gestión de Proyectos de Software (5 horas)**

**Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● La participación del equipo: <ul style="list-style-type: none"> <li>– Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo</li> <li>– Roles y responsabilidades en un equipo de software</li> <li>– Equipo de resolución de conflictos</li> <li>– Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura)</li> </ul> </li> <li>● Estimación de esfuerzo (a nivel personal)</li> <li>● Riesgo. <ul style="list-style-type: none"> <li>– El papel del riesgo en el ciclo de vida</li> <li>– Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de</li> </ul> </li> <li>● Gestión de equipos: <ul style="list-style-type: none"> <li>– Organización de equipo y la toma de decisiones</li> <li>– Roles de identificación y asignación</li> <li>– Individual y el desempeño del equipo de evaluación</li> </ul> </li> <li>● Gestión de proyectos: <ul style="list-style-type: none"> <li>– Programación y seguimiento de elementos</li> <li>– Herramientas de gestión de proyectos</li> <li>– Análisis de Costo/Beneficio</li> </ul> </li> <li>● Software de medición y técnicas de estimación.</li> <li>● Aseguramiento de la calidad del software y el rol de las mediciones.</li> <li>● Riesgo. <ul style="list-style-type: none"> <li>– Identificación de riesgos y gestión.</li> <li>– Análisis riesgo y evaluación.</li> <li>– La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo)</li> <li>– Planificación de Riesgo</li> </ul> </li> <li>● En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.</li> </ul>	

**Lecturas :** [Coh09], [Schwaber2020], [Reinertsen2009]

<b>Unidad 12: Evolución de Software (3 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Desarrollo de Software en el contexto de código grande pre existente <ul style="list-style-type: none"> <li>– Cambios de software</li> <li>– Preocupaciones y ubicación de preocupaciones</li> <li>– <i>Refactoring</i></li> </ul> </li> <li>• Evolución de Software.</li> <li>• Características de Software mantenible.</li> <li>• Sistemas de Reingeniería.</li> <li>• Reuso de Software. <ul style="list-style-type: none"> <li>– Segmentos de código</li> <li>– Bibliotecas y <i>frameworks</i></li> <li>– Componentes</li> <li>– Líneas de Producto</li> </ul> </li> </ul>	
<b>Lecturas :</b> [Fea04], [Fow18], [Mar17]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [Bec02] Kent Beck. *Test-Driven Development: By Example*. Addison-Wesley, 2002.
- [Eva03] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley, 2003.
- [Coh04] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley, 2004.
- [Fea04] Michael Feathers. *Working Effectively with Legacy Code*. Prentice Hall, 2004.
- [McC04] Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*. 2nd. Microsoft Press, 2004.
- [HL06] Michael Howard and David LeBlanc. *Writing Secure Code*. 2nd. Microsoft Press, 2006.
- [Mes07] Gerard Meszaros. *xUnit Test Patterns: Refactoring Test Code*. Addison-Wesley, 2007.
- [Coh09] Mike Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley, 2009.
- [FP09] Steve Freeman and Nat Pryce. *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley, 2009.
- [SW11] Robert Sedgewick and Kevin Wayne. *Algorithms*. 4th. Addison-Wesley, 2011.
- [WB13] Karl Wiegers and Joy Beatty. *Software Requirements*. 3rd. Microsoft Press, 2013.

- 
- [Sho14] Adam Shostack. *Threat Modeling: Designing for Security*. Wiley, 2014.
- [Mar17] Robert Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017.
- [Fow18] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. 2nd. Addison-Wesley, 2018.
- [Fow19] Martin Fowler. *Patterns for Managing Source Code Branches*. Artículo fundamental sobre flujos modernos. 2019. URL: <https://martinfowler.com/articles/branching-patterns.html>.
- [PM19] Roger Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 9th. McGraw-Hill, 2019.
- [Sho19] William E. Shotts. *The Linux Command Line*. 5th. No Starch Press, 2019.
- [Som20] Ian Sommerville. *Software Engineering*. 10th. Pearson, 2020.
- [Ker21] Michael Kerrisk. *The Linux Programming Interface*. No Starch Press, 2021.
- [New21] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. 2nd. O'Reilly Media, 2021.
- [CS23] Scott Chacon and Ben Straub. *Pro Git*. 2nd. Disponible en <https://git-scm.com/book/en/v2>. Apress, 2023.
- [Fou23] Free Software Foundation. *Bash Reference Manual*. Documentación oficial de Bash. 2023. URL: <https://www.gnu.org/software/bash/manual/>.
- [Mic23] Microsoft. *Azure DevOps Labs: Git Exercises*. Recursos prácticos y tutoriales. 2023. URL: <https://www.azuredevopslabs.com/>.
- [RN24] Arnold Robbins and Andrew Nelson. *GAWK: Effective AWK Programming*. Guía oficial de GAWK. 2024. URL: <https://www.gnu.org/software/gawk/manual/>.
- [TH24] Linus Torvalds and Junio Hamano. *Git Documentation*. Documentación oficial. 2024. URL: <https://git-scm.com/doc>.

## 1. CURSO

CS2S1-CS2023. Sistemas Operativos (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS2S1-CS2023. Sistemas Operativos
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS221. Arquitectura de Computadores. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y la máquina.

El propósito de un sistema operativo es proveer un ambiente en que el usuario pueda ejecutar sus aplicaciones.

En este curso se estudiará el diseño del núcleo de los sistemas operativos. Además el curso contempla actividades prácticas en donde se resolverán problemas de concurrencia y se modificará el funcionamiento de un pseudo Sistema Operativo.

## 5. OBJETIVOS

- Conocer los elementos básicos del diseño de los sistemas operativos.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Visión general de Sistemas Operativos (3 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Papel y el propósito del sistema operativo.</li> <li>• Funcionalidad de un sistema operativo típico.</li> <li>• Los mecanismos de apoyo modelos cliente-servidor, dispositivos de mano.</li> <li>• Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad)</li> <li>• Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse]</li> <li>• Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar]</li> <li>• Describir las funciones de un sistema operativo contemporaneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse]</li> <li>• Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse]</li> <li>• Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 2: Principios de Sistemas Operativos (6 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Métodos de estructuración (monolítico, capas, modular, los modelos micro-kernel)</li> <li>• Abstracciones, procesos y recursos.</li> <li>• Los conceptos de interfaces de programa de aplicación (API)</li> <li>• La evolución de las técnicas de hardware / software y las necesidades de aplicación</li> <li>• Organización de dispositivos.</li> <li>• Interrupciones: métodos e implementaciones.</li> <li>• Concepto de usuario de estado / sistema y la protección, la transición al modo kernel.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de una capa lógica [Familiarizarse]</li> <li>• Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse]</li> <li>• Describir el valor de la API y <i>middleware</i> [Familiarizarse]</li> <li>• Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse]</li> <li>• Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar]</li> <li>• Discutir las ventajas y desventajas del uso de procesamiento interrumpido [Familiarizarse]</li> <li>• Explicar el concepto de una capa lógica [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 3: Concurrency (9 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Diagramas de estado.</li> <li>• Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente)</li> <li>• Despacho y cambio de contexto.</li> <li>• El papel de las interrupciones.</li> <li>• Gestionar el acceso a los objetos del sistema operativo atómica.</li> <li>• La implementación de primitivas de sincronización.</li> <li>• Cuestiones multiprocesador (spin-locks, reentrada)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse]</li> <li>• Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar]</li> <li>• Resumir el rango de mecanismos que pueden ser usados a nivel del sistema operativo para realizar sistemas concurrentes y describir los beneficios de cada uno [Familiarizarse]</li> <li>• Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse]</li> <li>• Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse]</li> <li>• Describir las razones para usar interruptores, despacho, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse]</li> <li>• Crear diagramas de estado y transición para los dominios de problemas simples [Usar]</li> </ul>

**Lecturas :** [Avi12], [Sta05], [Tan06], [AD14]

<b>Unidad 4: Planificación y despacho (6 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Planificación preventiva y no preferente.</li> <li>• Planificadores y políticas.</li> <li>• Procesos y subprocesos.</li> <li>• Plazos y cuestiones en tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar y contrastar los algoritmos comunes que se utilizan tanto para un programa preferente y no preferente de las tareas en los sistemas operativos, como la comparación de prioridad, el rendimiento, y los esquemas de distribución equitativa [Evaluar]</li> <li>• Describir las relaciones entre los algoritmos de planificación y dominios de aplicación [Familiarizarse]</li> <li>• Discutir los tipos de planeamiento de procesos <i>scheduling</i> de corto, a mediano, a largo plazo y I/O [Familiarizarse]</li> <li>• Describir las diferencias entre procesos y hebras [Familiarizarse]</li> <li>• Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar]</li> <li>• Hablar sobre la necesidad de tiempos límites de <i>scheduling</i> [Familiarizarse]</li> <li>• Identificar formas en que la lógica expresada en algoritmos de planificación son de aplicación a otros ámbitos, tales como I/O del disco, la programación de disco de red, programación de proyectos y problemas más allá de la computación [Familiarizarse]</li> </ul>

**Lecturas :** [Avi12], [Sta05], [Tan06], [AD14]

<b>Unidad 5: Manejo de memoria (6 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Revisión de la memoria física y hardware de gestión de memoria.</li> <li>Conjuntos de trabajo y thrashing.</li> <li>El almacenamiento en caché</li> </ul> <p><b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]</p>	

<b>Unidad 6: Seguridad y protección (6 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>

- Visión general de la seguridad del sistema .
- Política / mecanismo de separación.
- Métodos de seguridad y dispositivos.
- Protección, control de acceso y autenticación.
- Las copias de seguridad.

- Articular la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse]
- Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse]
- Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse]
- Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarizarse]

**Lecturas :** [Avi12], [Sta05], [Tan06], [AD14]

<b>Unidad 7: Máquinas virtuales (6 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red)</li> <li>• Paginación y la memoria virtual.</li> <li>• Sistemas de archivos virtuales.</li> <li>• Los Hypervisor.</li> <li>• Virtualización portátil; emulación vs aislamiento.</li> <li>• Costo de la virtualización.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse]</li> <li>• Diferenciar emulación y el aislamiento [Familiarizarse]</li> <li>• Evaluar virtualización de compensaciones [Evaluar]</li> <li>• Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 8: Manejo de dispositivos (6 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Características de los dispositivos serie y paralelo.</li> <li>• Haciendo de abstracción de dispositivos.</li> <li>• Estrategias de buffering.</li> <li>• Acceso directo a memoria.</li> <li>• La recuperación de fallos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse]</li> <li>• Identificar los requerimientos para recuperación de errores [Familiarizarse]</li> <li>• Explique <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse]</li> <li>• Diferenciar los mecanismos utilizados en la interconexión de un rango de dispositivos (incluyendo dispositivos portátiles, redes, multimedia) a un ordenador y explicar las implicaciones de éstas para el diseño de un sistema operativo [Familiarizarse]</li> <li>• Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse]</li> <li>• Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse]</li> <li>• Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 9: Sistema de archivos (6 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial.</li> <li>Directorios: contenido y estructura.</li> <li>Los sistemas de archivos: partición, montar sistemas de archivos / desmontar, virtuales.</li> <li>Técnicas estándar de implementación .</li> <li>Archivos asignados en memoria.</li> <li>Sistemas de archivos de propósito especial.</li> <li>Naming, búsqueda, acceso, copias de seguridad.</li> <li>La bitácora y los sistemas de archivos estructurados (log)</li> </ul>	<ul style="list-style-type: none"> <li>Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse]</li> <li>Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar]</li> <li>Resumir cómo el desarrollo de hardware ha dado lugar a cambios en las prioridades para el diseño y la gestión de sistemas de archivos [Familiarizarse]</li> <li>Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 10: Sistemas empotrados y de tiempo real (6 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Proceso y programación de tareas.</li> <li>Los requisitos de gestión de memoria / disco en un entorno en tiempo real.</li> <li>Los fracasos, los riesgos y la recuperación.</li> <li>Preocupaciones especiales en sistemas de tiempo real.</li> </ul>	<ul style="list-style-type: none"> <li>Describir que hace a un sistema un sistema en tiempo real [Familiarizarse]</li> <li>Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse]</li> <li>Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 11: Tolerancia a fallas (3 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Conceptos fundamentales: sistemas fiables y disponibles.</li> <li>Redundancia espacial y temporal.</li> <li>Los métodos utilizados para implementar la tolerancia a fallos.</li> <li>Los ejemplos de los mecanismos del sistema operativo para la detección, recuperación, reinicio para implementar la tolerancia a fallos, el uso de estas técnicas para los servicios propios del sistema operativo.</li> </ul>	<ul style="list-style-type: none"> <li>Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse]</li> <li>Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse]</li> <li>Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

<b>Unidad 12: Evaluación del desempeño de sistemas (3 horas)</b>	
<b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• ¿Por qué el rendimiento del sistema debe ser evaluado?</li> <li>• ¿Qué se va a evaluar?</li> <li>• Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad.</li> <li>• Modelos de evaluación: analítica, simulación, o de implementación específico determinista.</li> <li>• Cómo recoger los datos de evaluación (perfiles y mecanismos de localización)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse]</li> <li>• Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Avi12], [Sta05], [Tan06], [AD14]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Sta05] William Stallings. *Operating Systems: Internals and Design Principles*, 5/E. Prentice Hall, 2005.
- [Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation*, 3/E. Prentice Hall, 2006.
- [Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts*, 9/E. John Wiley & Sons, Inc., 2012.
- [AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice*. 2nd. Recursive Books, 2014.

## 1. CURSO

ID105. Inglés V (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	ID105. Inglés V
<b>2.2 Semestre</b>	:	5 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BEI202. Inglés IV. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Parte fundamental de la formación integral de un profesional es la habilidad de comunicarse en un idioma extranjero además del propio idioma nativo. No solamente amplía su horizonte cultural sino que permite una visión más humana y comprensiva de la vida. En el caso de los idiomas extranjeros, indudablemente el Inglés es el más práctico porque es hablado alrededor de todo el mundo. No hay país alguno donde éste no sea hablado. En las carreras relacionadas con los servicios al turista el inglés es tal vez la herramienta práctica más importante que el alumno debe dominar desde el primer momento como parte de su formación integral.

## 5. OBJETIVOS

- Incrementar la capacidad y la fluidez de hablar y entender el idioma Inglés.
- Hacer que los alumnos interactúen con mayor énfasis en la creación de diálogos.

## 6. RESULTADOS DEL ESTUDIANTE

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Assessment)

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

## 7. TEMAS

<b>Unidad 1: It's a wonderful world (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Verbos Auxiliares</li> <li>• Tiempos Verbales</li> <li>• Preguntas y oraciones negativas</li> <li>• Respuestas cortas</li> <li>• Formación de palabras</li> <li>• Expresiones coloquiales</li> <li>• Corrección de errores</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la primera unidad, cada uno de los alumnos, comprendiendo la gramática de los auxiliares y de los diferentes tipos de oración es capaz de expresar una mayor cantidad de expresiones de tiempo y además usar preposiciones para describir lugares y tiempos variados. Además es capaz de analizar y expresar ideas acerca de formación de palabras.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

<b>Unidad 2: Happiness! (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Presente Simple</li> <li>• Presente Continuo</li> <li>• Voz Pasiva en Presente</li> <li>• Verbos para deportes y tiempo libre</li> <li>• Tipos de números y fechas</li> <li>• Inventos/Mundo Moderno</li> <li>• Corrección de errores</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la segunda unidad, los alumnos habiendo identificado la forma de expresar deportes y actividades de tiempo libre. Utiliza todo tipo de expresiones numéricas. Expresar situaciones y estados relacionados con formas de presente. Explica y aplica vocabulario de actividades al aire libre.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

<b>Unidad 3: Telling tales! (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempo Pasado Simple</li> <li>• Pasado Continuo</li> <li>• Voz Pasiva en Pasado</li> <li>• Vocabulario de Arte y Literatura</li> <li>• Expresiones para dar y pedir opiniones</li> <li>• Cuentos e historias</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la tercera unidad, los alumnos habiendo reconocido las características de las formas de pasado pasivo, utilizan éstos para hacer descripciones de diversos tipos. Describen arte y literatura y dan indicaciones de opinión. Utilizarán conjunciones para unir ideas tipo.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

<b>Unidad 4: Doing the right thing! (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Verbos Auxiliares de Modo I</li> <li>• Oraciones Afirmativas, Negativas y Preguntas con Modals</li> <li>• Uso de nacionalidades i otros adjetivos</li> <li>• Expresiones de pedidos y Ofrecimientos</li> <li>• Guía de los Buenos Modales</li> <li>• Llenado de Formatos</li> <li>• Símbolos fonéticos</li> </ul>	<ul style="list-style-type: none"> <li>• Al terminar la cuarta unidad, los alumnos habiendo identificado la idea de expresar ideas de modos de acciones que suceden en el momento o que se relacionan a cualquier tiempo estructuran oraciones en Presente. Expresan ideas de nacionalidades y hacen pedidos y ofrecimientos variados.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

<b>Unidad 5: On the move! (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Futuro con Hill</li> <li>• Oraciones en Tiempo Futuro con going to</li> <li>• Uso de might para futuro</li> <li>• Expresiones del clima</li> <li>• Vocabulario del clima</li> <li>• Expresiones para hoteles y transporte</li> <li>• E-mails</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la quinta unidad, los alumnos, a partir de la comprensión del tiempo futuro, elaborarán oraciones utilizando los elementos necesarios. Asimilarán además la necesidad de expresar ideas del clima. Adquirirán vocabulario para describir uso de transporte público. Se presentará expresiones para hacer pedidos en hoteles.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

<b>Unidad 6: I just love it! (0 horas)</b> <b>Resultados esperados: 5</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Preguntas con formas Like</li> <li>• Patrones Verbales II</li> <li>• Vocabulario de Comida, Lugares y ocupaciones</li> <li>• Palabras que van unidas en contexto</li> <li>• Expresiones para vistas y sonidos</li> <li>• Composición de Impresiones personales</li> </ul>	<ul style="list-style-type: none"> <li>• Al finalizar la sexta unidad, los alumnos habiendo conocido los fundamentos de la estructuración de preguntas con like y con patrones verbales trabajos aplicados a contextos adecuados. Enfatizan la diferencia entre comidas, lugares y personas. Describen vistas y sonidos. Utilizan expresiones comparar la vida diaria en diferentes lugares. Asumen la idea de estilos de vida diferentes.</li> </ul>
<b>Lecturas :</b> [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### **10. BIBLIOGRAFÍA BÁSICA**

- [Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.
- [SJ02a] Liz Soars and John. *American Headway N 3 Student Book*. Editorial Oxford, 2002.
- [SJ02b] Liz Soars and John. *American Headway N 3 Teachers Book*. Editorial Oxford, 2002.
- [SJ02c] Liz Soars and John. *American Headway N 3 Work Book*. Editorial Oxford, 2002.
- [Cam06] Cambridge. *Diccionario Inglés-Español Cambridge*. Editorial Oxford, 2006.



## Capítulo 6

### Sexto Semestre

## 1. CURSO

CS231-CS2023. Redes y Comunicación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS231-CS2023. Redes y Comunicación
<b>2.2 Semestre</b>	:	6 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS2S1. Sistemas Operativos. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El siempre creciente desarrollo de las tecnologías de comunicación y la información hace que exista una marcada tendencia a establecer más redes de computadores que permitan una mejor gestión de la información.

En este segundo curso se brindará a los participantes una introducción a los problemas que conlleva la comunicación entre computadores, a través del estudio e implementación de protocolos de comunicación como TCP/IP y la implementación de software sobre estos protocolos.

## 5. OBJETIVOS

- Que el alumno implemente y/o modifique un protocolo de comunicación de datos.
- Que el alumno domine las técnicas de transmisión de datos utilizadas por los protocolos de red existentes.
- Que el alumno conozca las ultimas tendencias en redes que se están aplicando en el Internet.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Introducción a redes (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc)</li> <li>Técnicas de Switching (por ejemplo, de circuitos, de paquetes)</li> <li>Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls.</li> <li>Principios de capas (encapsulación, multiplexación)</li> <li>Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)</li> </ul>	<ul style="list-style-type: none"> <li>Articular la organización de la Internet [Familiarizarse]</li> <li>Listar y definir la terminología de red apropiada [Familiarizarse]</li> <li>Describir la estructura en capas de una arquitectura típica en red [Familiarizarse]</li> <li>Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]</li> </ul>

**Lecturas :** [KR13]

<b>Unidad 2: Aplicaciones en red (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Esquemas de denominación y dirección (DNS, direcciones IP, identificadores de recursos uniformes, etc)</li> <li>Las aplicaciones distribuidas (cliente / servidor, peer-to-peer, nube, etc)</li> <li>HTTP como protocolo de capa de aplicación .</li> <li>Multiplexación con TCP y UDP</li> <li>API de Socket</li> </ul>	<ul style="list-style-type: none"> <li>Listar las diferencias y las relaciones entre los nombres y direcciones en una red [Familiarizarse]</li> <li>Definir los principios detrás de esquemas de denominación y ubicación del recurso [Familiarizarse]</li> <li>Implementar una aplicación simple cliente-servidor basada en <i>sockets</i> [Usar]</li> </ul>

**Lecturas :** [KR13]

<b>Unidad 3: Entrega confiable de datos (10 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Control de errores (técnicas de retransmisión, temporizadores)</li> <li>El control de flujo (agradecimientos, ventana deslizante)</li> <li>Problemas de rendimiento (pipelining)</li> <li>TCP</li> </ul>	<ul style="list-style-type: none"> <li>Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse]</li> <li>Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse]</li> <li>Diseñar e implementar un protocolo confiable simple [Usar]</li> </ul>

**Lecturas :** [KR13]

<b>Unidad 4: Ruteo y reenvío (12 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Enrutamiento vs reenvío .</li> <li>• Enrutamiento estático .</li> <li>• Protocolo de Internet (IP)</li> <li>• Problemas de escalabilidad (direcciónamiento jerárquico)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la organización de la capa de red [Familiarizarse]</li> <li>• Describir cómo los paquetes se envían en una red IP [Familiarizarse]</li> <li>• Listar las ventajas de escalabilidad de direcciónamiento jerárquico [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [KR13]	
<b>Unidad 5: Redes de área local (10 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Problemas de Acceso Múltiple.</li> <li>• Enfoques comunes a Acceso múltiple (exponencial backoff, multiplexación por división de tiempo, etc)</li> <li>• Redes de área local .</li> <li>• Ethernet .</li> <li>• Switching .</li> </ul>	<ul style="list-style-type: none"> <li>• Describir como los paquetes son enviados en una red Ethernet [Familiarizarse]</li> <li>• Describir las diferencias entre IP y Ethernet [Familiarizarse]</li> <li>• Describir las etapas usadas en un enfoque común para el problema de múltiples accesos [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [KR13]	
<b>Unidad 6: Asignación de recursos (12 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Necesidad de asignación de recursos .</li> <li>• Asignación fija (TDM, FDM, WDM) versus la asignación dinámica .</li> <li>• De extremo a extremo frente a las red de enfoque asistida .</li> <li>• Justicia.</li> <li>• Principios del control de congestión.</li> <li>• Enfoques para la congestión (por ejemplo, redes de distribución de contenidos)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir como los recursos pueden ser almacenados en la red [Familiarizarse]</li> <li>• Describir los problemas de congestión en una red grande [Familiarizarse]</li> <li>• Comparar y contrastar las técnicas de almacenamiento estático y dinámico [Familiarizarse]</li> <li>• Comparar y contrastar los enfoques actuales de la congestión [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [KR13]	
<b>Unidad 7: Celulares (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Principios de redes celulares.</li> <li>• Redes 802.11</li> <li>• Problemas en el apoyo a los nodos móviles (agente local)</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la organización de una red inalámbrica [Familiarizarse]</li> <li>• Describir como las redes inalámbricas soportan usuarios móviles [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [KR13], [Cha16]	

<b>Unidad 8: Redes sociales (5 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Panorama de las redes sociales.</li> <li>• Ejemplo plataformas de redes sociales.</li> <li>• Estructura de los grafos de redes sociales.</li> <li>• Análisis de redes sociales.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir los principios fundamentales (como pertenencia, confianza) de una red social [Familiarizarse]</li> <li>• Describir como redes sociales existentes operan [Familiarizarse]</li> <li>• Construir un grafo de una red social a partir de datos de la red [Usar]</li> <li>• Analizar una red social para determinar quienes son las personas importantes [Usar]</li> <li>• Evaluar una determinada interpretación de una pregunta de red social con los datos asociados [Familiarizarse]</li> </ul>

**Lecturas :** [KR13], [Kad11]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011.
- [KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013.
- [Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016.

## 1. CURSO

CS311-CS2023. Programación Competitiva (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS311-CS2023. Programación Competitiva
<b>2.2 Semestre</b>	:	6 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS212. Análisis y Diseño de Algoritmos. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La Programación Competitiva combina retos de solucionar problemas con el añadido de poder competir con otras personas. Enseña a los participantes a pensar más rápido y desarrollar habilidades para resolver problemas, que son de gran demanda en la industria. Este curso enseñará la resolución de problemas algorítmicos de manera rápida combinando la teoría de algoritmos y estructuras de datos con la práctica la solución de los problemas.

## 5. OBJETIVOS

- Que el alumno utilice técnicas de estructuras de datos y algoritmos complejos.
- Que el alumno aplique los conceptos aprendidos para la aplicación sobre un problema real.
- Que el alumno investigue la posibilidad de crear un nuevo algoritmo y/o técnica nueva para resolver un problema real.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Introducción (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Introducción a la Programación competitiva</li> <li>• Modelo computacional</li> <li>• Complejidad algorítmica</li> <li>• Problemas sobre búsqueda y ordenamiento</li> <li>• Recursión y recurrencia</li> <li>• Estrategia divide y conquista</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer y saber como usar los recursos del modelo de computación RAM (Random Access Machine). [Usar]</li> <li>• Determinar el tiempo y espacio de complejidad de algoritmos. [Usar]</li> <li>• Determinar relaciones de recurrencia para algoritmos recursivos. [Usar]</li> <li>• Resolver problemas de búsqueda y ordenamiento. [Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para problemas de tipo divide y conquista. [Usar]</li> <li>• Diseñar nuevos algoritmos para la resolución de problemas. [Usar]</li> </ul>

**Lecturas :** [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]

<b>Unidad 2: Estructuras de datos (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Problemas sobre arrays y strings</li> <li>• Problemas sobre listas enlazadas</li> <li>• Problemas sobre pilas, colas</li> <li>• Problemas sobre árboles</li> <li>• Problemas sobre Hash tables</li> <li>• Problemas sobre Heaps</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer las distintas estructuras de datos sus complejidades usos y restricciones. [Usar]</li> <li>• Identificar el tipo de estructura de datos adecuado a la resolución del problema. [Usar]</li> <li>• Reconocer tipos de problemas asociado a operaciones sobre estructuras de datos como búsqueda, inserción, eliminación y actualización. [Usar]</li> </ul>

**Lecturas :** [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]

<b>Unidad 3: Paradigmas de diseño (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fuerza bruta</li> <li>• Divide y conquista</li> <li>• Backtracking</li> <li>• Greedy</li> <li>• Programación Dinámica</li> </ul>	<ul style="list-style-type: none"> <li>• Aprender los distintos paradigmas de resolución de problemas. [Usar]</li> <li>• Aprender a seleccionar los algoritmos adecuados para distintos problemas según el tipo de paradigma. [Usar]</li> </ul>

**Lecturas :** [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]

<b>Unidad 4: Gráficos (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Recorrido de gráficos</li> <li>• Aplicaciones y problemas sobre gráficos</li> <li>• Camino mas corto</li> <li>• Redes y flujos</li> </ul>	
<b>Lecturas :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	
<b>Unidad 5: Tópicos avanzados (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Teoría de números</li> <li>• Probabilidad y combinaciones</li> <li>• Algoritmos para manejos de strings (tries, string hashing, z-algorithm)</li> <li>• Geometría y sweep line algorithms, segment trees</li> </ul>	
<b>Lecturas :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	
<b>Unidad 6: Problemas de dominio específico (20 horas)</b> <b>Resultados esperados: 1</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Latencia y rendimiento</li> <li>• Paralelismo</li> <li>• Redes</li> <li>• Almacenamiento</li> <li>• Alta disponibilidad</li> <li>• Caching</li> <li>• Proxies</li> <li>• Equilibradores de carga</li> <li>• Almacenamiento clave-valor</li> <li>• Replicar y compartir</li> <li>• Elección del líder</li> <li>• Limitación de la tasa</li> <li>• Registro y monitoreo</li> </ul>	
<b>Lecturas :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## **10. BIBLIOGRAFÍA BÁSICA**

- [Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003.
- [Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.
- [ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOf-ProgrammingInterviews.com, 2012. URL: <https://books.google.com.pe/books?id=y6FLBQAAQBAJ>.
- [Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.
- [Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.
- [Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.

## 1. CURSO

CS312-CS2023. Estructuras de Datos Avanzadas (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS312-CS2023. Estructuras de Datos Avanzadas
<b>2.2 Semestre</b>	:	6 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	<ul style="list-style-type: none"><li>CS212. Análisis y Diseño de Algoritmos. (5<sup>to</sup> Sem)</li><li>BEI202. Inglés IV. (4<sup>to</sup> Sem)</li></ul>

## 3. PROFESORES

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 permiten organizar la información de una manera más eficiente, por lo que es importante para todo profesional del área tener una sólida formación en este aspecto.

En el curso de estructuras de datos avanzadas nuestro objetivo es que el alumno conozca y analice estructuras complejas, como los Métodos de Acceso Multidimensional, Métodos de Acceso Espacio-Temporal y Métodos de Acceso Métrico, etc.

## 5. OBJETIVOS

- Que el alumno entienda, diseñe, implemente, aplique y proponga estructuras de datos innovadoras para solucionar problemas relacionados al tratamiento de datos multidimensionales, recuperación de información por similitud, motores de búsqueda y otros problemas computacionales.

## 6. RESULTADOS DEL ESTUDIANTE

- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

- 2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Multidimensional Data (16 horas)</b> <b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Introducción al curso.</li> <li>• Introducción a datos multidimensionales.</li> <li>• Maldición de la dimensionalidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir la transcendencia de la representación multidimensional de datos. [Usar]</li> <li>• Entender la complejidad de lidiar con datos multidimensional y de alta dimensión.[Usar]</li> <li>• Entender la maldición de la dimensionalidad, y su impacto en el indizado de grandes volúmenes de datos.[Usar]</li> <li>• Presentar y discutir aplicaciones reales de datos multidimensionales en motores de búsqueda.[Usar]</li> </ul>
<b>Lecturas :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

<b>Unidad 2: Multidimensional Acces Data Structures (16 horas)</b> <b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Introducción a estructuras de datos espaciales.</li> <li>• Estructuras espaciales, Quadtree,Octree y visualización.</li> <li>• Kd-Tree.</li> <li>• Introducción a R-Tress.</li> <li>• R tree (Guttman).</li> <li>• R+ tree.</li> <li>• R* tree.</li> <li>• Variación R*-tree y relación con paginación y tamaño de bloques.</li> <li>• X-tree.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir los fundamentos teóricos de estructuras de datos espaciales.</li> <li>• Entender los beneficios y limitaciones de estructuras de datos espaciales basadas en árbol.</li> <li>• Implementar diferentes estructuras de datos para el indizado de grandes volúmenes de datos.</li> <li>• Entender los fundamentos e implementar estrategias de búsqueda como vecinos mas próximos y búsquedas por rango.</li> </ul>
<b>Lecturas :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

<b>Unidad 3: Approximate Access Methods (20 horas)</b> <b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Métodos de Acceso Métrico para distancias discretas</li> <li>• Métodos de Acceso Métrico para distancias continuas</li> </ul>	<ul style="list-style-type: none"> <li>• Que el alumno entienda conozca e implemente algunos métodos de acceso métrico[Usar]</li> <li>• Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud[Usar]</li> </ul>
<b>Lecturas :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

<b>Unidad 4: Métodos de Acceso Aproximados (20 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Space Filling Curves: Hilbert curve y Z-order</li> <li>• Proyecciones y complejidad.</li> <li>• Locally sensitive hashing (LSH)</li> </ul>	<ul style="list-style-type: none"> <li>• Entender, conocer e implementar algunos métodos de acceso aproximados.</li> <li>• Entender la importancia de estos métodos de Acceso para la recuperación de información por similitud en entornos donde la escalabilidad sea un factor muy importante.</li> </ul>

**Lecturas :** [Sam06], [PI06], [Zez+07]

<b>Unidad 5: Clustering (8 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Introducción a Clustering.</li> <li>• Kmeans y DBScan.</li> <li>• Clustering Applications.</li> <li>• Clustering Ensemble.</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir los fundamentos teóricos para el clustering de datos multidimensionales.</li> <li>• Implementar diferentes estrategias para el clustering de datos multidimensionales, como basados en partición, en jerarquía o en densidad.</li> <li>• Entender los fundamentos, aplicaciones e implementar ensambles de métodos de clustering.</li> <li>• Implementar ensambles de métodos de clustering con datos reales.</li> </ul>

**Lecturas :** [Cua+04], [Knu07a], [Knu07b], [Gam+94]

<b>Unidad 6: Temporal Data Structures (8 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Introducción a Estructuras de datos temporales.</li> <li>• Versionando la estructura de Datos.</li> <li>• Persistencia</li> <li>• Retroactividad</li> </ul>	<ul style="list-style-type: none"> <li>• Introducir los fundamentos teóricos de estructuras de datos temporales.</li> <li>• Entender, discutir e implementar Persistencia y sus tipos.</li> <li>• Entender, discutir e implementar Retroactividad y sus tipos.</li> <li>• Entender y discutir los beneficios y limitaciones entre persistencia y retroactividad.</li> </ul>

**Lecturas :** [Cua+04], [Knu07a], [Knu07b], [Gam+94]

<b>Unidad 7: Final Talks (8 horas)</b>	
<b>Resultados esperados: 1,2</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Seminarios de trabajo de investigación.</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar sobre nuevos métodos para el indizado de grandes volúmenes de datos complejos.</li> <li>• Presentar y dirigir la discusión sobre métodos para indizados de Big Data investigado.</li> </ul>

**Lecturas :** [Cua+04], [Knu07a], [Knu07b], [Gam+94]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.
- [Cua+04] Ernesto Cuadros-Vargas et al. "Implementing data structures: An incremental approach". <http://socios.spc.org.pe/ecuad04.pdf>
- [PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.
- [Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Illustrated. Elsevier/Morgan Kaufmann, Aug. 2006. URL: <http://books.google.com.pe/books?id=v0-NRRKHG84C>.
- [Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms*. 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.
- [Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching*. 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.
- [Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach*. 1st. ISBN-10: 0387291466. Springer, Nov. 2007.

## 1. CURSO

CS342-CS2023. Compiladores (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS342-CS2023. Compiladores
<b>2.2 Semestre</b>	:	6 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS211. Teoría de la Computación. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Que el alumno conozca y comprenda los conceptos y principios fundamentales de la teoría de compilación para realizar la construcción de un compilador

## 5. OBJETIVOS

- Conocer las técnicas básicas empleadas durante el proceso de generación intermedio, optimización y generación de código.
- Aprender a implementar pequeños compiladores.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

**Unidad 1: Representación de programas (5 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Programas que tienen otros programas como entrada tales como interpretadores, compiladores, revisores de tipos y generadores de documentación.</li> <li>• Árboles de sintaxis abstracta, para contrastar la sintaxis correcta.</li> <li>• Estructuras de datos que representan código para ejecución, traducción o transmisión.</li> <li>• Compilación en tiempo just-in time y re-compilación dinámica.</li> <li>• Otras características comunes de las máquinas virtuales, tales como carga de clases, hilos y seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse]</li> <li>• Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Familiarizarse]</li> <li>• Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse]</li> <li>• Escribir un programa para procesar alguna representación de código para algún propósito, tales como un interpretador, una expresión optimizada, o un generador de documentación [Familiarizarse]</li> <li>• Explicar el uso de metadatos en las representaciones de tiempo de ejecución de objetos y registros de activación, tales como los punteros de la clase, las longitudes de arreglos, direcciones de retorno, y punteros de <i>frame</i> [Familiarizarse]</li> <li>• Discutir las ventajas, desventajas y dificultades del término (<i>just-in-time</i>) y recompilación automática [Familiarizarse]</li> <li>• Identificar los servicios proporcionados por los sistemas de tiempo de ejecución en lenguajes modernos [Familiarizarse]</li> </ul>

**Lecturas :** [Lou04b]

## Unidad 2: Traducción y ejecución de lenguajes (10 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje (Learning Outcomes)
<ul style="list-style-type: none"> <li>Interpretación vs. compilación a código nativo vs. compilación de representación portable intermedia.</li> <li>Pipeline de traducción de lenguajes: análisis, revisión opcional de tipos, traducción, enlazamiento, ejecución: <ul style="list-style-type: none"> <li>Ejecución como código nativo o con una máquina virtual</li> <li>Alternativas como carga dinámica y codificación dinámica de código (o “just-in-time””)</li> </ul> </li> <li>Representación en tiempo de ejecución de construcción del lenguaje núcleo tales como objetos (tablas de métodos) y funciones de primera clase (cerradas)</li> <li>Ejecución en tiempo real de asignación de memoria: pila de llamas, montículo, datos estáticos: <ul style="list-style-type: none"> <li>Implementación de bucles, recursividad y llamadas de cola</li> </ul> </li> <li>Gestión de memoria: <ul style="list-style-type: none"> <li>Gestión manual de memoria: asignación, limpieza y reuso de la pila de memoria</li> <li>Gestión automática de memoria: recolección de datos no utilizados (<i>garbage collection</i>) como una técnica automática usando la noción de accesibilidad</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Distinguir una definición de un lenguaje de una implementación particular de un lenguaje (compilador vs interprete, tiempo de ejecución de la representación de los objetos de datos, etc) [Evaluar]</li> <li>Distinguir sintaxis y parseo de la semántica y la evaluación [Evaluar]</li> <li>Bosqueje una representación de bajo nivel de tiempo de ejecución de construcciones del lenguaje base, tales como objetos o cierres (<i>closures</i>) [Evaluar]</li> <li>Explicar cómo las implementaciones de los lenguajes de programación tipicamente organizan la memoria en datos globales, texto, <i>heap</i>, y secciones de pila y cómo las características tales como recursión y administración de memoria son mapeados a este modelo de memoria [Evaluar]</li> <li>Identificar y corregir las pérdidas de memoria y punteros desreferenciados [Evaluar]</li> <li>Discutir los beneficios y limitaciones de la recolección de basura (<i>garbage collection</i>), incluyendo la noción de accesibilidad [Evaluar]</li> </ul>

Lecturas : [Aho+11], [Lou04a], [TS98], [App02]

## Unidad 3: Análisis de sintaxis (10 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje (Learning Outcomes)
<ul style="list-style-type: none"> <li>Exploración (análisis léxico) usando expresiones regulares.</li> <li>Estrategias de análisis incluyendo técnicas de arriba a abajo (top-down) (p.e. descenso recursivo, análisis temprano o LL) y de abajo a arriba (bottom-up) (ej, ‘llamadas hacia atrás - backtracking, o LR); rol de las gramáticas libres de contexto.</li> <li>Generación de exploradores (scanners) y analizadores a partir de especificaciones declarativas.</li> </ul>	<ul style="list-style-type: none"> <li>Usar gramáticas formales para especificar la sintaxis de los lenguajes [Evaluar]</li> <li>Usar herramientas declarativas para generar parseadores y escáneres [Evaluar]</li> <li>Identificar las características clave en las definiciones de sintaxis: ambigüedad, asociatividad, precedencia [Evaluar]</li> </ul>

Lecturas : [Aho+11], [Lou04a], [TS98], [App02]

<b>Unidad 4: Análisis semántico de compiladores (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Representaciones de programas de alto nivel tales como árboles de sintaxis abstractas.</li> <li>• Alcance y resolución de vínculos.</li> <li>• Revisión de tipos.</li> <li>• Especificaciones declarativas tales como gramáticas atribuídas.</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar analizadores sensibles al contexto y estáticos a nivel de fuente, tales como, verificadores de tipos o resolvedores de identificadores para identificar las ocurrencias de vínculo [Evaluar]</li> <li>• Describir analizadores semánticos usando una gramática con atributos [Evaluar]</li> </ul>
<b>Lecturas :</b> [Aho+11], [Lou04a], [TS98], [App02]	

<b>Unidad 5: Generación de código (20 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Llamadas a procedimientos y métodos en envío.</li> <li>• Compilación separada; vinculación.</li> <li>• Selección de instrucciones.</li> <li>• Calendarización de instrucciones.</li> <li>• Asignación de registros.</li> <li>• Optimización por rendija (peephole)</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar todos los pasos esenciales para convertir automáticamente código fuente en código ensamblador o otros lenguajes de bajo nivel [Evaluar]</li> <li>• Generar código de bajo nivel para llamadas a funciones en lenguajes modernos [Evaluar]</li> <li>• Discutir por qué la compilación separada requiere convenciones de llamadas uniformes [Evaluar]</li> <li>• Discutir por qué la compilación separada limita la optimización debido a efectos de llamadas desconocidas [Evaluar]</li> <li>• Discutir oportunidades para optimización introducida por la traducción y enfoques para alcanzar la optimización, tales como la selección de la instrucción, planificación de instrucción, asignación de registros y optimización de tipo mirilla (<i>peephole optimization</i>) [Evaluar]</li> </ul>
<b>Lecturas :</b> [Aho+11], [Lou04a], [TS98], [App02]	

## 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 plantearamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

## 9. SISTEMA DE EVALUACIÓN

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [TS98] Bernard Teufel and Stephanie Schmidt. *Fundamentos de Compiladores*. Addison Wesley Iberoamericana, 1998.  
 [App02] A. W. Appel. *Modern compiler implementation in Java*. 2.a edición. Cambridge University Press, 2002.

- 
- [Lou04a] Kenneth C. Louden. *Compiler Construction: Principles and Practice*. Thomson, 2004.
  - [Lou04b] Kenneth C. Louden. *Lenguajes de Programacion*. Thomson, 2004.
  - [Aho+11] Alfred Aho et al. *Compilers Principles Techniques And Tools*. 2nd. ISBN:10-970-26-1133-4. Pearson, 2011.

## 1. CURSO

AI263. Introducción al Aprendizaje de Máquina (Obligatorio)

## 2. INFORMACIÓN GENERAL

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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

<b>Unidad 1: Fundamentos de ML (15 horas)</b> <b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Aprendizaje supervisado vs no supervisado</li> <li>• Overfitting y regularización</li> <li>• Validación cruzada y curvas de aprendizaje</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el bias-variance tradeoff [Familiarizarse]</li> <li>• Implementar k-fold cross-validation [Usar]</li> </ul>

<b>Unidad 2: Modelos Lineales (15 horas)</b> <b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Regresión lineal y logística</li> <li>• Máquinas de vectores de soporte (SVM)</li> <li>• Análisis discriminante lineal</li> </ul>	<ul style="list-style-type: none"> <li>• Programar modelos lineales con scikit-learn [Usar]</li> <li>• Interpretar coeficientes de regresión [Evaluar]</li> </ul>

<b>Unidad 3: Ensamblles y Árboles (15 horas)</b> <b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Árboles de decisión y random forests</li> <li>• Gradient Boosting (XGBoost, LightGBM)</li> <li>• Stacking y votación</li> </ul>	<ul style="list-style-type: none"> <li>• Optimizar hiperparámetros con GridSearch [Usar]</li> <li>• Visualizar árboles de decisión [Evaluar]</li> </ul>

<b>Unidad 4: Redes Neuronales Básicas (15 horas)</b> <b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Perceptrones multicapa (MLP)</li> <li>• Backpropagation y optimizadores</li> <li>• Introducción a Keras/TensorFlow</li> </ul>	<ul style="list-style-type: none"> <li>• Construir redes neuronales simples [Usar]</li> <li>• Monitorizar entrenamiento con TensorBoard [Evaluar]</li> </ul>

**Lecturas :** [HTF09], [Gér22]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

---

## 10. BIBLIOGRAFÍA BÁSICA

- [Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. 2nd. Springer, 2009.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [Cho21] François Chollet. *Deep Learning with Python*. 2nd. Manning, 2021.
- [Gér22] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 3rd. O'Reilly, 2022.

## 1. CURSO

FI201FCCS. Física Computacional (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FI201FCCS. Física Computacional
<b>2.2 Semestre</b>	:	6 <sup>to</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BFI101. Física I. (2 <sup>do</sup> 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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Usage)

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

## 7. TEMAS

<b>Unidad 1: Óptica y Propagación de la Luz (10 horas)</b> <b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Naturaleza de la luz.</li> <li>• Reflexión y refracción.</li> <li>• Lentes y espejos.</li> <li>• Interferencia y difracción.</li> <li>• Modelos de iluminación (ej. Phong, Blinn-Phong).</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las propiedades de la luz y su propagación. [Familiarizarse]</li> <li>• Aplicar las leyes de la reflexión y refracción. [Usar]</li> <li>• Implementar modelos de iluminación en gráficos por computadora. [Evaluar]</li> </ul>
<b>Lecturas :</b> [YF18], [Hec17]	

<b>Unidad 2: Colisiones y Transferencia de Energía (8 horas)</b> <b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Impulso y momento lineal.</li> <li>• Colisiones elásticas e inelásticas.</li> <li>• Conservación de la energía en colisiones.</li> <li>• Deformación de mallas elásticas (ej. modelo de masa-resorte).</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar los principios de conservación del momento lineal y la energía en colisiones. [Familiarizarse]</li> <li>• Modelar la deformación de mallas elásticas producto de un impacto. [Usar]</li> <li>• Implementar simulaciones de colisiones en un entorno computacional. [Evaluar]</li> </ul>
<b>Lecturas :</b> [YF18], [Tay05]	

<b>Unidad 3: Mecánica de Sólidos Rígidos (8 horas)</b> <b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Rotación de cuerpos rígidos.</li> <li>• Momento de inercia.</li> <li>• Torque y energía cinética rotacional.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la rotación de cuerpos rígidos. [Familiarizarse]</li> <li>• Calcular el momento de inercia. [Usar]</li> <li>• Aplicar las ecuaciones de la dinámica rotacional. [Evaluar]</li> </ul>
<b>Lecturas :</b> [YF18], [Tay05]	

<b>Unidad 4: Dinámica de Fluidos (6 horas)</b> <b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Principios básicos de la dinámica de fluidos.</li> <li>• Viscosidad.</li> <li>• Flujo laminar y turbulento.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las propiedades de los fluidos. [Familiarizarse]</li> <li>• Explicar los conceptos de viscosidad y flujo laminar/turbulento. [Usar]</li> <li>• Resolver problemas sencillos de dinámica de fluidos. [Evaluar]</li> </ul>
<b>Lecturas :</b> [YF18]	

<b>Unidad 5: Termodinámica (6 horas)</b>	
<b>Resultados esperados: 1,6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Leyes de la termodinámica.</li> <li>• Transferencia de calor.</li> </ul>	<ul style="list-style-type: none"> <li>• Enunciar las leyes de la termodinámica. [Familiarizarse]</li> <li>• Describir los mecanismos de transferencia de calor. [Usar]</li> </ul>

**Lecturas :** [YF18]

<b>Unidad 6: Simulación Física (10 horas)</b>	
<b>Resultados esperados: 1,6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Métodos numéricos para simulación física.</li> <li>• Integración de Verlet.</li> <li>• Detección de colisiones.</li> <li>• Sistemas de partículas.</li> <li>• Restricciones y solvers.</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar métodos numéricos básicos para simulación física. [Familiarizarse]</li> <li>• Utilizar la integración de Verlet para simular el movimiento. [Usar]</li> <li>• Implementar algoritmos de detección de colisiones. [Evaluar]</li> </ul>

**Lecturas :** [Tay05]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Tay05] John R. Taylor. *Classical Mechanics*. University Science Books, 2005.

[Hec17] Eugene Hecht. *Optics*. Pearson, 2017.

[YF18] Hugh D. Young and Roger A. Freedman. *University Physics with Modern Physics*. Pearson, 2018.

## Capítulo 7

# Séptimo Semestre

## 1. CURSO

CS251-CS2023. Computación Gráfica (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS251-CS2023. Computación Gráfica
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	MA202. Métodos Numéricos. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Ofrece una introducción para el área de Computación Gráfica, la cual es una parte importante dentro de Ciencias de la Computación. El propósito de este curso es investigar los principios, técnicas y herramientas fundamentales para esta área.

## 5. OBJETIVOS

- Acercar al alumno a conceptos y técnicas usados en aplicaciones gráficas 3-D complejas.
- Dar al alumno las herramientas necesarias para determinar que software gráfico y que plataforma son los más adecuados para desarrollar una aplicación específica.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Conceptos Fundamentales (6 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual.</li> <li>• Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes.</li> <li>• Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores.</li> <li>• Animación como una secuencia de imágenes fijas.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar en términos generales cómo las señales analógicas pueden ser representadas por muestras discretas, por ejemplo, cómo las imágenes pueden ser representadas por pixeles [Familiarizarse]</li> <li>• Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse]</li> <li>• Describir las ventajas y desventajas entre el almacenamiento de información vs almacenar suficiente información para reproducir la información, como en la diferencia entre el vector y la representación de la trama [Familiarizarse]</li> <li>• Describir los procesos básicos de la producción de movimiento continuo a partir de una secuencia de cuadros discretos (algunas veces llamado it flicker fusion) [Familiarizarse]</li> </ul>

**Lecturas :** [HB90]

## Unidad 2: Rendering Básico (12 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica.</li> <li>• Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)</li> <li>• Radiometría básica, triángulos similares y modelos de proyecciones</li> <li>• Afinamiento y Transformaciones de Sistemas de coordenadas</li> <li>• <i>Ray tracing</i></li> <li>• Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos.</li> <li>• Rasterización triangular simple.</li> <li>• Renderización con una API basada en shader.</li> <li>• Aplicación de la representación de estructuras de datos espaciales.</li> <li>• Muestreo y anti-aliasing.</li> <li>• Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir el problema de transporte de la luz y su relación con la integración numérica, es decir, se emite luz, dispersa alrededor de la escena, y es medida por el ojo [Familiarizarse]</li> <li>• Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse]</li> <li>• Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar]</li> <li>• Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afín [Usar]</li> <li>• Aplicar sistema de coordenadas de 3-dimensiones y los cambios necesarios para extender las operaciones de transformación 2D para manejar las transformaciones en 3D [Usar]</li> <li>• Contrastar la renderización hacia adelante <i>forward</i> y hacia atrás <i>backward</i> [Evaluar]</li> <li>• Explicar el concepto y las aplicaciones de mapeo de texturas, muestreo y el <i>anti-aliasing</i> [Familiarizarse]</li> <li>• Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse]</li> <li>• Implementar un sencillo renderizador en tiempo real utilizando una API de rasterización (por ejemplo, OpenGL) utilizando buffers de vértices y <i>shaders</i> [Usar]</li> <li>• Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar]</li> <li>• Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]</li> </ul>

Lecturas : [HB90], [Hug+13], [Wol11], [Shr+13]

**Unidad 3: Programación de Sistemas Interactivos (2 horas)****Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Manejo de eventos e interacción de usuario.</li><li>• Enfoques para el diseño, implementación y evaluación de la interacción sin mouse<ul style="list-style-type: none"><li>– Interfaces táctiles y multitáctiles.</li><li>– Interfaces compartidas, incorporadas y grandes</li><li>– Nuevas modalidades de entrada (tales como datos de sensores y localización)</li><li>– Nuevas ventanas, por ejemplo, iPhone, Android</li><li>– Reconocimiento de voz y procesamiento del lenguaje natural</li><li>– Interfaces utilizables y tangibles</li><li>– Interacción persuasiva y emoción</li><li>– Tecnologías de interacción ubicuas y contextuales (Ubicomp)</li><li>– Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada)</li><li>– Visualización e interacción de ambiente / periféricos</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar]</li></ul>

**Lecturas :** [HB90]

<b>Unidad 4: Modelado Geométrico (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Operaciones geométricas básicas como cálculo de intersección y pruebas de proximidad.</li> <li>• Volúmenes, voxels y representaciones basadas en puntos.</li> <li>• Curvas polinomiales y Superficies paramétricas.</li> <li>• Representación implícita de curvas y superficies.</li> <li>• Técnicas de aproximación, tales como curvas polinómicas, curvas Bezier, curvas spline y superficies, y base racional no uniforme (NURB) espinas, y el método de ajuste de nivel.</li> <li>• Técnicas de superficie de representación incluyendo teselación, la representación de malla, carenado malla, y las técnicas de generación de mallas, como la triangulación de Delaunay, marchando cubos.</li> <li>• Técnicas de subdivisión espacial.</li> <li>• Modelos procedimentales como fractales, modelamiento generativo y sistemas L.</li> <li>• Modelos deformables de forma libre y elásticamente deformables.</li> <li>• Subdivisión de superficies.</li> <li>• Modelado multiresolución.</li> <li>• Reconstrucción.</li> <li>• Representación de Geometría Sólida Constructiva (GSC)</li> </ul>	<ul style="list-style-type: none"> <li>• Representar curvas y superficies utilizando formas tanto implícitas y paramétricas [Usar]</li> <li>• Crear modelos poliédrico simples por teselación de superficies [Usar]</li> <li>• Generar un modelo fractal o terreno usando un método de procedimiento [Usar]</li> <li>• Generar una malla de un conjunto de puntos adquiridos por un scanner laser [Usar]</li> <li>• Construct modelos de geometría sólida constructiva a partir de simples primitivas, tales como cubos y superficies cuádricas [Usar]</li> <li>• Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar]</li> </ul>

**Lecturas :** [HB90], [Shr+13]

<b>Unidad 5: Renderizado Avanzado (6 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tiempo (desenfoque de movimiento), la posición del objetivo (enfoque), y la frecuencia continua (color) y su impacto en la representación.</li> <li>• Mapeo de Sombras.</li> <li>• Selectiva de oclusión.</li> <li>• Dispersión de la Superficie.</li> <li>• Renderizado no fotorealístico.</li> <li>• Arquitectura del GPU.</li> <li>• Sistemas visuales humanos incluida la adaptación a la luz, la sensibilidad al ruido, y la fusión de parpadeo.</li> </ul>	<ul style="list-style-type: none"> <li>• Demostrar como un algoritmo calcula una solución a la ecuación de renderización [Evaluar]</li> <li>• Demostrar las propiedades de un algoritmo de renderización, por ejemplo, completo, consistente, e imparcial [Evaluar]</li> <li>• Implementar un algoritmo no trivial de sombreado (por ejemplo, sombreado caricaturizado (<i>toon shading</i>), mapas de sombras en cascada (<i>cascaded shadow maps</i>)) bajo una API de rasterización [Usar]</li> <li>• Discutir como una técnica artística particular puede ser implementada en un renderizador [Familiarizarse]</li> <li>• Explicar como reconocer las técnicas gráficas usadas para crear una imagen en particular [Familiarizarse]</li> </ul>

**Lecturas :** [HB90], [Hug+13], [Wol11], [Shr+13]

<b>Unidad 6: Animación por computadora (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Cinemática directa e inversa.</li> <li>• Detección de colisiones y respuesta.</li> <li>• Animación procedimental empleando ruido, reglas (boids/crowds) y sistemas de partículas.</li> <li>• Algoritmos Skinning.</li> <li>• Movimientos basado en la física, incluyendo la dinámica del cuerpo rígido, sistemas de partículas físicas, redes de masa-muelle de tela y la carne y el pelo.</li> <li>• Animación de Cuadros Principales</li> <li>• Splines</li> <li>• Estructuras de datos para rotaciones, como cuaterniones.</li> <li>• Animación de Cámara.</li> <li>• Captura de Movimiento.</li> </ul>	<ul style="list-style-type: none"> <li>• Calcular la orientación de partes articuladas de un modelo de una localización y orientación usando un enfoque de cinemática inversa [Usar]</li> <li>• Implementar el método de interpolación <i>spline</i> para producir las posiciones y orientaciones en medio [Usar]</li> <li>• Implementar algoritmos para el modelamiento físico de partículas dinámicas usando simplemente la mecánica de Newton, por ejemplo Witkin &amp; Kass, serpientes y gusanos, Euler simpléctica, Stormer/Verlet, o métodos de punto medio de Euler [Usar]</li> <li>• Discutir las ideas básicas detrás de algunos métodos para dinámica de fluidos para el modelamiento de trayectorias balísticas, por ejemplo salpicaduras, polvo, fuego, o humo [Familiarizarse]</li> <li>• Usar el software de animación común para construir formas orgánicas simples usando <i>metaball</i> y el esqueleto [Usar]</li> </ul>

**Lecturas :** [HB90], [Shr+13]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### 10. BIBLIOGRAFÍA BÁSICA

- [HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C*. Prentice Hall, 1990.
- [Wol11] David Wolff. *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing, 2011.
- [Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition*. Addison-Wesley, 2013.
- [Shr+13] Dave Shreiner et al. *OpenGL, Programming Guide, Eighth Edition*. Addison-Wesley, 2013.

## 1. CURSO

CS292-CS2023. Ingeniería de Software II (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS292-CS2023. Ingeniería de Software II
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS291. Ingeniería de Software I. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El desarrollo moderno de software requiere dominio de DevSecOps, infraestructura como código y prácticas SRE. Este curso prepara estudiantes para diseñar, construir y operar sistemas confiables y seguros en entornos cloud-native, integrando automatización, observabilidad y seguridad en todo el ciclo de vida del software.

## 5. OBJETIVOS

- Dominar pipelines DevSecOps completos integrando construcción, pruebas, seguridad y despliegues automatizados con herramientas modernas.
- Implementar infraestructura como código segura usando Terraform con políticas como código y prácticas de supply chain security.
- Diseñar y desplegar aplicaciones contenedorizadas en Docker y Kubernetes con observabilidad y prácticas SRE para alta confiabilidad.
- Aplicar metodologías ágiles con Git profesional, CI/CD integrado y métricas de equipo para desarrollo colaborativo efectivo.
- Integrar estándares NIST SSDF y SLSA en el ciclo de vida del software asegurando trazabilidad y cumplimiento desde desarrollo hasta producción.

## 6. RESULTADOS DEL ESTUDIANTE

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Métodos de Desarrollo (8 horas)</b>	
<b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Comprensión de programas.</li><li>• Exactitud del programa.<ul style="list-style-type: none"><li>– Tipos de error (sintaxis, logica, tiempo de ejecucion)</li><li>– El concepto de la especificacion</li><li>– Programación defensiva (ejem., codificación segura, manejo de excepciones)</li><li>– Revision de codigo</li><li>– Fundamentos de <i>testing</i> y generación de casos de prueba</li><li>– El rol y el uso de contratos, incluyendo pre- y post- condiciones</li><li>– Pruebas unitarias</li></ul></li><li>• Refactorización simple.</li><li>• Entornos modernos de programación:<ul style="list-style-type: none"><li>– Búsqueda de código.</li><li>– Programación usando librería de componentes y sus APIs.</li></ul></li><li>• Estrategias de depuración.</li><li>• Documentación y estilo de programas.</li></ul>	
<b>Lecturas :</b> [Kim+16], [Mor20], [FHK21]	

<b>Unidad 2: Conceptos Fundamentales de Programación (10 horas)</b>	
<b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Sintaxis y semántica básica de un lenguaje de alto nivel.</li><li>• Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos)</li><li>• Expresiones y asignaciones.</li><li>• Operaciones básicas I/O incluyendo archivos I/O.</li><li>• Estructuras de control condicional e iterativas.</li><li>• Paso de funciones y parámetros.</li><li>• Concepto de recursividad.</li></ul>	
<b>Lecturas :</b> [Bur18], [MK18], [LF20]	

<b>Unidad 3: Métodos de Desarrollo (8 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Comprensión de programas.</li> <li>• Exactitud del programa. <ul style="list-style-type: none"> <li>– Tipos de error (sintaxis, logica, tiempo de ejecucion)</li> <li>– El concepto de la especificacion</li> <li>– Programación defensiva (ejem., codificación segura, manejo de excepciones)</li> <li>– Revision de codigo</li> <li>– Fundamentos de <i>testing</i> y generación de casos de prueba</li> <li>– El rol y el uso de contratos, incluyendo pre- y post- condiciones</li> <li>– Pruebas unitarias</li> </ul> </li> <li>• Refactorización simple.</li> <li>• Entornos modernos de programación: <ul style="list-style-type: none"> <li>– Búsqueda de código.</li> <li>– Programación usando librería de componentes y sus APIs.</li> </ul> </li> <li>• Estrategias de depuración.</li> <li>• Documentación y estilo de programas.</li> </ul>	

**Lecturas :** [Bey+16], [Bey+18], [AR12]

<b>Unidad 4: Introducción a redes (12 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc)</li> <li>• Técnicas de Switching (por ejemplo, de circuitos, de paquetes)</li> <li>• Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls.</li> <li>• Principios de capas (encapsulación, multiplexación)</li> <li>• Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)</li> </ul>	

**Lecturas :** [Bur18], [BF16], [HBB17]

<b>Unidad 5: Principios de Sistemas Operativos (8 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Métodos de estructuración (monolítico, capas, modular, los modelos micro-kernel)</li> <li>• Abstracciones, procesos y recursos.</li> <li>• Los conceptos de interfaces de programa de aplicación (API)</li> <li>• La evolución de las técnicas de hardware / software y las necesidades de aplicación</li> <li>• Organización de dispositivos.</li> <li>• Interrupciones: métodos e implementaciones.</li> <li>• Concepto de usuario de estado / sistema y la protección, la transición al modo kernel.</li> </ul>	
<b>Lecturas :</b> [Lov10], [Ker10], [TB15]	

<b>Unidad 6: Introducción (10 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial)</li> <li>• Programación a través de APIs específicos.</li> <li>• Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5)</li> <li>• Programación bajo restricciones de plataforma.</li> </ul>	
<b>Lecturas :</b> [Kle17], [SF13], [Fow12]	

<b>Unidad 7: Principios de Diseño Seguro (8 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Menor privilegio y aislamiento.</li> <li>• Valores predeterminados a prueba de fallos.</li> <li>• Diseño abierto.</li> <li>• La seguridad de extremo a extremo.</li> <li>• La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas)</li> <li>• Diseño de seguridad.</li> <li>• Las tensiones entre la seguridad y otros objetivos de diseño.</li> <li>• Mediación completa.</li> <li>• El uso de componentes de seguridad vetados.</li> <li>• Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque)</li> <li>• Seguridad utilizable.</li> <li>• Componibilidad de seguridad.</li> <li>• Prevención, detección y disuasión.</li> </ul>	
<b>Lecturas :</b> [Smi21], [Sho14], [HL06]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [HL06] Michael Howard and David LeBlanc. *Writing Secure Code*. 2nd. Microsoft Press, 2006.
- [Ker10] Michael Kerrisk. *The Linux Programming Interface*. No Starch Press, 2010.
- [Lov10] Robert Love. *Linux System Programming: Talking Directly to the Kernel and C Library*. 2nd. O'Reilly Media, 2010.
- [AR12] John Allspaw and Jesse Robbins. *Web Operations: Keeping the Data On Time*. O'Reilly Media, 2012.
- [Fow12] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2012.
- [SF13] Pramod Sadalage and Martin Fowler. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2013.
- [Sho14] Adam Shostack. *Threat Modeling: Designing for Security*. Wiley, 2014.

- 
- [TB15] Andrew Tanenbaum and Herbert Bos. *Modern Operating Systems*. 4th. Pearson, 2015.
- [Bey+16] Betsy Beyer et al. *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016.
- [BF16] Matt Butcher and Matt Farina. *Docker in Practice*. 2nd. Manning Publications, 2016.
- [Kim+16] Gene Kim et al. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.
- [HBB17] Kelsey Hightower, Brendan Burns, and Joe Beda. *Kubernetes: Up and Running*. O'Reilly Media, 2017.
- [Kle17] Martin Kleppmann. *Designing Data-Intensive Applications*. O'Reilly Media, 2017.
- [Bey+18] Betsy Beyer et al. *The Site Reliability Workbook: Practical Ways to Implement SRE*. O'Reilly Media, 2018.
- [Bur18] Brendan Burns. *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services*. O'Reilly Media, 2018.
- [MK18] Karl Matthias and Sean Kane. *Docker: Up and Running*. 2nd. O'Reilly Media, 2018.
- [LF20] James Lewis and Martin Fowler. *Microservices: A Definition of This New Architectural Term*. Available online. ThoughtWorks, 2020.
- [Mor20] Kief Morris. *Infrastructure as Code: Dynamic Systems for the Cloud Age*. 2nd. O'Reilly Media, 2020.
- [FHK21] Nicole Forsgren, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press, 2021.
- [Smi21] Richard Smith. *DevSecOps: A Leader's Guide to Producing Secure Software without Compromising Speed, Agility, and Innovation*. Apress, 2021.

## 1. CURSO

CS2H1-CS2023. Experiencia de Usuario (UX) (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS2H1-CS2023. Experiencia de Usuario (UX)
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS291. Ingeniería de Software I. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El lenguaje ha sido una de las creaciones más significativas de la humanidad. Desde el lenguaje corporal y gestual, pasando por la comunicación verbal y escrita, hasta códigos simbólicos icónicos y otros, ha posibilitado interacciones complejas entre los seres humanos y facilitado considerablemente la comunicación de información. Con la invención de dispositivos automáticos y semiautomáticos, entre los que se cuentan las computadoras, la necesidad de lenguajes o interfaces para poder interactuar con ellos, ha cobrado gran importancia.

La usabilidad del software, aunada a la satisfacción del usuario y su incremento de productividad, depende de la eficacia de la Interfaz Usuario-Computador. Tanto es así, que a menudo la interfaz es el factor más importante en el éxito o el fracaso de cualquier sistema computacional. El diseño e implementación de adecuadas Interfaces Humano-Computador, que además de cumplir los requisitos técnicos y la lógica transaccional de la aplicación, considere las sutiles implicaciones psicológicas, culturales y estéticas de los usuarios, consume buena parte del ciclo de vida de un proyecto software, y requiere habilidades especializadas, tanto para la construcción de las mismas, como para la realización de pruebas de usabilidad.

## 5. OBJETIVOS

- Conocer y aplicar criterios de usabilidad y accesibilidad al diseño y construcción de interfaces humano-computador, buscando siempre que la tecnología se adapte a las personas y no las personas a la tecnología.
- Que el alumno tenga una visión centrada en la experiencia de usuario al aplicar apropiados enfoques conceptuales y tecnológicos.
- Entender como la tecnologica emergente hace posible nuevos estilos de interacción.
- Determinar los requerimientos básicos a nivel de interfaces, hardware y software para la construcción de ambientes inmersivos.

## 6. RESULTADOS DEL ESTUDIANTE

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

- 4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos (8 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Contextos para IHC (cualquiera relacionado con una interfaz de usuario, p.e., página web, aplicaciones de negocios, aplicaciones móviles y juegos)</li> <li>• Heurística de usabilidad y los principios de pruebas de usabilidad.</li> <li>• Procesos para desarrollo centrado en usuarios, p.e., enfoque inicial en usuarios, pruebas empíricas, diseño iterativo.</li> <li>• Principios del buen diseño y buenos diseñadores; ventajas y desventajas de ingeniería.</li> <li>• Diferentes medidas para evaluación, p.e., utilidad, eficiencia, facilidad de aprendizaje, satisfacción de usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir por qué el desarrollo de software centrado en el hombre es importante [Familiarizarse]</li> <li>• Define un proceso de diseño centralizado en el usuario que de forma explícita considere el hecho que un usuario no es como un desarrollador o como sus conocimientos [Familiarizarse]</li> <li>• Resumir los preceptos básicos de la interacción psicológica y social [Familiarizarse]</li> <li>• Desarrollar y usar un vocabulario conceptual para analizar la interacción humana con el software: disponibilidad, modelo conceptual, retroalimentación, y demás [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Dix+04], [Sto+05], [RS11]	

<b>Unidad 2: Factores Humanos (8 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Modelos cognoscitivos que informan diseño de interacciones, p.e., atención, percepción y reconocimiento, movimiento, memoria, golfos de expectativa y ejecución.</li> <li>• Capacidades físicas que informan diseño de interacción, p.e. percepción del color, ergonomía.</li> <li>• Accesibilidad, p.e., interfaces para poblaciones con diferentes habilidades (p.e., invidentes, discapacitados)</li> <li>• Interfaces para grupos de población de diferentes edades (p.e., niños, mayores de 80)</li> </ul>	<ul style="list-style-type: none"> <li>• Crear y dirigir una simple prueba de usabilidad para una aplicación existente de software [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

### Unidad 3: Diseño y Testing centrados en el usuario (16 horas)

#### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Enfoque y características del proceso de diseño.</li> <li>• Requerimientos de funcionalidad y usabilidad.</li> <li>• Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual.</li> <li>• Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas.</li> <li>• Análisis de tareas, incluidos los aspectos cualitativos de la generación de modelos de análisis de tareas.</li> <li>• Consideración de IHC como una disciplina de diseño: <ul style="list-style-type: none"> <li>– Sketching</li> <li>– Diseño participativo</li> <li>– Sketching</li> <li>– Diseño participativo</li> </ul> </li> <li>• Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página.</li> <li>• Prototipos de baja fidelidad (papel)</li> <li>• Técnicas de evaluación cuantitativa ej. evaluación Keystroke-level.</li> <li>• Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar.</li> <li>• Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación.</li> <li>• Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización.</li> <li>• Reportar los resultados de las evaluaciones.</li> <li>• Internacionalización, diseño para usuarios de otras culturas, intercultural.</li> </ul>	<ul style="list-style-type: none"> <li>• Llevar a cabo una evaluación cuantitativa y discutir / informar sobre los resultados [Familiarizarse]</li> <li>• Para un grupo de usuarios determinado, realizar y documentar un análisis de sus necesidades [Familiarizarse]</li> <li>• Discutir al menos un standard nacional o internacional de diseño de interfaz de usuario [Familiarizarse]</li> <li>• Explicar cómo el diseño centrado en el usuario complementa a otros modelos de proceso software [Familiarizarse]</li> <li>• Utilizar <i>lo-fi</i> (baja fidelidad) técnicas de prototipado para recopilar y reportar, las respuestas del usuario [Usar]</li> <li>• Elegir los métodos adecuados para apoyar el desarrollo de una específica interfaz de usuario [Evaluar]</li> <li>• Utilizar una variedad de técnicas para evaluar una interfaz de usuario dada [Evaluar]</li> <li>• Comparar las limitaciones y beneficios de los diferentes métodos de evaluación [Evaluar]</li> </ul>

Lecturas : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]

<b>Unidad 4: Diseño de Interacción (8 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Principios de interfaces gráficas de usuario (GUIs)</li> <li>• Elementos de diseño visual (disposición, color, fuentes, etiquetado)</li> <li>• Manejo de fallas humanas/sistema.</li> <li>• Estándares de interfaz de usuario.</li> <li>• Presentación de información: navegación, representación, manipulación.</li> <li>• Técnicas de animación de interfaz (ej. grafo de escena)</li> <li>• Clases Widget y bibliotecas.</li> <li>• Internacionalización, diseño para usuarios de otras culturas, intercultural.</li> <li>• Elección de estilos de interacción y técnicas de interacción.</li> </ul>	<ul style="list-style-type: none"> <li>• Crear una aplicación simple, junto con la ayuda y la documentación, que soporta una interfaz gráfica de usuario [Usar]</li> </ul>
<b>Lecturas :</b> [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

## Unidad 5: Nuevas Tecnologías Interactivas (8 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Elección de estilos de interacción y técnicas de interacción.</li> <li>• Enfoques para el diseño, implementación y evaluación de la interacción sin mouse <ul style="list-style-type: none"> <li>– Interfaces táctiles y multitáctiles.</li> <li>– Interfaces compartidas, incorporadas y grandes</li> <li>– Nuevas modalidades de entrada (tales como datos de sensores y localización)</li> <li>– Nuevas ventanas, por ejemplo, iPhone, Android</li> <li>– Reconocimiento de voz y procesamiento del lenguaje natural</li> <li>– Interfaces utilizables y tangibles</li> <li>– Interacción persuasiva y emoción</li> <li>– Tecnologías de interacción ubicuas y contextuales (Ubicomp)</li> <li>– Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada)</li> <li>– Visualización e interacción de ambiente / periféricos</li> </ul> </li> <li>• Salida: <ul style="list-style-type: none"> <li>– Sonido</li> <li>– Visualización estereoscópica</li> <li>– Forzar la simulación de retroalimentación, dispositivos hápticos</li> </ul> </li> <li>• Arquitectura de Sistemas: <ul style="list-style-type: none"> <li>– Motores de Juego</li> <li>– Relidad Aumentada móvil</li> <li>– Simuladores de vuelo</li> <li>– CAVEs</li> <li>– Imágenes médicas</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Describe cuando son adecuadas las interfaces sin uso de ratón [Familiarizarse]</li> <li>• Comprende las posibilidades de interacción que van más allá de las interfaces de ratón y puntero [Familiarizarse]</li> <li>• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Usar]</li> <li>• Describir el modelo óptico realizado por un sistema de gráficos por computadora para sintetizar una visión estereoscópica [Familiarizarse]</li> <li>• Describir los principios de las diferentes tecnologías de seguimiento de espectador [Familiarizarse]</li> <li>• Determinar los requerimientos básicos en interfaz, software, hardware, y configuraciones de software de un sistema VR para una aplicación específica [Evaluar]</li> </ul>

Lecturas : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]

<b>Unidad 6: Colaboración y Comunicación (8 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• La comunicación asíncrona en grupo, por ejemplo, el correo electrónico, foros, redes sociales.</li> <li>• Medios de comunicación social, informática social, y el análisis de redes sociales.</li> <li>• Colaboración en línea, espacios "inteligentes" y aspectos de coordinación social de tecnologías de flujo de trabajo.</li> <li>• Comunidades en línea.</li> <li>• Personajes de Software y agentes inteligentes, mundos virtuales y avatares.</li> <li>• Psicología Social</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la diferencia entre la comunicación sincrónica y asíncrona [Familiarizarse]</li> <li>• Comparar los problemas de IHC en la interacción individual con la interacción del grupo [Familiarizarse]</li> <li>• Discutir varias problemas de interés social planteados por el software colaborativo [Usar]</li> <li>• Discutir los problemas de IHC en software que personifica la intención humana [Evaluar]</li> </ul>

**Lecturas :** [Dix+04], [Sto+05], [RS11]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.
- [Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.
- [Sto+05] D. Stone et al. *User Interface Design and Evaluation*. Morgan Kaufmann Series in Interactive Technologies, 2005.
- [LS06] M. Leavitt and B. Schneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.
- [Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.
- [Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.
- [Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.
- [RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction*. 3 ed. John Wiley and Sons Ltd, 2011.
- [WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc, 2011.

## 1. CURSO

AI264. Aprendizaje Profundo (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	AI264. Aprendizaje Profundo
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI263. Introducción al Aprendizaje de Máquina. (6 <sup>to</sup> 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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos de DL (15 horas)</b> <b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Backpropagation moderno</li> <li>• Regularización (Dropout, BatchNorm)</li> <li>• Optimizadores avanzados</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar redes DNN desde cero [Evaluar]</li> <li>• Diagnosticar problemas de entrenamiento [Usar]</li> </ul>

**Lecturas :** [GBC16]

<b>Unidad 2: Arquitecturas Avanzadas (25 horas)</b> <b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• ResNets y CNN profundas</li> <li>• Transformers y atención</li> <li>• Modelos generativos (VAEs, GANs)</li> </ul>	<ul style="list-style-type: none"> <li>• Adaptar arquitecturas a dominios específicos [Evaluar]</li> <li>• Entrenar transformers en datasets reales [Usar]</li> </ul>

**Lecturas :** [Vas+17], [He+16]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [He+16] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CVPR* (2016), pp. 770–778.
- [Vas+17] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems* 30 (2017).

## 1. CURSO

BI101FCCS. Biología (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	BI101FCCS. Biología
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La biología es la ciencia de la vida y proporciona una base fundamental para campos interdisciplinarios como la bioinformática, la biología computacional y la inteligencia artificial inspirada en la biología. Este curso introduce los conceptos fundamentales de la biología, desde la célula hasta la evolución, con un enfoque especial en aplicaciones computacionales.

## 5. OBJETIVOS

- Comprender los principios básicos de la biología celular y molecular.
- Conocer los procesos fundamentales de la vida, como la replicación del ADN, la transcripción y la traducción.
- Apreciar la interrelación entre la biología y la computación en áreas como la bioinformática.
- Desarrollar habilidades para analizar datos biológicos de manera cualitativa y cuantitativa.
- Fomentar el pensamiento interdisciplinario entre biología y ciencias computacionales.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)
- 6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Usage)

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

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Usage)

## 7. TEMAS

<b>Unidad 1: Introducción a la Biología y la Vida (6 horas)</b>	
<b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Características de los seres vivos y definición de vida.</li> <li>• Niveles de organización biológica (de moléculas a ecosistemas).</li> <li>• El método científico en biología.</li> <li>• Modelos computacionales de sistemas vivos.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las características de los seres vivos. [Familiarizarse]</li> <li>• Identificar los diferentes niveles de organización biológica. [Usar]</li> <li>• Aplicar el método científico en el contexto de la biología. [Evaluar]</li> <li>• Explicar modelos computacionales de ecosistemas. [Usar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 2: Química de la Vida (6 horas)</b>	
<b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Átomos, moléculas y enlaces químicos.</li> <li>• Biomoléculas: proteínas, carbohidratos, lípidos, ácidos nucleicos.</li> <li>• Relación con estructuras de datos en computación.</li> <li>• Cadenas de ADN como secuencias computacionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar las principales biomoléculas y sus funciones. [Familiarizarse]</li> <li>• Explicar la relación entre estructuras biológicas y de datos. [Usar]</li> <li>• Analizar secuencias biológicas desde una perspectiva computacional. [Evaluar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 3: Estructura y Función Celular (8 horas)</b>	
<b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Tipos de células: procariotas y eucariotas.</li> <li>• Orgánulos celulares y sus funciones.</li> <li>• Membranas celulares y transporte.</li> <li>• Redes biológicas y modelado computacional.</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar y contrastar células procariotas y eucariotas. [Familiarizarse]</li> <li>• Describir la función de los principales orgánulos celulares. [Usar]</li> <li>• Explicar los mecanismos de transporte a través de membranas. [Evaluar]</li> <li>• Introducir conceptos de redes biológicas y su modelado. [Usar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 4: Metabolismo y Energía (6 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fotosíntesis y respiración celular.</li> <li>• Enzimas y catálisis biológica.</li> <li>• Analogías con algoritmos de optimización.</li> <li>• Eficiencia energética en sistemas biológicos y computacionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los procesos de fotosíntesis y respiración celular. [Familiarizarse]</li> <li>• Describir el papel de las enzimas en el metabolismo. [Usar]</li> <li>• Establecer analogías entre eficiencia energética biológica y computacional. [Evaluar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 5: Genética Básica (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• ADN: estructura y replicación.</li> <li>• ARN: transcripción y traducción.</li> <li>• Conceptos de herencia mendeliana.</li> <li>• Aplicación en algoritmos genéticos.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la estructura y replicación del ADN. [Familiarizarse]</li> <li>• Explicar los procesos de transcripción y traducción. [Usar]</li> <li>• Comprender los principios de la herencia mendeliana. [Evaluar]</li> <li>• Aplicar conceptos genéticos en algoritmos computacionales. [Usar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 6: Evolución y Diversidad (6 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Teoría de la evolución por selección natural.</li> <li>• Evidencias de la evolución.</li> <li>• Biodiversidad y clasificación biológica.</li> <li>• Simulación evolutiva en computación.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar la teoría de la evolución por selección natural. [Familiarizarse]</li> <li>• Describir las evidencias de la evolución. [Usar]</li> <li>• Analizar los mecanismos de la evolución. [Evaluar]</li> <li>• Aplicar conceptos evolutivos en simulaciones computacionales. [Usar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

<b>Unidad 7: Ecología Básica y Aplicaciones (8 horas)</b> <b>Resultados esperados: 1,AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Interacciones entre organismos.</li> <li>• Ciclos biogeoquímicos.</li> <li>• Dinámicas poblacionales.</li> <li>• Modelado computacional de sistemas ecológicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las interacciones ecológicas básicas. [Familiarizarse]</li> <li>• Explicar los ciclos biogeoquímicos. [Usar]</li> <li>• Analizar dinámicas poblacionales. [Evaluar]</li> <li>• Aplicar modelado computacional a problemas ecológicos. [Usar]</li> </ul>

**Lecturas :** [Urr+17], [Rav+17]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[Rav+17] Peter H. Raven et al. *Biology*. McGraw-Hill Education, 2017.

[Urr+17] Lisa A. Urry et al. *Campbell Biology*. Pearson, 2017.

## 1. CURSO

FG211-ACM. Ética Profesional (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FG211-ACM. Ética Profesional
<b>2.2 Semestre</b>	:	7 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	1 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	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

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos Éticos en IA (10 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Comparación: Códigos de ética ACM vs IEEE para IA.</li><li>• Principios de transparencia y justicia algorítmica.</li></ul>	<ul style="list-style-type: none"><li>• Explicar diferencias entre marcos éticos [Familiarizarse].</li><li>• Evaluar compliance en casos reales [Evaluar].</li></ul>
<b>Lecturas :</b> [ACM18], [IEE20]	

<b>Unidad 2: Privacidad y Regulaciones (14 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>GDPR aplicado a ML: derecho a explicación.</li> <li>Auditoría ética de datasets (ej: ImageNet).</li> </ul>	<ul style="list-style-type: none"> <li>Implementar checklist GDPR para modelos [Usar].</li> <li>Identificar violaciones en estudios de caso [Evaluar].</li> </ul>

**Lecturas :** [VB18], [Mül21]

<b>Unidad 3: Sesgos y Equidad (18 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Métricas de equidad (paridad demográfica, igualdad de oportunidades).</li> <li>Herramientas: IBM Fairness 360, Google What-If.</li> </ul>	<ul style="list-style-type: none"> <li>Medir sesgos en modelos con Python [Usar].</li> <li>Proponer estrategias de mitigación [Evaluar].</li> </ul>

**Lecturas :** [For20], [Meh+21]

<b>Unidad 4: Responsabilidad en Sistemas Autónomos (18 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Atribución legal en fallos de IA (ej: vehículos autónomos).</li> <li>Normas ISO para IA confiable.</li> </ul>	<ul style="list-style-type: none"> <li>Redactar cláusulas de responsabilidad [Evaluar].</li> <li>Analizar conflictos legales [Familiarizarse].</li> </ul>

**Lecturas :** [Bos14], [Com19]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Bos14] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.
- [ACM18] ACM. *ACM Code of Ethics and Professional Conduct*. Tech. rep. 2018. URL: <https://www.acm.org/code-of-ethics>.
- [VB18] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 2018.
- [Com19] IEEE Ethics Committee. *Ethical Dilemmas in Engineering*. Tech. rep. 2019. URL: <https://www.ieee.org/ethics>.
- [For20] ACM Ethics Task Force. *Case Studies in Computing and Society*. Tech. rep. 2020. URL: <https://ethics.acm.org/case-studies/>.
- [IEE20] IEEE. *IEEE Code of Ethics*. Tech. rep. 2020. URL: <https://www.ieee.org/about/corporate/governance/p7-8.html>.

- 
- [Meh+21] Ninareh Mehrabi et al. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Computing Surveys* 54.6 (2021). doi: 10.1145/3457607.
- [Mül21] Vincent C. Müller. *Ethics of Artificial Intelligence and Robotics*. Cambridge University Press, 2021.



## **Capítulo 8**

### **Octavo Semestre**

## 1. CURSO

CS281-CS2023. Computación en la Sociedad (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS281-CS2023. Computación en la Sociedad
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	2 HT;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Ofrece una visión amplia de los aspectos éticos y profesionales relacionados con la computación. Los tópicos que se incluyen abarcan los aspectos éticos, sociales y políticos. Las dimensiones morales de la computación. Los métodos y herramientas de análisis. Administración de los recursos computacionales. Seguridad y control de los sistemas computacionales. Responsabilidades profesionales y éticas. Propiedad intelectual.

## 5. OBJETIVOS

- Hacer que el alumno entienda la importancia del cuidado y la ética en la transferencia y uso de la información.
- Inculcar en el alumno que las tendencias de mejoramiento de la tecnología, no debe ser llevada a degradar la moral de la sociedad.

## 6. RESULTADOS DEL ESTUDIANTE

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

3) Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

**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)

## 7. TEMAS

<b>Unidad 1: Historia (2 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Pre-historia – El mundo antes de 1946.</li> <li>• Historia del hardware, software, redes.</li> <li>• Pioneros de la Computación.</li> <li>• Historia de Internet.</li> </ul>	<ul style="list-style-type: none"> <li>• Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse]</li> <li>• Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse]</li> <li>• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]</li> <li>• Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [LL04], [McL00]	

Unidad 2: Contexto Social (4 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Implicancias sociales de la computación en un mundo conectado en red.</li> <li>• Impacto de los medios sociales en el individualismo, colectivismo y en la cultura.</li> <li>• Crecimiento y control de la Internet</li> <li>• A menudo se refiere como la brecha digital, las diferencias en el acceso a los recursos de la tecnología digital y sus ramificaciones resultantes para el género, la clase, la etnia, la geografía, y/o los países subdesarrollados.</li> <li>• Los problemas de accesibilidad, incluyendo los requisitos legales.</li> <li>• Computación consciente del contexto.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir las formas positivas y negativas en las que la tecnología computacional (redes, computación móvil, <i>cloud computing</i>) altera los modos de interacción social en el plano personal [Familiarizarse]</li> <li>• Identificar los supuestos y valores incorporados en el hardware y el software de diseño de los desarrolladores, especialmente lo que se refiere a la facilidad de uso para diversas poblaciones incluyendo minorías poblaciones y los discapacitados [Usar]</li> <li>• Interpretar el contexto social de un determinado diseño y su aplicación [Evaluar]</li> <li>• Evaluar la eficacia de un diseño y aplicación dada a partir de datos empíricos [Familiarizarse]</li> <li>• Resumir las implicaciones de los medios sociales en el individualismo frente al colectivismo y la cultura [Familiarizarse]</li> <li>• Discutir cómo el acceso a Internet sirve como una fuerza liberadora para las personas que viven bajo las formas opresivas de gobierno; explicar la utilización los límites al acceso a Internet como herramientas de represión política y social [Familiarizarse]</li> <li>• Analizar los pros y los contras de la dependencia de la computación en la implementación de la democracia (por ejemplo, prestación de servicios sociales, votación electrónica) [Familiarizarse]</li> <li>• Describir el impacto de la escasa representación de las diversas poblaciones en la profesión (por ejemplo, la cultura de la industria, la diversidad de productos) [Usar]</li> <li>• Explicar las consecuencias de la sensibilidad al contexto en los sistemas de computación ubicua [Familiarizarse]</li> </ul>

Lecturas : [LL04], [McL00]

<b>Unidad 3: Herramientas de Análisis (2 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Argumentación ética.</li> <li>• Teorías éticas y toma de decisiones.</li> <li>• Suposiciones morales y valores.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluar las posiciones de las partes interesadas en una situación dada [Familiarizarse]</li> <li>• Analizar errores lógicos básicos en una discusión [Usar]</li> <li>• Analizar un argumento para identificar premisas y la conclusión [Familiarizarse]</li> <li>• Ilustrar el uso de ejemplo y analogía en el argumento ético [Familiarizarse]</li> <li>• Evaluar compensaciones éticos / sociales en las decisiones técnicas [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [LL04], [McL00]	

#### Unidad 4: Ética Profesional (4 horas)

##### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Community values and the laws by which we live.</li> <li>La naturaleza del profesionalismo incluido el cuidado, la atención y la disciplina, la responsabilidad fiduciaria y mentoría.</li> <li>Mantenerse al día como profesional de computación en términos de familiaridad, herramientas, habilidades, marco legal y profesional, así como la capacidad de autoevaluarse y avances en el campo de la computación.</li> <li>La certificación profesional, códigos de ética, conducta y práctica, como la ACM / IEEE-CS, SE, AITP, IFIP y las sociedades internacionales.</li> <li>Rendición de cuentas, la responsabilidad y la confiabilidad (por ejemplo, la corrección de software, fiabilidad y seguridad, así como la confidencialidad ética de los profesionales de seguridad cibernética)</li> <li>El papel del profesional de computación en las políticas públicas.</li> <li>Mantenimiento de la conciencia en relación a las consecuencias.</li> <li>Disidencia ética y la denuncia de irregularidades.</li> <li>La relación entre la cultura regional y dilemas éticos.</li> <li>Tratar con el acoso y la discriminación.</li> <li>Formas de credenciamiento profesional.</li> <li>Políticas de uso aceptable para la computación en el lugar de trabajo.</li> <li>Ergonomía y entornos de trabajo computacionales saludables.</li> <li>Consideraciones a tiempos de entrega de mercado vs estándares de calidad profesional.</li> </ul>	<ul style="list-style-type: none"> <li>Identificar los problemas éticos que se plantean en el desarrollo de software y determinar cómo abordarlos técnica y éticamente [Usar]</li> <li>Explicar la responsabilidad ética de velar por la corrección de software, confiabilidad y seguridad [Evaluar]</li> <li>Describir los mecanismos que normalmente existen para que profesional se mantenga al día [Familiarizarse]</li> <li>Describir las fortalezas y debilidades de códigos profesionales relevantes como expresiones de profesionalismo y guías para la toma de decisiones [Familiarizarse]</li> <li>Analizar un problema mundial de computación, observando el papel de los profesionales y funcionarios del gobierno en el manejo de este problema [Familiarizarse]</li> <li>Evaluar los códigos de ética profesional de la ACM, la Sociedad de Computación de la IEEE, y otras organizaciones [Familiarizarse]</li> <li>Describir las formas en que los profesionales pueden contribuir a las políticas públicas [Familiarizarse]</li> <li>Describir las consecuencias de la conducta profesional inadecuada [Usar]</li> <li>Identificar las etapas progresivas en un incidente de denuncia de irregularidades [Usar]</li> <li>Identificar ejemplos de cómo interactúa la cultura regional con dilemas éticos [Familiarizarse]</li> <li>Investigar las formas de acoso, discriminación y formas de ayuda [Usar]</li> <li>Examine las diversas formas de acreditación de profesionales [Usar]</li> <li>Explicar la relación entre la ergonomía en los ambientes y la salud de las personas de computación [Usar]</li> <li>Desarrollar un uso del computador/política de uso aceptable con medidas coercitivas [Familiarizarse]</li> <li>Describir los problemas asociados con la presión de la industrias para centrarse en el tiempo de comercialización en comparación con la aplicación de normas de calidad profesional [Usar]</li> </ul>

Lecturas : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]

<b>Unidad 5: Propiedad Intelectual (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fundamentos filosóficos de propiedad intelectual.</li> <li>• Derechos de propiedad intelectual.</li> <li>• Propiedad intelectual digital intangible (IDIP).</li> <li>• Fundamentos legales para protección de la propiedad intelectual.</li> <li>• Gestión de derechos digitales.</li> <li>• Copyrights, patentes, secretos de comercio, marcas registradas.</li> <li>• Plagiarismo.</li> <li>• Fundamentos del movimiento Open Source.</li> <li>• Piratería de Software.</li> </ul>	<ul style="list-style-type: none"> <li>• Discute la racionalidad de la protección legal de la propiedad intelectual [Evaluar]</li> <li>• Discute las bases filosóficas de la propiedad intelectual [Familiarizarse]</li> <li>• Describe la legislación orientada a los delitos de derechos de autor digitales [Evaluar]</li> <li>• Critica la legislación orientada a los delitos digitales de derechos de autor [Familiarizarse]</li> <li>• Identifica ejemplos contemporáneos de propiedad intelectual digital intangible [Evaluar]</li> <li>• Justifica el uso de material con derechos de autor [Evaluar] [Familiarizarse]</li> <li>• Evalúa los asuntos éticos inherentes a diversos mecanismos de detección de plagio [Familiarizarse]</li> <li>• Interpreta el intento y la implementación de licencias de software [Familiarizarse]</li> <li>• Discute asuntos que involucran la seguridad de patentes en software [Familiarizarse]</li> <li>• Caracteriza y contrasta los conceptos de derechos de autor, patentes y de marcas comerciales [Familiarizarse]</li> <li>• Identifica los objetivos del movimiento de software libre [Evaluar]</li> <li>• Identifica los objetivos del movimiento de software libre [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

**Unidad 6: Privacidad y Libertades Civiles (4 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Fundamentos filosóficos de derechos de privacidad.</li> <li>• Fundamentos legales de protección de privacidad.</li> <li>• Implicaciones de privacidad de recopilación de datos generalizada de bases de datos transaccionales, almacenes de datos, sistemas de vigilancia y la computación en la nube.</li> <li>• Ramificaciones de privacidad diferencial.</li> <li>• Soluciones basadas en la tecnología para la protección de la privacidad.</li> <li>• Legislación de privacidad en áreas de práctica.</li> <li>• Libertades civiles y diferencias culturales.</li> <li>• Libertad de expresión y sus limitaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Discute las bases filosóficas para la protección legal de la privacidad personal [Familiarizarse]</li> <li>• Evalúa soluciones para amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Familiarizarse]</li> <li>• Describe los roles de la recolección de datos en la implementación de sistemas de vigilancia intrusiva (ejm. RFID, reconocimiento de rostro, cobro electrónico, computación móvil) [Familiarizarse]</li> <li>• Describe las ramificaciones de la privacidad diferenciada [Familiarizarse]</li> <li>• Investiga el impacto de soluciones tecnológicas a los problemas de privacidad [Familiarizarse]</li> <li>• Critica la intención, el valor potencial y la implementación de las diversas formas de legislación en privacidad [Familiarizarse]</li> <li>• Identifica estrategias que permitan la apropiada libertad de expresión [Familiarizarse]</li> </ul>

**Lecturas :** [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]

**Unidad 7: Políticas de seguridad, Leyes y crímenes computacionales (2 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Ejemplos de delitos informáticos y reparación legal para delincuentes informáticos.</li> <li>• Ingeniería social, robo de identidad y recuperación.</li> <li>• Tópicos relacionados al uso de acceso indebido y las infracciones y materia de seguridad.</li> <li>• Motivaciones y ramificaciones del ciberterrorismo y el hacking criminal, cracking.</li> <li>• Efectos de malware, como virus, worms y Trojan horses.</li> <li>• Estrategias de prevención de Crimen.</li> <li>• Políticas de Seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>• Listar ejemplos clásicos de delitos informáticos y incidentes de ingeniería social con impacto social [Familiarizarse]</li> <li>• Identificar leyes que se aplican a delitos informáticos [Familiarizarse]</li> <li>• Describir la motivación y ramificaciones de cyberterrorismo y hackeo criminal [Familiarizarse]</li> <li>• Examinar los problemas éticos y legales relacionados con el mal uso de accesos y diversas violaciones en la seguridad [Familiarizarse]</li> <li>• Discutir el rol del profesional en seguridad y los problemas que están envueltos [Familiarizarse]</li> <li>• Investigar medidas que puedan ser consideradas por personas y organizaciones incluyendo al gobierno para prevenir o mitigar efectos indeseables de los delitos informáticos y robo de identidad [Familiarizarse]</li> <li>• Escribir una política de seguridad de una empresa, la cual incluye procedimientos para administrar contraseñas y monitorizar a los empleados [Familiarizarse]</li> </ul>

**Lecturas :** [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]

<b>Unidad 8: Economía de la Computación (2 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Monopolio y sus implicaciones económicas.</li> <li>• Efecto del suministro de mano de obra calificada y la demanda sobre la calidad de los productos de computación.</li> <li>• Estrategias de precio en el dominio de la computación.</li> <li>• El fenómeno del desarrollo de software outsourcing y off-shoring; impactos en el empleo y la economía.</li> <li>• Consecuencias de la globalización para la profesión de Ciencias de la Computación.</li> <li>• Diferencias en acceso a recursos de computación y el posible efecto de los mismos.</li> <li>• Análisis costo/beneficio de trabajos con consideraciones para manufactura, hardware, software e implicaciones de ingeniería.</li> <li>• Costo estimado versus costo actual en relación al costo total.</li> <li>• Emprendimiento: perspectivas y entrampamientos.</li> <li>• Efectos de red o economías de escala del lado de la demanda.</li> <li>• El uso de la ingeniería económica para hacer frente a las finanzas.</li> </ul>	<ul style="list-style-type: none"> <li>• Resumir los fundamentos para los esfuerzos antimonopolio [Familiarizarse]</li> <li>• Identificar diversas maneras en que la industria de la tecnología de la información está afectada por la escasez de la oferta de trabajo [Familiarizarse]</li> <li>• Identificar la evolución de la estrategia de precios para el cálculo de los bienes y servicios [Familiarizarse]</li> <li>• Discutir los beneficios, los inconvenientes y las implicaciones de <i>off-shoring</i> y <i>outsourcing</i> [Familiarizarse]</li> <li>• Investigar y defender maneras de tratar las limitaciones en el acceso a la computación. [Usar]</li> <li>• Describir los beneficios económicos de efectos de la red [Usar]</li> </ul>
<b>Lecturas :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [McL00] Raymond McLeod Jr. *Sistemas de Información Gerencial*. Prentice Hall, 2000.
- [LL04] Kenneth C. Laudon and Jane P. Laudon. *Sistemas de Información Gerencial*. Prentice Hall, 2004.
- [Edi09a] Datamation Ediciones, ed. *Revista Datamation MC Ediciones*. 2009.
- [Edi09b] Datamation Ediciones, ed. *Understanding the Digital Economy*. 2009.
- [Edi10] Datamation Ediciones, ed. *Financial Times Mastering Information Management*. 2010.

## 1. CURSO

CS3I1-CS2023. Seguridad en Computación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS3I1-CS2023. Seguridad en Computación
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS231. Redes y Comunicación. (6 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Hoy en dia la información es uno de los activos máspreciados en cualquier organización. Este cursos está orientado a poder brindar al alumno los elementos de seguridad orientados a proteger la información de la organización y principalmente poder prever los posibles problemas relacionados con este rubro. Esta materia involucra el desarrollo de una actitud preventiva por parte del alumno en todas las áreas relacionadas al desarrollo de software.

## 5. OBJETIVOS

- Discutir a un nivel intermedio avanzado los los fundamentos de la Seguridad Informática.
- Brindar los diferentes aspectos que presenta el código malicioso.
- Que el alumno conozca los conceptos de criptografía y seguridad en redes de computadoras.
- Discutir y analizar junto con el alumno los aspectos de la Seguridad en Internet.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**4)** Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Familiarity)

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

## 7. TEMAS

**Unidad 1: Fundamentos y Conceptos en Seguridad (5 horas)****Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• CIA (Confidencialidad, Integridad, Disponibilidad)</li><li>• Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque .</li><li>• Autenticación y autorización, control de acceso (vs. obligatoria discrecional)</li><li>• Concepto de la confianza y la honradez .</li><li>• Ética (revelación responsable)</li></ul>	<ul style="list-style-type: none"><li>• Analizar las ventajas y desventajas de equilibrar las propiedades clave de seguridad(Confidencialidad, Integridad, Disponibilidad) [Familiarizarse]</li><li>• Describir los conceptos de riesgo, amenazas, vulnerabilidades y vectores de ataque(incluyendo el hecho de que no existe tal cosa como la seguridad perfecta) [Familiarizarse]</li><li>• Explicar los conceptos de autentificación, autorización, control de acceso [Familiarizarse]</li><li>• Explicar el concepto de confianza y confiabilidad [Familiarizarse]</li><li>• Reconocer de que hay problemas éticos más importantes que considerar en seguridad computacional, incluyendo problemas éticos asociados a arreglar o no arreglar vulnerabilidades y revelar o no revelar vulnerabilidades [Familiarizarse]</li></ul>

**Lecturas :** [Sta17], [KR17]

**Unidad 2: Principios de Diseño Seguro (5 horas)**  
**Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● Menor privilegio y aislamiento.</li> <li>● Valores predeterminados a prueba de fallos.</li> <li>● Diseño abierto.</li> <li>● La seguridad de extremo a extremo.</li> <li>● La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas)</li> <li>● Diseño de seguridad.</li> <li>● Las tensiones entre la seguridad y otros objetivos de diseño.</li> <li>● Mediación completa.</li> <li>● El uso de componentes de seguridad vetados.</li> <li>● Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque)</li> <li>● Seguridad utilizable.</li> <li>● Componibilidad de seguridad.</li> <li>● Prevención, detección y disuasión.</li> </ul>	<ul style="list-style-type: none"> <li>● Describir el principio de privilegios mínimos y el aislamiento que se aplican al diseño del sistema [Familiarizarse]</li> <li>● Resumir el principio de prueba de fallos y negar por defecto [Familiarizarse]</li> <li>● Discutir las implicaciones de depender de diseño abierto o secreto de diseño para la seguridad [Familiarizarse]</li> <li>● Explicar los objetivos de seguridad de datos de extremo a extremo [Familiarizarse]</li> <li>● Discutir los beneficios de tener múltiples capas de defensas [Familiarizarse]</li> <li>● Por cada etapa en el ciclo de vida de un producto, describir que consideraciones de seguridad deberían ser evaluadas [Familiarizarse]</li> <li>● Describir el costo y ventajas y desventajas asociadas con el diseño de seguridad de un producto. [Familiarizarse]</li> <li>● Describir el concepto de mediación y el principio de mediación completa [Familiarizarse]</li> <li>● Conocer los componentes estándar para las operaciones de seguridad, en lugar de reinventar las operaciones fundamentales [Familiarizarse]</li> <li>● Explicar el concepto de computación confiable incluyendo base informática confiable y de la superficie de ataque y el principio de minimización de base informática confiable [Familiarizarse]</li> <li>● Discutir la importancia de la usabilidad en el diseño de mecanismos de seguridad [Familiarizarse]</li> <li>● Describir problemas de seguridad que surgen en los límites entre varios componentes [Familiarizarse]</li> <li>● Identificar los diferentes roles de mecanismos de prevención y mecanismos de eliminación/disuasión [Familiarizarse]</li> </ul>

**Lecturas :** [Sta17]

<b>Unidad 3: Programación Defensiva (5 horas)</b> <b>Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>• Validación de datos de entrada y sanitización</li> <li>• Elección del lenguaje de programación y lenguajes con tipos de datos seguro.</li> <li>• Ejemplos de validación de entrada de datos y sanitización de errores. <ul style="list-style-type: none"> <li>– Desbordamiento de búfer</li> <li>– Errores enteros</li> <li>– Inyección SQL</li> <li>– Vulnerabilidad XSS</li> </ul> </li> <li>• Las condiciones de carrera.</li> <li>• Manejo correcto de las excepciones y comportamientos inesperados.</li> <li>• Uso correcto de los componentes de terceros.</li> <li>• Desplegar eficazmente las actualizaciones de seguridad.</li> <li>• Información de control de flujo.</li> <li>• Generando correctamente el azar con fines de seguridad.</li> <li>• Mecanismos para la detección y mitigación de datos de entrada y errores de sanitización.</li> <li>• Fuzzing</li> <li>• El análisis estático y análisis dinámico.</li> <li>• Programa de verificación.</li> <li>• Soporte del sistema operativo (por ejemplo, la asignación al azar del espacio de direcciones, canarios)</li> <li>• El soporte de hardware (por ejemplo, el DEP, TPM)</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar por que la validación de entrada y desinfección de datos es necesario en el frente del control contencioso del canal de entrada [Usar]</li> <li>• Explicar por que uno debería escoger para desarrollar un programa en un lenguaje tipo seguro como Java, en contraste con un lenguaje de programación no seguro como C/C++ [Usar]</li> <li>• Clasificar los errores de validación de entrada común, y escribir correctamente el código de validación de entrada [Usar]</li> <li>• Demostrar el uso de un lenguaje de programación de alto nivel cómo prevenir una condición de competencia que ocurran y cómo manejar una excepción [Usar]</li> <li>• Demostrar la identificación y el manejo elegante de las condiciones de error [Familiarizarse]</li> <li>• Explique los riesgos de mal uso de las interfaces con código de terceros y cómo utilizar correctamente el código de terceros [Familiarizarse]</li> <li>• Discutir la necesidad de actualizar el software para corregir las vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Familiarizarse]</li> </ul>

**Lecturas :** [Sta17]

<b>Unidad 4: Ataques y Amenazas (5 horas)</b> <b>Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Atacante metas, capacidades y motivaciones (como economía sumergida, el espionaje digital, la guerra cibernetica, las amenazas internas, hacktivismo, las amenazas persistentes avanzadas)</li> <li>Los ejemplos de malware (por ejemplo, virus, gusanos, spyware, botnets, troyanos o rootkits)</li> <li>Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS)</li> <li>Ingeniería social (por ejemplo, perscando)</li> <li>Los ataques a la privacidad y el anonimato .</li> <li>El malware / comunicaciones no deseadas, tales como canales encubiertos y esteganografía.</li> </ul>	<ul style="list-style-type: none"> <li>Describir tipos de ataques similares en contra de un sistema en particular [Familiarizarse]</li> <li>Discutir los limitantes de las medidas en contra del malware (ejm. detección basada en firmas, detección de comportamiento) [Familiarizarse]</li> <li>Identificar las instancias de los ataques de ingeniería social y de los ataques de negación de servicios [Familiarizarse]</li> <li>Discutir como los ataques de negación de servicios puede ser identificados y reducido [Familiarizarse]</li> <li>Describir los riesgos de la privacidad y del anonimato en aplicaciones comunmente usadas [Familiarizarse]</li> <li>Discutir los conceptos de conversión de canales y otros procedimientos de filtrado de datos [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Abd+21]	

<b>Unidad 5: Seguridad de Red (4 horas)</b> <b>Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>Red de amenazas y tipos de ataques específicos (por ejemplo, la denegación de servicio, spoofing, olfateando y la redirección del tráfico, el hombre en el medio, ataques integridad de los mensajes, los ataques de enrutamiento, y el análisis de tráfico)</li> <li>El uso de cifrado de datos y seguridad de la red .</li> <li>Arquitecturas para redes seguras (por ejemplo, los canales seguros, los protocolos de enrutamiento seguro, DNS seguro, VPN, protocolos de comunicación anónimos, aislamiento)</li> <li>Los mecanismos de defensa y contramedidas (por ejemplo, monitoreo de red, detección de intrusos, firewalls, suplantación de identidad y protección DoS, honeypots, seguimientos)</li> <li>Seguridad para redes inalámbricas, celulares .</li> <li>Otras redes no cableadas (por ejemplo, ad hoc, sensor, y redes vehiculares)</li> <li>Resistencia a la censura.</li> <li>Gestión de la seguridad operativa de la red (por ejemplo, control de acceso a la red configure)</li> </ul>	<ul style="list-style-type: none"> <li>Describir las diferentes categorías de amenazas y ataques en redes [Familiarizarse]</li> <li>Describir las arquitecturas de criptografía de clave pública y privada y cómo las ICP brindan apoyo a la seguridad en redes [Familiarizarse]</li> <li>Describir ventajas y limitaciones de las tecnologías de seguridad en cada capa de una torre de red [Familiarizarse]</li> <li>Identificar los adecuados mecanismos de defensa y sus limitaciones dada una amenaza de red [Usar]</li> </ul>
<b>Lecturas :</b> [KR17], [Uni15]	

<b>Unidad 6: Criptografía (5 horas)</b> <b>Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>● Terminología básica de criptografía cubriendo las nociones relacionadas con los diferentes socios (comunicación), canal seguro / inseguro, los atacantes y sus capacidades, cifrado, descifrado, llaves y sus características, firmas.</li> <li>● Tipos de cifrado (por ejemplo, cifrado César, cifrado affine), junto con los métodos de ataque típicas como el análisis de frecuencia.</li> <li>● Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos.</li> <li>● Criptografía de clave simétrica: <ul style="list-style-type: none"> <li>– El secreto perfecto y el cojín de una sola vez</li> <li>– Modos de funcionamiento para la seguridad semántica y encriptación autenticada (por ejemplo, cifrar-entonces-MAC, OCB, GCM)</li> <li>– Integridad de los mensajes (por ejemplo, CMAC, HMAC)</li> </ul> </li> <li>● La criptografía de clave pública: <ul style="list-style-type: none"> <li>– Permutación de trampilla, por ejemplo, RSA</li> <li>– Cifrado de clave pública, por ejemplo, el cifrado RSA, cifrado El Gamal</li> <li>– Las firmas digitales</li> <li>– Infraestructura de clave pública (PKI) y certificados</li> <li>– Supuestos de dureza, por ejemplo, Diffie-Hellman, factoring entero</li> </ul> </li> <li>● Protocolos de intercambio de claves autenticadas, por ejemplo, TLS .</li> <li>● Primitivas criptográficas: <ul style="list-style-type: none"> <li>– generadores pseudo-aleatorios y cifrados de flujo</li> <li>– cifrados de bloque (permutaciones pseudo-aleatorios), por ejemplo, AES</li> <li>– funciones de pseudo-aleatorios</li> <li>– funciones de hash, por ejemplo, SHA2, resistencia colisión</li> <li>– códigos de autenticación de mensaje</li> <li>– funciones derivaciones clave</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse]</li> <li>● Definir los siguientes términos: Cifrado, Criptoanálisis, Algoritmo Criptográfico, y Criptología y describe dos métodos básicos (cifrados) para transformar texto plano en un texto cifrado [Familiarizarse]</li> <li>● Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Familiarizarse]</li> <li>● Ilustrar como medir la entropía y como generar aleatoriedad criptográfica [Usar]</li> <li>● Usa primitivas de clave pública y sus aplicaciones [Usar]</li> <li>● Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse]</li> <li>● Discutir protocolos criptográficos y sus propiedades [Familiarizarse]</li> </ul>

**Lecturas :** [PP10]

<b>Unidad 7: Seguridad en la Web (5 horas)</b> <b>Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>● Modelo de seguridad Web <ul style="list-style-type: none"> <li>– Modelo de seguridad del navegador incluida la política de mismo origen</li> <li>– Los límites de confianza de cliente-servidor, por ejemplo, no pueden depender de la ejecución segura en el cliente</li> </ul> </li> <li>● Gestión de sesiones, la autenticación: <ul style="list-style-type: none"> <li>– Single Sign-On</li> <li>– HTTPS y certificados</li> </ul> </li> <li>● Vulnerabilidades de las aplicaciones y defensas : <ul style="list-style-type: none"> <li>– Inyección SQL</li> <li>– XSS</li> <li>– CSRF</li> </ul> </li> <li>● Seguridad del lado del cliente : <ul style="list-style-type: none"> <li>– Política de seguridad Cookies</li> <li>– Extensiones de seguridad HTTP, por ejemplo HSTS</li> <li>– Plugins, extensiones y aplicaciones web</li> <li>– Seguimiento de los usuarios Web</li> </ul> </li> <li>● Herramientas de seguridad del lado del servidor, por ejemplo, los cortafuegos de aplicación Web (WAFS) y fuzzers</li> </ul>	<ul style="list-style-type: none"> <li>● Describe el modelo de seguridad de los navegadores incluyendo las políticas del mismo origen y modelos de amenazas en seguridad web [Familiarizarse]</li> <li>● Discutir los conceptos de sesiones web, canales de comunicación seguros tales como Seguridad en la Capa de Transporte(<i>TLS</i>) y la importancia de certificados de seguridad, autenticación incluyendo inicio de sesión único, como OAuth y Lenguaje de Marcado para Confirmaciones de Seguridad(<i>SAML</i>) [Familiarizarse]</li> <li>● Investigar los tipos comunes de vulnerabilidades y ataques en las aplicaciones web, y defensas contra ellos [Familiarizarse]</li> <li>● Utilice las funciones de seguridad del lado del cliente [Usar]</li> </ul>

**Lecturas :** [Uni15]

**Unidad 8: Seguridad de plataformas (5 horas)**  
**Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Integridad de código y firma de código.</li> <li>• Arranque seguro, arranque medido, y la raíz de confianza.</li> <li>• Testimonio.</li> <li>• TPM y coprocesadores seguros.</li> <li>• Las amenazas de seguridad de los periféricos, por ejemplo, DMA, IOMMU.</li> <li>• Ataques físicos: troyanos de hardware, sondas de memoria, ataques de arranque en frío.</li> <li>• Seguridad de dispositivos integrados, por ejemplo, dispositivos médicos, automóviles.</li> <li>• Ruta confiable.</li> </ul>	<ul style="list-style-type: none"> <li>• Explica el concepto de integridad de código y firma de códigos, así como el alcance al cual se aplica [Familiarizarse]</li> <li>• Discute los conceptos del origen de la confidencialidad y el de los procesos de arranque y carga segura [Familiarizarse]</li> <li>• Describe los mecanismos de arresto remoto de la integridad de un sistema [Familiarizarse]</li> <li>• Resume las metas y las primitivas claves de los modelos de plataforma confiable (TPM) [Familiarizarse]</li> <li>• Identifica las amenazas de conectar periféricos en un dispositivo [Familiarizarse]</li> <li>• Identifica ataques físicos y sus medidas de control [Familiarizarse]</li> <li>• Identifica ataques en plataformas con hardware que no son del tipo PC [Familiarizarse]</li> <li>• Discute los conceptos y la importancia de ruta confiable [Familiarizarse]</li> </ul>

**Lecturas :** [Sta17]

**Unidad 9: Investigación digital (Digital Forensics) (5 horas)**

**Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Principios básicos y metodologías de análisis digital forense.</li> <li>Diseñar sistemas con necesidades forenses en mente.</li> <li>Reglas de Evidencia - conceptos generales y las diferencias entre las jurisdicciones y la Cadena de Custodia.</li> <li>Búsqueda y captura de comprobación: requisitos legales y de procedimiento.</li> <li>Métodos y normas de evidencia digital.</li> <li>Las técnicas y los estándares para la conservación de los datos.</li> <li>Cuestiones legales y reportes incluyendo el trabajo como perito.</li> <li>Investigación digital de los sistema de archivos.</li> <li>Los forenses de aplicación.</li> <li>Investigación digital en la web.</li> <li>Investigación digital en redes.</li> <li>Investigación digital en dispositivos móviles.</li> <li>Ataques al computador/red/sistema.</li> <li>Detección e investigación de ataque.</li> <li>Contra investigación digital.</li> </ul>	<ul style="list-style-type: none"> <li>Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse]</li> <li>Explica como diseñar software de apoyo a técnicas forenses [Familiarizarse]</li> <li>Describe los requisitos legales para usar datos recuperados [Familiarizarse]</li> <li>Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse]</li> <li>Describe como se realiza la recolección de datos y el adecuado almacenamiento de los datos originales y de la copia forense [Familiarizarse]</li> <li>Realiza recolección de datos en un disco duro [Usar]</li> <li>Describe la responsabilidad y obligación de una persona mientras testifica como un examinador forense [Familiarizarse]</li> <li>Recupera datos basados en un determinado término de búsqueda en una imagen del sistema [Usar]</li> <li>Reconstruye el historial de una aplicación a partir de los artefactos de la aplicación [Familiarizarse]</li> <li>Reconstruye el historial de navegación web de los artefactos web [Familiarizarse]</li> <li>Captura e interpreta el tráfico de red [Familiarizarse]</li> <li>Discute los retos asociados con técnicas forenses de dispositivos móviles [Familiarizarse]</li> </ul>

**Lecturas :** [Sta17]

**Unidad 10: Seguridad en Ingeniería de Software (4 horas)**

**Resultados esperados: 2,4,AG-C02,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>La construcción de la seguridad en el ciclo de vida de desarrollo de software.</li> <li>Principios y patrones de diseño seguros.</li> <li>Especificaciones de software seguros y requisitos.</li> <li>Prácticas de desarrollo de software de seguros.</li> <li>Asegure probar el proceso de las pruebas de que se cumplan los requisitos de seguridad (incluyendo análisis estático y dinámico)</li> </ul>	<ul style="list-style-type: none"> <li>Describir los requisitos para la integración de la seguridad en el SDL [Familiarizarse]</li> <li>Aplicar los conceptos de los principios de diseño para mecanismos de protección, los principios para seguridad de software (Viega and McGraw) y los principios de diseño de seguridad (Morrie Gasser) en un proyecto de desarrollo de software [Familiarizarse]</li> <li>Desarrollar especificaciones para un esfuerzo de desarrollo de software que especifica completamente los requisitos funcionales y se identifican las rutas de ejecución esperadas [Familiarizarse]</li> </ul>

**Lecturas :** [Sta17]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [PP10] Christof Paar and Jan Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Corrected edition. Springer, 2010. URL: [https://www.academia.edu/18966194/Understanding\\_Cryptography\\_A\\_Textbook\\_for\\_Students\\_and\\_Practitioners](https://www.academia.edu/18966194/Understanding_Cryptography_A_Textbook_for_Students_and_Practitioners).
- [Uni15] Babasaheb Ambedkar Open University. *Fundamentals of Computer Networking*. PGDCA201. Course material for Post Graduate Diploma in Computer Applications. Babasaheb Ambedkar Open University, 2015. URL: [https://baou.edu.in/assets/pdf/PGDCA201\\_slm.pdf](https://baou.edu.in/assets/pdf/PGDCA201_slm.pdf).
- [KR17] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*. 7th. 7th edition. Pearson, 2017. URL: [https://www.ucg.ac.me/skladiste/blog\\_44233/objava\\_64433/fajlovi/Computer%20Networking%20\\_%20A%20Top%20Down%20Approach,%207th,%20converted.pdf](https://www.ucg.ac.me/skladiste/blog_44233/objava_64433/fajlovi/Computer%20Networking%20_%20A%20Top%20Down%20Approach,%207th,%20converted.pdf).
- [Sta17] William Stallings. *Cryptography and Network Security: Principles and Practice*. 7th. Global edition. Pearson, 2017. URL: <http://www.cs.vsb.cz/ochodkova/courses/kpb/cryptography-and-network-security-principles-and-practice-7th-global-edition.pdf>.
- [Abd+21] N. Abdullah et al. *Advances in Cyber Security*. Ed. by N. Abdullah et al. Vol. 788. Lecture Notes in Electrical Engineering. Springer, 2021. doi: 10.1007/978-981-16-8059-5. URL: <https://doi.org/10.1007/978-981-16-8059-5>.

## 1. CURSO

CS3P1-CS2023. Computación Paralela y Distribuida (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS3P1-CS2023. Computación Paralela y Distribuida
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	<ul style="list-style-type: none"><li>CS212. Análisis y Diseño de Algoritmos. (5<sup>to</sup> Sem)</li><li>CS231. Redes y Comunicación. (6<sup>to</sup> Sem)</li></ul>

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La última década ha traído un crecimiento explosivo en computación con multiprocesadores, incluyendo los procesadores de varios núcleos y centros de datos distribuidos. Como resultado, la computación paralela y distribuida se ha convertido de ser un tema ampliamente electivo para ser uno de los principales componentes en la malla estudios en ciencia de la computación de pregrado. Tanto la computación paralela como la distribuida implica la ejecución simultánea de múltiples procesos, cuyas operaciones tienen el potencial para intercalar de manera compleja. La computación paralela y distribuida construye sobre cimientos en muchas áreas, incluyendo la comprensión de los conceptos fundamentales de los sistemas, tales como: concurrencia y ejecución en paralelo, consistencia en el estado/manipulación de la memoria, y latencia. La comunicación y la coordinación entre los procesos tiene sus cimientos en el paso de mensajes y modelos de memoria compartida de la computación y conceptos algorítmicos como atomicidad, el consenso y espera condicional. El logro de aceleración en la práctica requiere una comprensión de algoritmos paralelos, estrategias para la descomposición problema, arquitectura de sistemas, estrategias de implementación y análisis de rendimiento. Los sistemas distribuidos destacan los problemas de la seguridad y tolerancia a fallos, hacen hincapié en el mantenimiento del estado replicado e introducen problemas adicionales en el campo de las redes de computadoras.

## 5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente máquinas con múltiples núcleos.
- Que el alumno sea capaz de comparar aplicaciones secuenciales y paralelas.
- Que el alumno sea capaz de convertir, cuando la situación lo amerite, aplicaciones secuenciales a paralelas de forma eficiente.

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos de paralelismo (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Procesamiento Simultáneo Múltiple.</li><li>• Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos)</li><li>• Paralelismo, comunicación, y coordinación:<ul style="list-style-type: none"><li>– Paralelismo, comunicación, y coordinación</li><li>– Necesidad de Sincronización</li></ul></li><li>• Errores de Programación ausentes en programación secuencial:<ul style="list-style-type: none"><li>– Tipos de Datos ( lectura/escritura simultánea o escritura/escritura compartida)</li><li>– Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado)</li><li>– Falta de vida/progreso (deadlock, starvation)</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse]</li><li>• Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse]</li><li>• Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]</li></ul>
<b>Lecturas :</b> [Pac11], [Mat14], [Qui03], [HW10]	

<b>Unidad 2: Arquitecturas paralelas (12 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Procesadores multinúcleo.</li> <li>• Memoria compartida vs memoria distribuida.</li> <li>• Multiprocesamiento simétrico.</li> <li>• SIMD, procesamiento de vectores.</li> <li>• GPU, coprocesamiento.</li> <li>• Taxonomía de Flynn.</li> <li>• Soporte a nivel de instrucciones para programación paralela. <ul style="list-style-type: none"> <li>– Instrucciones atómicas como Compare/Set (Comparar / Establecer)</li> </ul> </li> <li>• Problemas de Memoria: <ul style="list-style-type: none"> <li>– Caches multiprocesador y coherencia de cache</li> <li>– Acceso a Memoria no uniforme (NUMA)</li> </ul> </li> <li>• Topologías. <ul style="list-style-type: none"> <li>– Interconexiones</li> <li>– Clusters</li> <li>– Compartir recursos (p.e., buses e interconexiones)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar]</li> <li>• Describir la arquitectura SMP y observar sus principales características [Evaluar]</li> <li>• Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar]</li> <li>• Describir las ventajas y limitaciones de GPUs vs CPUs [Usar]</li> <li>• Explicar las características de cada clasificación en la taxonomía de Flynn [Usar]</li> <li>• Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]</li> <li>• Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]</li> </ul>

**Lecturas :** [Pac11], [KH13], [SK10], [HW10]

<b>Unidad 3: Descomposición en paralelo (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Necesidad de Comunicación y coordinación/sincronización.</li> <li>• Independencia y Particionamiento.</li> <li>• Conocimiento Básico del Concepto de Descomposición Paralela.</li> <li>• Decomposición basada en tareas: <ul style="list-style-type: none"> <li>– Implementación de estrategias como hebras</li> </ul> </li> <li>• Descomposición de Información Paralela <ul style="list-style-type: none"> <li>– Estrategias como SIMD y MapReduce</li> </ul> </li> <li>• Actores y Procesos Reactivos (solicitud de gestores)</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar]</li> <li>• Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse]</li> <li>• Escribir un algoritmo paralelo correcto y escalable [Usar]</li> <li>• Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar]</li> <li>• Paralelizar un algoritmo mediante la aplicación de descomposición datos en paralelo [Usar]</li> <li>• Escribir un programa usando actores y/o procesos reactivos [Usar]</li> </ul>

**Lecturas :** [Pac11], [Mat14], [Qui03], [HW10]

## Unidad 4: Comunicación y coordinación (18 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● Memoria Compartida.</li> <li>● La consistencia, y su papel en los lenguajes de programación garantizan las condiciones de carrera para los programas de carrera libre.</li> <li>● Pasos de Mensaje: <ul style="list-style-type: none"> <li>– Mensajes Punto a Punto versus multicast (o basados en eventos)</li> <li>– Estilos para enviar y recibir mensajes Blocking vs non-blocking</li> <li>– Buffering de mensajes</li> </ul> </li> <li>● Atomicidad: <ul style="list-style-type: none"> <li>– Especificar y probar atomicidad y requerimientos de seguridad</li> <li>– Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas</li> <li>– Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas <ul style="list-style-type: none"> <li>* Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención)</li> </ul> </li> <li>– Composición <ul style="list-style-type: none"> <li>* Componiendo acciones atómicas granulares más grandes usando sincronización</li> <li>* Transacciones, incluyendo enfoques optimistas y conservadores</li> </ul> </li> </ul> </li> <li>● Consensos: <ul style="list-style-type: none"> <li>– (Cíclicos) bareras, contadores y estructuras relacionadas</li> </ul> </li> <li>● Acciones condicionales: <ul style="list-style-type: none"> <li>– Espera condicional (p.e., empleando variables de condición)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Usar exclusión mútua para evitar una condición de carrera [Usar]</li> <li>● Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse]</li> <li>● Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar]</li> <li>● Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse]</li> <li>● Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar]</li> <li>● Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse]</li> <li>● Usar semáforos o variables de condición para bloquear hebras hasta una necesaria precondición de mantenga [Usar]</li> </ul>

Lecturas : [Pac11], [Mat14], [Qui03], [HW10]

**Unidad 5: Análisis y programación de algoritmos paralelos (18 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl.</li> <li>• Aceleración y escalabilidad.</li> <li>• Naturalmente (vergonzosamente) algoritmos paralelos.</li> <li>• Patrones Algorítmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) <ul style="list-style-type: none"> <li>– Algoritmos específicos (p.e., MergeSort paralelo)</li> </ul> </li> <li>• Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela)</li> <li>• Cálculos de matriz paralelas.</li> <li>• Productor-consumidor y algoritmos paralelos segmentados.</li> <li>• Ejemplos de algoritmos paralelos no-escalables.</li> </ul>	<ul style="list-style-type: none"> <li>• Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse]</li> <li>• Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar]</li> <li>• Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse]</li> <li>• Identificar tareas independientes en un programa que debe ser paralelizado [Usar]</li> <li>• Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse]</li> <li>• Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su análogo secuencial [Usar]</li> <li>• Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) vía operaciones <i>map</i> y <i>reduce</i> [Usar]</li> <li>• Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar]</li> <li>• Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar]</li> <li>• Implementar un algoritmo de matriz paralela [Usar]</li> <li>• Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]</li> </ul>

**Lecturas :** [Mat14], [Qui03], [HW10]

<b>Unidad 6: Desempeño en paralelo (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Equilibrio de carga.</li> <li>• La medición del desempeño.</li> <li>• Programación y contención.</li> <li>• Evaluación de la comunicación de arriba.</li> <li>• Gestión de datos: <ul style="list-style-type: none"> <li>– Costos de comunicación no uniforme debidos a proximidad</li> <li>– Efectos de Cache (p.e., false sharing)</li> <li>– Manteniendo localidad espacial</li> </ul> </li> <li>• Consumo de energía y gestión.</li> </ul>	<ul style="list-style-type: none"> <li>• Detectar y corregir un desbalanceo de carga [Usar]</li> <li>• Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar]</li> <li>• Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse]</li> <li>• Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar]</li> <li>• Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse]</li> <li>• Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse]</li> <li>• Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]</li> </ul>

**Lecturas :** [Pac11], [Mat14], [KH13], [SK10], [HW10]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003.
- [HW10] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science)*. Ed. by CRC Press. 1st. 2010.
- [SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010.
- [Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011.
- [KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013.
- [Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.

## 1. CURSO

CS402. Proyecto de tesis 1 (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS402. Proyecto de tesis 1
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS401. Metodología de la Investigación. (4 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno pueda realizar un estudio del estado del arte de un tema escogido para investigar.

## 5. OBJETIVOS

- Que el alumno realice una investigación inicial en un tema específico realizando el estudio del estado del arte del tema elegido.
- Que el alumno muestre dominio en el tema de la línea de investigación elegida.
- Que el alumno elija un docente que domine el de investigación elegida como asesor.
- Los entregables de este curso son:

**Avance parcial:** Bibliografía sólida y avance de un Reporte Técnico.

**Final:** Reporte Técnico con experimentos preliminares comparativos que demuestren que el alumno ya conoce las técnicas existentes en el área de su proyecto y elegir a un docente que domine el área de su proyecto como asesor de su proyecto.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

- 2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

- 4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

## 7. TEMAS

<b>Unidad 1: Levantamiento del estado del arte (60 horas)</b>	
<b>Resultados esperados: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Realizar un estudio profundo del estado del arte en un determinado tópico del área de Computación.</li><li>Redacción de artículos técnicos en computación.</li></ul>	<ul style="list-style-type: none"><li>Hacer un levantamiento bibliográfico del estado del arte del tema escogido (esto significa muy probablemente 1 o 2 capítulos de marco teórico además de la introducción que es el capítulo I de la tesis) [Usar]</li><li>Redactar un documento en latex en formato artículo (<i>paper</i>) con mayor calidad que en Proyecto I (dominar tablas, figuras, ecuaciones, índices, bibtex, referencias cruzadas, citaciones, pstricks) [Usar]</li><li>Tratar de hacer las presentaciones utilizando prosper [Usar]</li><li>Mostrar experimentos básicos [Usar]</li><li>Elegir un asesor que domine el área de investigación realizada [Usar]</li></ul>

**Lecturas :** [IEE08], [ACM08], [Cit08]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[ACM08] ACM. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.

## 1. CURSO

FG106. Teatro (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FG106. Teatro
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	1 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Favorece al estudiante a identificarse a la “Comunidad Académica” de la Universidad, en la medida en que le brinda canales naturales de integración a su grupo y a su Centro de Estudios y le permite, desde una visión alternativa, visualizar la valía interior de las personas a su alrededor, a la vez que puede conocer mejor la suya propia. Relaciona al universitario, a través de la experimentación, con un nuevo lenguaje, un medio de comunicación y expresión que va más allá de la expresión verbal conceptualizada. Coadyuva al estudiante en su formación integral, desarrollando en él capacidades corporales. Estimula en él, actitudes anímicas positivas, aptitudes cognitivas y afectivas. Enriquece su sensibilidad y despierta su solidaridad. Desinhibe y socializa, relaja y alegra, abriendo un camino de apertura de conocimiento del propio ser y el ser de los demás.

## 5. OBJETIVOS

- Contribuir a la formación personal y profesional del estudiante, reconociendo, valorando y desarrollando su lenguaje corporal, integrándolo a su grupo, afianzando su seguridad personal, enriqueciendo su intuición, su imaginación y creatividad, motivándolo a abrir caminos de búsqueda de conocimiento de sí mismo y de comunicación con los demás a través de su sensibilidad, de ejercicios de introspección y de nuevas vías de expresión.

## 6. RESULTADOS DEL ESTUDIANTE

3) Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

## 7. TEMAS

<b>Unidad 1: (6 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• ¿Qué es el Arte? Una experiencia vivencial y personal.</li> <li>• La llave maestra: la creatividad.</li> <li>• La importancia del teatro en la formación personal y profesional.</li> <li>• Utilidad y enfoque del arte teatral.</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer la vigencia del Arte y la creatividad en el desarrollo personal y social [Usar].</li> <li>• Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usar].</li> <li>• Reconocer nociones básicas del teatro [Usar].</li> </ul>
<b>Lecturas :</b> [Maj58], [PAVIS]	

<b>Unidad 2: (6 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Juego, luego existo.</li> <li>• El juego del niño y el juego dramático.</li> <li>• Juegos de integración grupal y juegos de creatividad.</li> <li>• La secuencia teatral.</li> </ul>	<ul style="list-style-type: none"> <li>• Reconocer el juego como herramienta fundamental del teatro [Usar].</li> <li>• Interiorizar y revalorar el juego como aprendizaje creativo [Usar].</li> <li>• Acercar al estudiante de manera espontánea y natural, a la vivencia teatral [Usar].</li> </ul>
<b>Lecturas :</b> [Maj58], [PAVIS]	

<b>Unidad 3: (9 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Toma de conciencia del cuerpo.</li> <li>• Toma de conciencia del espacio</li> <li>• Toma de conciencia del tiempo</li> <li>• Creación de secuencias individuales y colectivas: Cuerpo, espacio y tiempo.</li> <li>• El uso dramático del elemento: El juego teatral.</li> <li>• Presentaciones teatrales con el uso del elemento.</li> </ul>	<ul style="list-style-type: none"> <li>• Experimentar con nuevas formas de expresión y comunicación [Usar].</li> <li>• Conocer algunos mecanismos de control y manejo corporal [Usar].</li> <li>• Brindar caminos para que el alumno pueda desarrollar creativamente su imaginación, su capacidad de relación y captación de estímulos auditivos, rítmicos y visuales [Usar].</li> <li>• Conocer y desarrollar el manejo de su espacio propio y de sus relaciones espaciales [Usar].</li> <li>• Experimentar estados emocionales diferentes y climas colectivos nuevos [Usar].</li> </ul>
<b>Lecturas :</b> [Maj58], [PAVIS]	

<b>Unidad 4: (12 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Relajación, concentración y respiración.</li> <li>• Desinhibición e interacción con el grupo.</li> <li>• La improvisación.</li> <li>• Equilibrio, peso, tiempo y ritmo.</li> <li>• Análisis del movimiento. Tipos de movimiento.</li> <li>• La presencia teatral.</li> <li>• La danza, la coreografía teatral.</li> </ul>	<ul style="list-style-type: none"> <li>• Ejercitarse en el manejo de destrezas comunicativas no verbales [Usar].</li> <li>• Practicar juegos y ejercicios de lenguaje corporal, individual y grupalmente [Usar].</li> <li>• Expresar libre y creativamente sus emociones y sentimientos y su visión de la sociedad a través de representaciones originales con diversos lenguajes [Usar].</li> <li>• Conocer los tipos de actuación [Usar].</li> </ul>

Lecturas : [Maj58], [PAVIS]

<b>Unidad 5: (3 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• El origen del teatro, el teatro griego y el teatro romano.</li> <li>• El teatro medieval , la comedia del arte.</li> <li>• De la pasión a la razón: Romanticismo e Ilustración.</li> <li>• El teatro realista, teatro épico. Brech y Stanislavski.</li> <li>• El teatro del absurdo, teatro contemporáneo y teatro total.</li> <li>• Teatro en el Perú: Yuyashkani, La Tarumba, pataclau, otros.</li> </ul>	<ul style="list-style-type: none"> <li>• Conocer la influencia que la sociedad ha ejercido en el teatro y la respuesta de este arte ante los diferentes momentos de la historia [Usar].</li> <li>• Apreciar el valor y aporte de las obras de dramaturgos importantes [Usar].</li> <li>• Analizar el contexto social del arte teatral [Usar].</li> <li>• Reflexionar sobre el Teatro Peruano y arequipeño [Usar].</li> </ul>

Lecturas : [Maj58], [PAVIS]

<b>Unidad 6: (12 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Apreciación teatral. Expectación de una o más obras teatrales.</li> <li>• El espacio escénico.</li> <li>• Construcción del personaje</li> <li>• Creación y montaje de una obra teatral .</li> <li>• Presentación en público de pequeñas obras haciendo uso de vestuario, maquillaje, escenografía, utilería y del empleo dramático del objeto.</li> </ul>	<ul style="list-style-type: none"> <li>• Emplear la creación teatral, como manifestación de ideas y sentimientos propios ante la sociedad [Usar].</li> <li>• Aplicar las técnicas practicadas y los conocimientos aprendidos en una apreciación y/o expresión teatral concreta que vincule el rol de la educación [Usar].</li> <li>• Intercambiar experiencias y realizar presentaciones breves de ejercicios teatrales en grupo, frente a público [Usar].</li> </ul>

Lecturas : [Maj58], [PAVIS]

## 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**

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## **10. BIBLIOGRAFÍA BÁSICA**

[Maj58] A. Majorana. *El arte de hablar en público*. La España Moderna, 1958.

## 1. CURSO

EX301FCCS. Actividades Extracurriculares (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	EX301FCCS. Actividades Extracurriculares
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	1 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	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

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Familiarity)

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

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

## 7. TEMAS

<b>Unidad 1: Actividades Extracurriculares (48 horas)</b> <b>Resultados esperados: 3,5,AG-C03,AG-C04</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Talleres técnicos o interdisciplinarios.</li> <li>• Asistencia a congresos, seminarios o charlas.</li> <li>• Proyección social (voluntariados, mentorías).</li> <li>• Competencias académicas (hackathons, ferias científicas).</li> </ul>	<ul style="list-style-type: none"> <li>• Demostrar participación en al menos 3 actividades [Familiarizarse].</li> <li>• Elaborar informes reflexivos sobre las experiencias [Usar].</li> <li>• Colaborar en equipos para proyectos sociales o técnicos [Evaluar].</li> </ul>
<b>Lecturas :</b> [ACM19], [ABE22]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [ACM19] ACM. *Guidelines for Extracurricular Activities in Computing*. Tech. rep. 2019. URL: <https://www.acm.org/education/curricula-recommendations>.
- [ABE22] ABET. *ABET Criteria for Student Professional Development*. Tech. rep. 2022. URL: <https://www.abet.org/accreditation/>.

## 1. CURSO

CS361-CS2023. Visión Computacional (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS361-CS2023. Visión Computacional
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI264. Aprendizaje Profundo. (7 <sup>mo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso cubre técnicas fundamentales para el análisis automático de imágenes digitales, esencial en aplicaciones como diagnóstico médico, vehículos autónomos y sistemas de vigilancia. Alinea con los estándares ACM/IEEE-CS para visión por computadora.

## 5. OBJETIVOS

- Implementar algoritmos de extracción de características y reconocimiento de objetos usando OpenCV/Python.
- Evaluar métodos de deep learning para segmentación semántica (ej. Mask R-CNN).

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**1)** Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos de Imágenes Digitales (16 horas)</b> <b>Resultados esperados: 1,6,AG-C08</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Filtrado espacial (Gaussiano, Sobel)</li> <li>• Transformaciones geométricas y morfológicas</li> <li>• Espacios de color (RGB, HSV, LAB)</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar operaciones básicas de procesamiento de imágenes [Usar]</li> <li>• Calibrar parámetros de filtros para casos reales [Evaluar]</li> </ul>

Lecturas : [Sze10], [GW18]

<b>Unidad 2: Geometría Epipolar y Reconstrucción (16 horas)</b> <b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Matriz fundamental y esencial</li> <li>• Triangulación y estructura desde movimiento</li> <li>• Nube de puntos con Open3D</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar pipelines de reconstrucción 3D [Usar]</li> <li>• Documentar resultados técnicos en informes [Evaluar]</li> </ul>

Lecturas : [HZ04]

<b>Unidad 3: Redes Neuronales para Visión (16 horas)</b> <b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Arquitecturas CNN (ResNet, YOLO)</li> <li>• Transfer learning con TensorFlow</li> <li>• Segmentación semántica (U-Net)</li> </ul>	<ul style="list-style-type: none"> <li>• Entrenar modelos para clasificación de imágenes [Usar]</li> <li>• Colaborar en equipos para proyectos integrados [Usar]</li> </ul>

Lecturas : [He+17]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [HZ04] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.
- [Sze10] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010. DOI: 10.1007/978-1-84882-935-0. URL: <https://szeliski.org/Book/>.
- [He+17] Kaiming He et al. “Mask R-CNN”. In: *IEEE ICCV* (2017). URL: <https://arxiv.org/abs/1703.06870>.
- [GW18] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. 4th. Pearson, 2018.

## 1. CURSO

CS391-CS2023. Ingeniería de Software III (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS391-CS2023. Ingeniería de Software III
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS292. Ingeniería de Software II. (7 <sup>mo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La ingeniería de software evoluciona hacia el paradigma AI-Native donde agentes autónomos co-crean software de forma segura y observable. Este curso forma ingenieros capaces de diseñar plataformas de desarrollo aumentadas con IA, implementar LLMOps/AgentOps y garantizar calidad en sistemas donde gran parte del ciclo de vida está automatizado por inteligencia artificial.

## 5. OBJETIVOS

- Diseñar y operar Internal Developer Platforms aumentadas con agentes autónomos que permitan crear servicios completos desde lenguaje natural con trazabilidad y gobernanza.
- Implementar pipelines completos de LLMOps y AgentOps con versionado de prompts, registro de modelos, evaluación continua y orquestación segura de flotas de agentes.
- Desarrollar sistemas de evaluación automática para software generado por IA usando benchmarks propietarios, mutation testing y red teaming de prompts y agentes.
- Integrar agentes como participantes activos en metodologías ágiles modernas con planning poker automatizado, retrospectivas con análisis de sentimientos y generación de documentación.
- Alcanzar niveles SLSA 3-4 y cumplimiento NIST SSDF en repositorios donde el código y la infraestructura son generados por IA con provenance y firmas digitales.

## 6. RESULTADOS DEL ESTUDIANTE

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Cuestiones fundamentales (8 horas)</b>	
<b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"><li>● Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial.</li><li>● ¿Qué es comportamiento inteligente?<ul style="list-style-type: none"><li>– El Test de Turing</li><li>– Razonamiento Racional versus No Racional</li></ul></li><li>● Características del Problema:<ul style="list-style-type: none"><li>– Observable completamente versus observable parcialmente</li><li>– Individual versus multi-agente</li><li>– Determinístico versus estocástico</li><li>– Estático versus dinámico</li><li>– Discreto versus continuo</li></ul></li><li>● Naturaleza de agentes:<ul style="list-style-type: none"><li>– Autónomo versus semi-autónomo</li><li>– Reflexivo, basado en objetivos, y basado en utilidad</li><li>– La importancia en percepción e interacciones con el entorno</li></ul></li><li>● Cuestiones filosóficas y éticas.</li></ul>	
<b>Lecturas :</b> [NC24], [Cha24], [Tea25]	

<b>Unidad 2: Cuestiones fundamentales (10 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	
<ul style="list-style-type: none"> <li>● Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial.</li> <li>● ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> <li>– El Test de Turing</li> <li>– Razonamiento Racional versus No Racional</li> </ul> </li> <li>● Características del Problema: <ul style="list-style-type: none"> <li>– Observable completamente versus observable parcialmente</li> <li>– Individual versus multi-agente</li> <li>– Determinístico versus estocástico</li> <li>– Estático versus dinámico</li> <li>– Discreto versus continuo</li> </ul> </li> <li>● Naturaleza de agentes: <ul style="list-style-type: none"> <li>– Autónomo versus semi-autónomo</li> <li>– Reflexivo, basado en objetivos, y basado en utilidad</li> <li>– La importancia en percepción e interacciones con el entorno</li> </ul> </li> <li>● Cuestiones filosóficas y éticas.</li> </ul>	

**Lecturas :** [Bom+21], [Wei+21], [Tea24]

**Unidad 3: Programación de Sistemas Interactivos (12 horas)****Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Patrones de arquitectura de software. Ej Modelo Vista Controlador, Objetos de comando, online, offline.</li><li>• Patrones de diseño de Interacción: jerarquía visual, distancia navegacional.</li><li>• Manejo de eventos e interacción de usuario.</li><li>• Manejo de geometría.</li><li>• Elección de estilos de interacción y técnicas de interacción.</li><li>• Presentación de información: navegación, representación, manipulación.</li><li>• Técnicas de animación de interfaz (ej. grafo de escena)</li><li>• Clases Widget y bibliotecas.</li><li>• Bibliotecas modernas de GUI (ej. iOS, Android, JavaFX) constructores de GUI y entornos de programación UI.</li><li>• Especificación declarativa de Interfaz: Hojas de Estilo y DOMs.</li><li>• Aplicaciones dirigidas a datos ( Páginas web respaldadas por base de datos)</li><li>• Diseño multiplataforma.</li><li>• Diseño para dispositivos con restricción de recursos (ej. dispositivos pequeños, móviles)</li></ul>	

**Lecturas :** [Bey+18], [KSF23], [Eng24]

<b>Unidad 4: Procesos de Software (14 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno.</li> <li>• Introducción a modelos del proceso de software (e.g., cascada, incremental, agil): <ul style="list-style-type: none"> <li>– Actividades con ciclos de vida de software.</li> </ul> </li> <li>• Programación a gran escala versus programación individual.</li> <li>• Evaluación de modelos de proceso de software.</li> <li>• Conceptos de calidad de software.</li> <li>• Mejoramiento de procesos.</li> <li>• Modelos de madurez de procesos de software.</li> <li>• Mediciones del proceso de software.</li> </ul>	
<b>Lecturas :</b> [KSF23], [Mic24], [Goo24]	

<b>Unidad 5: Diseño y Testing centrados en el usuario (12 horas)</b> <b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Enfoque y características del proceso de diseño.</li> <li>• Requerimientos de funcionalidad y usabilidad.</li> <li>• Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual.</li> <li>• Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas.</li> <li>• Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página.</li> <li>• Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar.</li> <li>• Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación.</li> <li>• Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización.</li> <li>• Reportar los resultados de las evaluaciones.</li> <li>• Internacionalización, diseño para usuarios de otras culturas, intercultural.</li> </ul>	
<b>Lecturas :</b> [Goo24], [Met24], [Ama24]	

<b>Unidad 6: Fundamentos y Conceptos en Seguridad (8 horas)</b>	
<b>Resultados esperados: 2,5,6,AG-C03,AG-C07,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• CIA (Confidencialidad, Integridad, Disponibilidad)</li> <li>• Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque .</li> <li>• Autenticación y autorización, control de acceso (vs. obligatoria discrecional)</li> <li>• Concepto de la confianza y la honradez .</li> <li>• Ética (revelación responsable)</li> </ul>	
<b>Lecturas :</b> [ST24], [Fou24], [Goo24]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Bey+18] Betsy Beyer et al. *The Site Reliability Workbook: Practical Ways to Implement SRE*. O'Reilly Media, 2018.
- [Bom+21] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint arXiv:2108.07258* (2021).
- [Wei+21] Laura Weidinger et al. "Ethical and social risks of harm from language models". In: *arXiv preprint arXiv:2112.04359* (2021).
- [KSF23] Gene Kim, Steven Spear, and Nicole Forsgren. *The Platform Engineering Playbook: Building Internal Developer Platforms*. IT Revolution Press, 2023.
- [Ama24] Amazon AWS AI. *Building Reliable AI Systems on AWS*. Amazon Press, 2024.
- [Cha24] Harrison Chase. *LangChain for LLM Application Development: From Prompt Engineering to Agent Orchestration*. O'Reilly Media, 2024.
- [Eng24] Spotify Engineering. *Backstage: Building an Internal Developer Portal*. O'Reilly Media, 2024.
- [Fou24] Linux Foundation. *SLSA Framework v1.0: Supply-chain Levels for Software Artifacts*. Linux Foundation, 2024.
- [Goo24] Google AI Research. *Evaluating AI-Generated Code: Benchmarks and Methodologies*. Google Press, 2024.
- [Met24] Meta AI Research. *Quality Assurance for AI-Native Systems*. Meta Press, 2024.
- [Mic24] Microsoft Research. *AI Software Development: From Copilot to Autonomous Agents*. Microsoft Press, 2024.
- [NC24] Andrew Ng and Harrison Chase. *AI-Native Software Engineering: Building Systems with Autonomous Agents*. DeepLearning.AI Press, 2024.
- [ST24] National Institute of Standards and Technology. *AI Risk Management Framework and Secure AI Supply Chain*. NIST, 2024.
- [Tea24] OpenAI Research Team. *GPT Engineering: Best Practices for Production Systems*. OpenAI, 2024.
- [Tea25] Anthropic Research Team. *Building AI-Native Applications: Principles and Practices*. Anthropic, 2025.

## 1. CURSO

CS393-CS2023. Sistemas de Infomación (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS393-CS2023. Sistemas de Infomación
<b>2.2 Semestre</b>	:	8 <sup>vo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS291. Ingeniería de Software I. (5 <sup>to</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Analizar técnicas para la correcta implementación de Sistemas de Información escalables, robustos, confiables y eficientes en las organizaciones.

## 5. OBJETIVOS

- Implementar de forma correcta (escalables, robustos, confiables y eficientes) Sistemas de Información en las organizaciones.

## 6. RESULTADOS DEL ESTUDIANTE

- 2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

- 5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Usage)

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

- 1)** Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Introducción (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Introducción a la gestión de la información</li> <li>• Software para gestión de información.</li> <li>• Tecnología para gestión de información.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar correctamente la tecnología para la gestión de la información [Evaluar]</li> </ul>

**Lecturas :** [Som17], [PM15], [LL17]

<b>Unidad 2: Estrategia (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Estrategia para gestión de información</li> <li>• Estrategia para gestión conocimiento</li> <li>• Estrategia para sistema de información.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar y evaluar correctamente estrategias de gestión [Evaluar]</li> </ul>

**Lecturas :** [Som17], [PM15]

<b>Unidad 3: Implementación (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Gestión de desarrollo de sistemas de información.</li> <li>• Gestión del cambio</li> <li>• Arquitectura de Información</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar y evaluar correctamente estrategias de implementación [Evaluar]</li> </ul>

**Lecturas :** [Som17], [PM15]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.
- [LL17] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. 15th. Pearson, Mar. 2017.
- [Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

## **Capítulo 9**

### **Noveno Semestre**

## 1. CURSO

CS370-CS2023. Big Data (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS370-CS2023. Big Data
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS3P1. Computación Paralela y Distribuida. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

En la era digital actual, el procesamiento de grandes volúmenes de datos (terabytes, petabytes y exabytes) es fundamental. Este curso cubre tecnologías modernas como Spark, Hadoop y procesamiento de grafos a escala, preparando a los estudiantes para diseñar soluciones escalables en entornos distribuidos y en la nube.

## 5. OBJETIVOS

- Diseñar aplicaciones distribuidas para procesamiento masivo de datos
- Evaluar arquitecturas Big Data según requisitos de escalabilidad y latencia
- Implementar soluciones con frameworks modernos (Spark, Flink, Hadoop)
- Analizar casos de estudio reales en industrias tecnológicas

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Fundamentos de Big Data (10 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Arquitecturas Lambda y Kappa</li> <li>• Sistemas distribuidos modernos</li> <li>• Almacenamiento en la nube (S3, GCS)</li> <li>• Introducción a Spark y Hadoop</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los desafíos del procesamiento distribuido [Familiarizarse]</li> <li>• Comparar arquitecturas batch vs streaming [Usar]</li> <li>• Configurar entornos cloud para Big Data [Usar]</li> </ul>
<b>Lecturas :</b> [Whi21], [Dam+23]	
<b>Unidad 2: Spark y Flink (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Arquitectura de Spark 3.0+</li> <li>• Spark SQL y DataFrames</li> <li>• Procesamiento streaming con Structured Streaming</li> <li>• Flink para tiempo real</li> <li>• Optimización de consultas</li> </ul>	<ul style="list-style-type: none"> <li>• Desarrollar pipelines ETL con PySpark [Usar]</li> <li>• Implementar aplicaciones streaming [Evaluar]</li> <li>• Optimizar jobs con particionamiento [Evaluar]</li> </ul>
<b>Lecturas :</b> [Dam+23], [Per23]	
<b>Unidad 3: Almacenamiento y Grafos (10 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Bases de datos distribuidas (Cassandra, HBase)</li> <li>• Procesamiento de grafos con GraphX</li> <li>• Almacenes delta (Delta Lake)</li> <li>• Data Lakes vs Data Warehouses</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar esquemas para NoSQL [Usar]</li> <li>• Implementar algoritmos en grafos [Evaluar]</li> <li>• Gestionar metadatos en data lakes [Usar]</li> </ul>
<b>Lecturas :</b> [RWE15], [Zah+16]	
<b>Unidad 4: Aplicaciones Avanzadas (13 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Machine Learning con Spark MLlib</li> <li>• Pipelines CI/CD para Big Data</li> <li>• Seguridad y gobernanza</li> <li>• Casos: Netflix, Uber, TikTok</li> </ul>	<ul style="list-style-type: none"> <li>• Desplegar modelos ML a escala [Evaluar]</li> <li>• Implementar políticas de seguridad [Usar]</li> <li>• Analizar arquitecturas industriales [Evaluar]</li> </ul>
<b>Lecturas :</b> [Kle17], [MW15]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

### 10. BIBLIOGRAFÍA BÁSICA

- [MW15] Nathan Marz and James Warren. *Big Data: Principles and Best Practices*. Lambda architecture deep dive. Manning, 2015.
- [RWE15] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. Graph processing fundamentals. O'Reilly, 2015.
- [Zah+16] Matei Zaharia et al. "Apache Spark: A unified engine for big data processing". In: *Communications of the ACM* 59.11 (2016), pp. 56–65.
- [Kle17] Martin Kleppmann. *Designing Data-Intensive Applications*. Architecture patterns for Big Data. O'Reilly, 2017.
- [Whi21] Tom White. *Hadoop: The Definitive Guide*. 5th. Covers Hadoop 3 and YARN. O'Reilly, 2021.
- [Dam+23] Jules Damji et al. *Learning Spark: Lightning-Fast Data Analytics*. 2nd. Spark 3.5 with Python/SQL examples. O'Reilly, 2023.
- [Per23] Jean-Georges Perrin. *Spark in Action*. 2nd. Practical Spark applications. Manning, 2023.

## 1. CURSO

CS400FCCS. Prácticas Pre-profesionales (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS400FCCS. Prácticas Pre-profesionales
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	1 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	Ninguno

## 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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**5)** Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Assessment)

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

**3)** Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

**4)** Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

**h)** Incorporarse a un proceso de aprendizaje profesional continuo. (Assessment)

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

**i)** Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Assessment)

**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)

## 7. TEMAS

<b>Unidad 1: Evaluación de Desempeño en Entorno Laboral (48 horas)</b>	
<b>Resultados esperados: 2,3,4,5,AG-C01,AG-C02,AG-C03,AG-C04,AG-C05,AG-C06,AG-C09,h,i</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Integración del estudiante en un equipo de trabajo profesional.</li><li>• Aplicación de conocimientos técnicos en proyectos reales.</li><li>• Evaluación de competencias éticas y profesionales en el ámbito laboral.</li><li>• Presentación de informes y retroalimentación por parte de la empresa y la universidad.</li></ul>	<ul style="list-style-type: none"><li>• Demostrar habilidades técnicas en un entorno laboral real [Evaluar].</li><li>• Aplicar principios éticos y profesionales en sus actividades [Evaluar].</li><li>• Trabajar efectivamente en equipo y comunicarse de manera profesional [Evaluar].</li></ul>
<b>Lecturas :</b> [Edu22], [Soc21]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Soc21] IEEE Computer Society. *Prácticas Profesionales en Computación: Ética y Gestión*. Manual de buenas prácticas para entornos laborales en TI. IEEE Press, 2021.
- [Edu22] Comité de Educación de la ACM. *Guía para Prácticas Pre-profesionales Exitosas en Ciencia de la Computación*. Tech. rep. Documento oficial con estándares para prácticas en computación. Association for Computing Machinery, 2022. URL: <https://www.acm.org/education>.

## 1. CURSO

CS403. Proyecto de tesis 2 (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS403. Proyecto de tesis 2
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS402. Proyecto de tesis 1. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno desarrolle su propuesta y sus experimentos en base al estado del arte ya levantado previamente.

## 5. OBJETIVOS

- Que el alumno esté en la capacidad de presentar formalmente su proyecto de tesis con el marco teórico y levantamiento bibliográfico completo.
- Que el alumno desarrolle su propuesta de investigación.
- Los entregables de este curso son:

**Avance parcial:** Avance parcial de su propuesta de investigación del plan de tesis incluyendo motivación, contexto, definición del problema, objetivos, cronograma de actividades hasta el proyecto final de tesis, el estado del arte del tema abordado y avances parciales de su propuesta de investigación incluyendo como propone usar/crear/mejorar/aplicar algoritmos nuevos o ya existentes.

**Final:** Plan de tesis en una etapa avanzada, avance de la tesis incluyendo los capítulos de propuesta de tesis y experimentos preliminares.

## 6. RESULTADOS DEL ESTUDIANTE

- 1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

- 2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

- 4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

## 7. TEMAS

<b>Unidad 1: Proyecto de Tesis (30 horas)</b>	
<b>Resultados esperados:</b> 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Proyecto de Tesis.</li></ul>	<ul style="list-style-type: none"><li>• Descripción del formato utilizado por la Universidad para el plan de tesis [Evaluar]</li><li>• Concluir el plan del proyecto de tesis [Evaluar]</li><li>• Presentar el estado del arte del tema de tesis (50%) [Evaluar]</li></ul>
<b>Lecturas :</b> [IEE08], [ACM08], [Cit08]	

<b>Unidad 2: Avance de Tesis (30 horas)</b>	
<b>Resultados esperados:</b>	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Avance de Tesis.</li></ul>	<ul style="list-style-type: none"><li>• Descripción del formato utilizado por la Universidad para la tesis [Evaluar]</li><li>• Concluir el capítulo del Marco Teórico de la Tesis [Evaluar]</li><li>• Concluir el capítulo de Trabajos Relacionados (35%) [Evaluar]</li><li>• Planear, desarrollar y presentar resultados (formales o estadísticos) de experimentos orientados a su tema de tesis (35%) [Evaluar]</li></ul>
<b>Lecturas :</b> [IEE08], [ACM08], [Cit08]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

---

## 10. BIBLIOGRAFÍA BÁSICA

- [ACM08] ACM. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.
- [Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.
- [IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.

## 1. CURSO

AI365. Modelos Generativos Avanzados en IA (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	AI365. Modelos Generativos Avanzados en IA
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI264. Aprendizaje Profundo. (7 <sup>mo</sup> 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

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

**4)** Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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

<b>Unidad 1: Fundamentos Generativos (12 horas)</b>	
<b>Resultados esperados: 6,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Teoría de GANs (minimax, Wasserstein)</li> <li>• Autoregressive models (PixelCNN, Transformers)</li> <li>• Métricas de evaluación (FID, Inception Score)</li> </ul>	<ul style="list-style-type: none"> <li>• Entrenar GANs básicas [Usar]</li> <li>• Analizar modos de falla en generación [Evaluar]</li> </ul>

Lecturas : [Goo+14]

<b>Unidad 2: Modelos de Difusión (24 horas)</b>	
<b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Procesos de difusión (forward/reverse)</li> <li>• Stable Diffusion y arquitecturas latentes</li> <li>• Técnicas de control (ControlNet, LoRA)</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar samplers de difusión [Usar]</li> <li>• Fine-tuning para dominios específicos [Evaluar]</li> </ul>

Lecturas : [SSE23], [Rom+22]

<b>Unidad 3: Ética en Medios Sintéticos (8 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Deepfakes y desinformación</li> <li>• Derechos de autor en contenido generado</li> <li>• Marcado de autenticidad (C2PA, watermarking)</li> </ul>	<ul style="list-style-type: none"> <li>• Detectar contenido sintético [Familiarizarse]</li> <li>• Implementar sistemas de trazabilidad [Usar]</li> </ul>

Lecturas : [SSE23]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Goo+14] Ian Goodfellow et al. “Generative Adversarial Networks”. In: *NeurIPS*. 2014.
- [Rom+22] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *CVPR* (2022).
- [SSE23] Jascha Sohl-Dickstein, Yang Song, and Stefano Ermon. *Diffusion Models: A Comprehensive Practical Guide*. AI Press, 2023.

## 1. CURSO

CB309. Bioinformática (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CB309. Bioinformática
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	BBI101. Biología. (7 <sup>mo</sup> Sem)

## 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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (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

<b>Unidad 1: Introducción a la Biología Molecular (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Revisión de la química orgánica: moléculas y macromoléculas, azúcares, ácidos nucleicos, nuclótidos, ARN, ADN, proteínas, aminoácidos y niveles de estructura en las proteínas.</li> <li>El dogma de la vida: del ADN a las proteínas, transcripción, traducción, síntesis de proteínas</li> <li>Estudio del genoma: Mapas y secuencias, técnicas específicas</li> </ul>	<ul style="list-style-type: none"> <li>Lograr un conocimiento general de los tópicos más importantes en Biología Molecular. [Familiarizarse]</li> <li>Entender que los problemas biológicos son un desafío al mundo computacional. [Evaluar]</li> </ul>
<b>Lecturas :</b> [CB00], [SM97]	

<b>Unidad 2: Comparación de Secuencias (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Secuencias de nucléotidos y secuencias de aminoácidos.</li> <li>Alineamiento de secuencias, el problema de alineamiento por pares, búsqueda exhaustiva, Programación dinámica, alineamiento global, alineamiento local, penalización por gaps</li> <li>Comparación de múltiples secuencias: suma de pares, análisis de complejidad por programación dinámica, heurísticas de alineamiento, algoritmo estrella, algoritmos de alineamiento progresivo.</li> </ul>	<ul style="list-style-type: none"> <li>Entender y solucionar el problema de alineamiento de un par de secuencias. [Usar]</li> <li>Comprender y solucionar el problema de alineamiento de múltiples secuencias. [Usar]</li> <li>Conocer los diversos algoritmos de alineamiento de secuencias existentes en la literatura. [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [CB00], [SM97], [Pev00]	

<b>Unidad 3: Árboles Filogenéticos (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>Filogenia: Introducción y relaciones filogenéticas.</li> <li>Árboles Filogenéticos: definición, tipo de árboles, problema de búsqueda y reconstrucción de árboles</li> <li>Métodos de Reconstrucción: métodos por parsimonia, métodos por distancia, métodos por máxima verosimilitud, confianza de los árboles reconstruidos</li> </ul>	<ul style="list-style-type: none"> <li>Comprender el concepto de filogenia, árboles filogenéticos y la diferencia metodológica entre biología y biología molecular. [Familiarizarse]</li> <li>Comprender el problema de reconstrucción de árboles filogenéticos, conocer y aplicar los principales algoritmos para reconstrucción de árboles filogenéticos. [Evaluar]</li> </ul>
<b>Lecturas :</b> [CB00], [SM97], [Pev00]	

<b>Unidad 4: Ensamblaje de Secuencias de ADN (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fundamento biológico: caso ideal, dificultades, métodos alternativos para secuenciamiento de ADN</li> <li>• Modelos formales de ensamblaje: <i>Shortest Common Superstring, Reconstruction, Multicontig</i></li> <li>• Algoritmos para ensamblaje de secuencias: representación de overlaps, caminos para crear <i>superstrings</i>, algoritmo voraz, grafos acíclicos.</li> <li>• Heurísticas para ensamblaje: búsqueda de sobreposiciones, ordenación de fragmentos, alineamientos y consenso.</li> </ul>	
<b>Lecturas :</b> [SM97], [Alu06]	
<b>Unidad 5: Estructuras secundarias y terciarias (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Estructuras moleculares: primaria, secundaria, terciaria, cuaternaria.</li> <li>• Predicción de estructuras secundarias de ARN: modelo formal, energía de pares, estructuras con bases independientes, solución con Programación Dinámica, estructuras con bucles.</li> <li>• <i>Protein folding</i>: Estructuras en proteínas, problema de <i>protein folding</i>.</li> <li>• <i>Protein Threading</i>: Definiciones, Algoritmo <i>Branch &amp; Bound</i>, <i>Branch &amp; Bound</i> para <i>protein threading</i>.</li> <li>• <i>Structural Alignment</i>: definiciones, algoritmo DALI</li> </ul>	
<b>Lecturas :</b> [SM97], [CB00], [Alu06]	
<b>Unidad 6: Modelos Probabilísticos en Biología Molecular (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Probabilidad: Variables aleatorias, Cadenas de Markov, Algoritmo de Metropoli-Hasting, Campos Aleatorios de Markov y Muestreador de Gibbs, Máxima Verosimilitud.</li> <li>• 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.</li> <li>• Filogenia Probabilística: Modelos probabilísticos de evolución, verosimilitud de alineamientos, verosimilitud para inferencia, comparación de métodos probabilísticos y no probabilísticos</li> </ul>	
<b>Lecturas :</b> [Dur+98], [CB00], [Alu06], [Kro+94]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Kro+94] Anders Krogh et al. "Hidden Markov Models in Computational Biology, Applications to Protein Modeling". In: *J Molecular Biology* 235 (1994), pp. 1501–1531.
- [SM97] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology*. Boston: PWS Publishing Company, 1997, pp. I–XIII, 1–296.
- [Dur+98] R. Durbin et al. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998, p. 357.
- [CB00] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. 279 pages. John Wiley & Sons Ltd., 2000.
- [Pev00] Pavel A. Pevzner. *Computational Molecular Biology: an Algorithmic Approach*. Cambridge, Massachusetts: The MIT Press, 2000.
- [Alu06] Srinivas Aluru, ed. *Handbook of Computational Molecular Biology*. Computer and Information Science Series. Boca Raton, FL: Chapman & Hall, CRC, 2006.

## 1. CURSO

CS351-CS2023. Tópicos en Computación Gráfica (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS351-CS2023. Tópicos en Computación Gráfica
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS251. Computación Gráfica. (7 <sup>mo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Curso avanzado que cubre técnicas modernas de renderizado (ray tracing en tiempo real, path tracing), shaders avanzados, y computación GPU. Incluye aplicaciones en videojuegos, realidad virtual/aumentada, y visualización científica con enfoque en DirectX 12 Ultimate y Vulkan Ray Tracing.

## 5. OBJETIVOS

- Implementar algoritmos de renderizado avanzado usando APIs modernas (Vulkan, DirectX 12)
- Diseñar shaders complejos (ray tracing, compute shaders)
- Optimizar pipelines gráficos para hardware actual (RTX, AMD RDNA3)
- Analizar técnicas de inteligencia artificial aplicadas a gráficos (DLSS, Neural Rendering)

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Renderizado Avanzado (20 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Ray Tracing en tiempo real (DXR, Vulkan RT)</li> <li>• Path Tracing diferencial</li> <li>• Denoising con IA (DLSS, XeSS)</li> <li>• Pipeline State Objects (PSOs)</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar efectos de iluminación global con RTX [Usar]</li> <li>• Integrar upscalers neuronales [Evaluar]</li> <li>• Optimizar pipelines para multi-GPU [Evaluar]</li> </ul>
<b>Lecturas :</b> [MSW21], [PM23]	

<b>Unidad 2: Computación GPU (20 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Arquitecturas GPU modernas (Ada Lovelace, RDNA3)</li> <li>• Compute shaders (Wave Intrinsics)</li> <li>• Culling avanzado (Mesh Shaders)</li> <li>• Renderizado sin rasterización (Nanite, Lumen)</li> </ul>	<ul style="list-style-type: none"> <li>• Programar kernels de computación paralela [Usar]</li> <li>• Diseñar sistemas de culling geométrico [Evaluar]</li> <li>• Perfilar código GPU con Nsight/RGP [Usar]</li> </ul>
<b>Lecturas :</b> [KG23], [LN22]	

<b>Unidad 3: Futuro de Gráficos (20 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Renderizado neuronal (NeRFs, Gaussian Splatting)</li> <li>• Realidad extendida (OpenXR)</li> <li>• Shaders probabilísticos</li> <li>• Hardware futuro (Chiplets, Photonics)</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar modelos NeRF básicos [Usar]</li> <li>• Evaluar técnicas para XR [Evaluar]</li> <li>• Discutir límites físicos de renderizado [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [HA23], [YR23]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[MSW21] Adam Marrs, Peter Shirley, and Ingo Wald. *Ray Tracing Gems II*. Advanced RT techniques with DLSS and hybrid rendering. Apress, 2021.

- 
- [LN22] Timothy Lottes and NVIDIA. “RTX Accelerated Wavelet Noise”. In: *SIGGRAPH Advances in Real-Time Rendering*. Procedural texturing with RT cores. 2022.
- [HA23] Eric Haines and Tomas Akenine-Möller. “The State of Ray Tracing in 2023”. In: *Journal of Computer Graphics Techniques* 11.2 (2023).
- [KG23] Brian Karis and Epic Games. “Nanite and Lumen Technical Deep Dive”. In: *Unreal Engine Whitepapers* (2023). Virtualized geometry and global illumination in UE5.
- [PM23] Nuno Pereira and Microsoft. *DirectX Raytracing Documentation*. Tech. rep. Official DXR programming guide. Microsoft, 2023.
- [YR23] Lingjie Yan and Google Research. “Neural Radiance Fields for Real-Time View Synthesis”. In: *ACM Transactions on Graphics* 42.4 (2023). Real-time NeRF optimizations.

## 1. CURSO

CS353-CS2023. Computación Cuántica (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS353-CS2023. Computación Cuántica
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS3I1. Seguridad en Computación. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso introduce los principios fundamentales de la computación cuántica, incluyendo qubits, superposición, entrelazamiento, algoritmos cuánticos y hardware. Los estudiantes contrastarán modelos clásicos y cuánticos, explorando aplicaciones en criptografía, optimización, simulación física y machine learning cuántico, utilizando frameworks como Qiskit o Cirq. Se incluye un módulo práctico sobre arquitecturas de hardware (superconductores, trampas de iones) y su impacto en el diseño algorítmico.

## 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 limitaciones de hardware en aplicaciones reales.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**1)** Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Usage)

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

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

## 7. TEMAS

<b>Unidad 1: Fundamentos de Mecánica Cuántica (12 horas)</b> <b>Resultados esperados: 1,AG-C08</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Qubits y representación en la esfera de Bloch</li> <li>• Postulados de la mecánica cuántica</li> <li>• Puertas lógicas cuánticas (Hadamard, CNOT, Toffoli)</li> </ul>	<ul style="list-style-type: none"> <li>• Modelar matemáticamente un qubit [Usar]</li> <li>• Simular operaciones básicas en Qiskit [Evaluar]</li> </ul>
<b>Lecturas :</b> [NC10], [Tea23]	

<b>Unidad 2: Hardware Cuántico (12 horas)</b> <b>Resultados esperados: AG-C07</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Arquitecturas: superconductores (IBM, Google) vs. trampas de iones (IonQ)</li> <li>• Métricas: coherencia, error rates, conectividad qubit</li> <li>• Compilación de circuitos para hardware real</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar ventajas/desventajas de plataformas de hardware [Familiarizarse]</li> <li>• Adaptar circuitos a restricciones de hardware [Usar]</li> </ul>
<b>Lecturas :</b> [MV23], [23]	

<b>Unidad 3: Algoritmos Cuánticos (12 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Algoritmo de Deutsch-Jozsa</li> <li>• Búsqueda de Grover (aplicado a optimización combinatoria)</li> <li>• Factorización de Shor y su impacto en RSA</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar Grover para problemas de satisfacción (SAT) [Usar]</li> </ul>
<b>Lecturas :</b> [Pre18], [Tea23]	

<b>Unidad 4: Aplicaciones en Optimización (8 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (<i>Learning Outcomes</i>)</b>
<ul style="list-style-type: none"> <li>• Quantum Approximate Optimization Algorithm (QAOA)</li> <li>• Aplicaciones en logística y finanzas (Portfolio Optimization)</li> <li>• Benchmarks vs. algoritmos clásicos (Simulated Annealing)</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar un QAOA para un problema de ruteo [Evaluar]</li> </ul>
<b>Lecturas :</b> [QMLBook], [QuantumOptimizationSurvey]	

<b>Unidad 5: Ética y Aplicaciones (4 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Criptografía post-cuántica (NIST PQC Standardización)</li> <li>• Supremacía cuántica: debate técnico/ético</li> <li>• Patentes y geopolítica cuántica</li> </ul>	<ul style="list-style-type: none"> <li>• Proponer estrategias de migración post-cuántica [Familiarizarse]</li> </ul>
<b>Lecturas :</b> [Tea19], [PostQuantumCrypto]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Texto clásico para fundamentos teóricos. Cambridge University Press, 2010.
- [Pre18] John Preskill. “Lecture Notes on Quantum Computation”. 2018. URL: <http://theory.caltech.edu/~preskill/ph219/>.
- [Tea19] Google AI Quantum Team. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574 (2019). URL: <https://www.nature.com/articles/s41586-019-1666-5>.
- [23] *IBM Quantum Hardware Roadmap*. Tech. rep. IBM Research, 2023. URL: <https://research.ibm.com/quantum-hardware>.
- [MV23] Mikko Möttönen and Juha Vartiainen. *Quantum Computing: From Qubits to Quantum Machines*. Springer, 2023.
- [Tea23] IBM Quantum Team. *Qiskit Textbook*. Guía práctica con ejemplos en Qiskit. 2023. URL: <https://qiskit.org/textbook>.

## 1. CURSO

CS369-CS2023. Robótica (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS369-CS2023. Robótica
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI268. Visión Computacional. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso integra principios de inteligencia artificial, control y programación para diseñar soluciones robóticas autónomas, enfatizando el análisis de problemas complejos, el trabajo en equipo y la ética profesional.

## 5. OBJETIVOS

- Diseñar algoritmos robóticos aplicando teoría computacional (ABET6/AG-C12).
- Evaluar impactos éticos y sociales de la robótica (ABET4/AG-C02).
- Funcionar como líder o miembro de un equipo multidisciplinario (ABET5/AG-C03).

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Fundamentos de Robótica (12 horas)</b>	
<b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Cinemática y dinámica de robots.</li> <li>• Modelado con <math>\lambda</math>-cálculo.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicar teoría computacional al diseño robótico [Usar].</li> </ul>
<b>Lecturas :</b> [Sic+09], [Mur00]	

<b>Unidad 2: Ética y Sociedad (12 horas)</b>	
<b>Resultados esperados: 4,AG-C02</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Dilemas éticos en autonomía robótica.</li> <li>• Impacto social de la robótica.</li> </ul>	<ul style="list-style-type: none"> <li>• Analizar responsabilidades profesionales [Evaluar].</li> </ul>
<b>Lecturas :</b> [Mur00], [TBF05]	

<b>Unidad 3: Proyecto en Equipo (24 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Colaboración en desarrollo de prototipos.</li> <li>• Documentación y presentación técnica.</li> </ul>	<ul style="list-style-type: none"> <li>• Liderar equipos en proyectos robóticos [Evaluar].</li> </ul>
<b>Lecturas :</b> [Sic+09], [LaV06]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Mur00] Robin R. Murphy. *Introduction to AI Robotics*. MIT Press, 2000.
- [TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [LaV06] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. URL: <http://planning.cs.uiuc.edu/>.
- [Sic+09] Bruno Siciliano et al. *Robotics: Modelling, Planning and Control*. Springer, 2009. doi: 10.1007/978-1-84628-642-1.

## 1. CURSO

CS392-CS2023. Tópicos en Ingeniería de Software (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS392-CS2023. Tópicos en Ingeniería de Software
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS391. Ingeniería de Software III. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

El desarrollo de software requiere del uso de mejores prácticas de desarrollo, gestión de proyectos de TI, manejo de equipos y uso eficiente y racional de frameworks de aseguramiento de la calidad y de Gobierno de Portfolios, estos elemento son pieza clave y transversal para el éxito del proceso productivo.

Este curso explora el diseño, selección, implementación y gestión de soluciones TI en las Organizaciones. El foco está en las aplicaciones y la infraestructura y su aplicación en el negocio.

## 5. OBJETIVOS

- Entender una variedad de frameworks para el análisis de arquitectura empresarial y la toma de decisiones
- Utilizar técnicas para la evaluación y gestión del riesgo en el portfolio de la empresa
- Evaluar y planificar la integración de tecnologías emergentes
- Entender el papel y el potencial de las TI para apoyar la gestión de procesos empresariales
- Entender los diferentes enfoques para modelar y mejorar los procesos de negocio
- Describir y comprender modelos de aseguramiento de la calidad como marco clave para el éxito de los proyectos de TI.
- Comprender y aplicar el framework de IT Governance como elemento clave para la gestión del portfolio de aplicaciones Empresariales

## 6. RESULTADOS DEL ESTUDIANTE

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

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

---

## 7. TEMAS

## Unidad 1: Diseño de Software (18 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>Principios de diseño del sistema: niveles de abstracción (diseño arquitectónico y el diseño detallado), separación de intereses, ocultamiento de información, de acoplamiento y de cohesión, de reutilización de estructuras estándar.</li> <li>Diseño de paradigmas tales como diseño estructurado (descomposición funcional de arriba hacia abajo), el análisis orientado a objetos y diseño, orientado a eventos de diseño, diseño de nivel de componente, centrado datos estructurada, orientada a aspectos, orientado a la función, orientado al servicio.</li> <li>Modelos estructurales y de comportamiento de los diseños de software.</li> <li>Diseño de patrones.</li> <li>Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.</li> <li>Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).</li> <li>El uso de componentes de diseño: selección de componentes, diseño, adaptación y componentes de ensamblaje, componentes y patrones, componentes y objetos (por ejemplo, construir una GUI usando un standar widget set)</li> <li>Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.</li> <li>Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.</li> <li>Medición y análisis de la calidad de un diseño.</li> <li>Compensaciones entre diferentes aspectos de la calidad.</li> <li>Aplicaciones en frameworks.</li> <li>Middleware: El paradigma de la orientación a objetos con middleware, requerimientos para correr y clasificar objetos, monitores de procesamiento de transacciones y el sistema de flujo de trabajo.</li> <li>Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design). <ul style="list-style-type: none"> <li>– Principio de privilegios mínimos</li> <li>– Principio de falla segura por defecto</li> <li>– Principio de aceptabilidad psicológica</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Usar]</li> <li>Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño [Usar]</li> <li>Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]</li> <li>En el contexto de un paradigma de diseño simple, describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple [Usar]</li> <li>Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]</li> <li>Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde las especificaciones de requisitos [Usar]</li> <li>Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Usar]</li> <li>Para el diseño de un sistema de software simple dentro del contexto de un único paradigma de diseño, describir la arquitectura de software de ese sistema [Usar]</li> <li>Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (<i>3-tier</i>), <i>pipe-and-filter</i>, y cliente-servidor [Usar]</li> <li>Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Usar]</li> <li>Aplicar ejemplos simples de patrones en un diseño de software [Usar]</li> <li>Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Usar]</li> <li>Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]</li> <li>Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Usar]</li> <li>Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar]</li> <li>Discutir y seleccionar la arquitectura de software adecuada para un sistema de software simple para un dado escenario [Usar]</li> </ul>

## Unidad 2: Gestión de Proyectos de Software (14 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● La participación del equipo: <ul style="list-style-type: none"> <li>– Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo</li> <li>– Roles y responsabilidades en un equipo de software</li> <li>– Equipo de resolución de conflictos</li> <li>– Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura)</li> </ul> </li> <li>● Estimación de esfuerzo (a nivel personal)</li> <li>● Riesgo. <ul style="list-style-type: none"> <li>– El papel del riesgo en el ciclo de vida</li> <li>– Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de</li> </ul> </li> <li>● Gestión de equipos: <ul style="list-style-type: none"> <li>– Organización de equipo y la toma de decisiones</li> <li>– Roles de identificación y asignación</li> <li>– Individual y el desempeño del equipo de evaluación</li> </ul> </li> <li>● Gestión de proyectos: <ul style="list-style-type: none"> <li>– Programación y seguimiento de elementos</li> <li>– Herramientas de gestión de proyectos</li> <li>– Análisis de Costo/Beneficio</li> </ul> </li> <li>● Software de medición y técnicas de estimación.</li> <li>● Aseguramiento de la calidad del software y el rol de las mediciones.</li> <li>● Riesgo. <ul style="list-style-type: none"> <li>– El papel del riesgo en el ciclo de vida</li> <li>– Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de</li> </ul> </li> <li>● En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.</li> </ul>	<ul style="list-style-type: none"> <li>● Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Usar]</li> <li>● Crear y seguir un programa para una reunión del equipo [Usar]</li> <li>● Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar]</li> <li>● Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar]</li> <li>● Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar]</li> <li>● Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar]</li> <li>● Listar varios ejemplos de los riesgos del software [Usar]</li> <li>● Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar]</li> <li>● Describir las diferentes categorías de riesgo en los sistemas de software [Usar]</li> <li>● Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar]</li> <li>● Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Usar]</li> <li>● Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar]</li> <li>● Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar]</li> <li>● Usando un software particular procesar, describir los aspectos de un proyecto que encesita ser planeado y monitoreado, (ejemplo, estimar el tamaño y esfuerzo, un horario, reasignación de recursos, control de configuración, gestión de cambios, identificación de riesgos en un proyecto y gestión) [Usar]</li> <li>● Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar]</li> <li>● Comparar las técnicas simples de tamaño de software y estimación de costos [Usar]</li> <li>● Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar]</li> <li>● Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar]</li> </ul>

<b>Unidad 3: ITIL (14 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Administración del servicio como práctica.</li> <li>• Ciclo de vida del servicio.</li> <li>• Definiciones y conceptos genéricos.</li> <li>• Modelos y principios claves.</li> <li>• Procesos.</li> <li>• Tecnología y arquitectura.</li> <li>• Competencia y entrenamiento.</li> </ul>	<ul style="list-style-type: none"> <li>• Utilizar y aplicar correctamente ITIL en el proceso de software. [Usar]</li> </ul>
<b>Lecturas :</b> [Som17], [PM15]	

<b>Unidad 4: COBIT (14 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Fundamentos e Introducción.</li> <li>• Frameworks de Control y IT Governance.</li> </ul>	<ul style="list-style-type: none"> <li>• Utilizar y aplicar correctamente COBIT en el proceso de software. [Usar]</li> </ul>
<b>Lecturas :</b> [Som17], [PM15]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

## 1. CURSO

CS3P3-CS2023. Internet de las Cosas (Electivo)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS3P3-CS2023. Internet de las Cosas
<b>2.2 Semestre</b>	:	9 <sup>no</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Electivo
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS3P1. Computación Paralela y Distribuida. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

La última década ha traído un crecimiento explosivo en computación con multiprocesadores, incluyendo los procesadores de varios núcleos y centros de datos distribuidos. Como resultado, la computación paralela y distribuida se ha convertido de ser un tema ampliamente electivo para ser uno de los principales componentes en la malla estudios en ciencia de la computación de pregrado. Tanto la computación paralela como la distribuida implica la ejecución simultánea de múltiples procesos en diferentes dispositivos que cambian de posición.

## 5. OBJETIVOS

- Que el alumno sea capaz de crear aplicaciones paralelas de mediana complejidad aprovechando eficientemente distintos dispositivos móviles.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

**Unidad 1: Fundamentos de paralelismo (18 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Procesamiento Simultáneo Múltiple.</li> <li>• Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos)</li> <li>• Paralelismo, comunicación, y coordinación: <ul style="list-style-type: none"> <li>– Paralelismo, comunicación, y coordinación</li> <li>– Necesidad de Sincronización</li> </ul> </li> <li>• Errores de Programación ausentes en programación secuencial: <ul style="list-style-type: none"> <li>– Tipos de Datos ( lectura/escritura simultánea o escritura/escritura compartida)</li> <li>– Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado)</li> <li>– Falta de vida/progreso (deadlock, starvation)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse]</li> <li>• Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse]</li> <li>• Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]</li> </ul>

**Lecturas :** [Pac11], [Mat14], [Qui03]

<b>Unidad 2: Arquitecturas paralelas (12 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Procesadores multinúcleo.</li> <li>• Memoria compartida vs memoria distribuida.</li> <li>• Multiprocesamiento simétrico.</li> <li>• SIMD, procesamiento de vectores.</li> <li>• GPU, coprocesamiento.</li> <li>• Taxonomía de Flynn.</li> <li>• Soporte a nivel de instrucciones para programación paralela. <ul style="list-style-type: none"> <li>– Instrucciones atómicas como Compare/Set (Comparar / Establecer)</li> </ul> </li> <li>• Problemas de Memoria: <ul style="list-style-type: none"> <li>– Caches multiprocesador y coherencia de cache</li> <li>– Acceso a Memoria no uniforme (NUMA)</li> </ul> </li> <li>• Topologías. <ul style="list-style-type: none"> <li>– Interconexiones</li> <li>– Clusters</li> <li>– Compartir recursos (p.e., buses e interconexiones)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar]</li> <li>• Describir la arquitectura SMP y observar sus principales características [Evaluar]</li> <li>• Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar]</li> <li>• Describir las ventajas y limitaciones de GPUs vs CPUs [Usar]</li> <li>• Explicar las características de cada clasificación en la taxonomía de Flynn [Usar]</li> <li>• Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]</li> <li>• Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]</li> </ul>

**Lecturas :** [Pac11], [KH13], [SK10]

<b>Unidad 3: Descomposición en paralelo (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Necesidad de Comunicación y coordinación/sincronización.</li> <li>• Independencia y Particionamiento.</li> <li>• Conocimiento Básico del Concepto de Descomposición Paralela.</li> <li>• Decomposición basada en tareas: <ul style="list-style-type: none"> <li>– Implementación de estrategias como hebras</li> </ul> </li> <li>• Descomposición de Información Paralela <ul style="list-style-type: none"> <li>– Estrategias como SIMD y MapReduce</li> </ul> </li> <li>• Actores y Procesos Reactivos (solicitud de gestores)</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar]</li> <li>• Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse]</li> <li>• Escribir un algoritmo paralelo correcto y escalable [Usar]</li> <li>• Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar]</li> <li>• Paralelizar un algoritmo mediante la aplicación de descomposición datos en paralelo [Usar]</li> <li>• Escribir un programa usando actores y/o procesos reactivos [Usar]</li> </ul>

**Lecturas :** [Pac11], [Mat14], [Qui03]

## Unidad 4: Comunicación y coordinación (18 horas)

### Resultados esperados:

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>● Memoria Compartida.</li> <li>● La consistencia, y su papel en los lenguajes de programación garantizan las condiciones para los programas de carrera libre.</li> <li>● Pasos de Mensaje: <ul style="list-style-type: none"> <li>– Mensajes Punto a Punto versus multicast (o basados en eventos)</li> <li>– Estilos para enviar y recibir mensajes Blocking vs non-blocking</li> <li>– Buffering de mensajes</li> </ul> </li> <li>● Atomicidad: <ul style="list-style-type: none"> <li>– Especificar y probar atomicidad y requerimientos de seguridad</li> <li>– Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas</li> <li>– Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas <ul style="list-style-type: none"> <li>* Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención)</li> <li>– Composición <ul style="list-style-type: none"> <li>* Componiendo acciones atómicas granulares más grandes usando sincronización</li> <li>* Transacciones, incluyendo enfoques optimistas y conservadores</li> </ul> </li> </ul> </li> <li>● Consensos: <ul style="list-style-type: none"> <li>– (Cíclicos) bareras, contadores y estructuras relacionadas</li> </ul> </li> <li>● Acciones condicionales: <ul style="list-style-type: none"> <li>– Espera condicional (p.e., empleando variables de condición)</li> </ul> </li> </ul> </li></ul>	<ul style="list-style-type: none"> <li>● Usar exclusión mútua para evitar una condición de carrera [Usar]</li> <li>● Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse]</li> <li>● Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar]</li> <li>● Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse]</li> <li>● Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar]</li> <li>● Dar un ejemplo de una ordenación de accesos entre actividades concurrentes (por ejemplo, un programa con condición de carrera) que no son secuencialmente consistentes [Familiarizarse]</li> <li>● Usar semáforos o variables de condición para bloquear hebras hasta una necesaria precondición de mantenga [Usar]</li> </ul>

Lecturas : [Pac11], [Mat14], [Qui03]

**Unidad 5: Análisis y programación de algoritmos paralelos (18 horas)**

**Resultados esperados:**

Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"> <li>• Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl.</li> <li>• Aceleración y escalabilidad.</li> <li>• Naturalmente (vergonzosamente) algoritmos paralelos.</li> <li>• Patrones Algorítmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) <ul style="list-style-type: none"> <li>– Algoritmos específicos (p.e., MergeSort paralelo)</li> </ul> </li> <li>• Algoritmos de grafos paralelos (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela)</li> <li>• Cálculos de matriz paralelas.</li> <li>• Productor-consumidor y algoritmos paralelos segmentados.</li> <li>• Ejemplos de algoritmos paralelos no-escalables.</li> </ul>	<ul style="list-style-type: none"> <li>• Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse]</li> <li>• Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar]</li> <li>• Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse]</li> <li>• Identificar tareas independientes en un programa que debe ser paralelizado [Usar]</li> <li>• Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse]</li> <li>• Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su análogo secuencial [Usar]</li> <li>• Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) vía operaciones <i>map</i> y <i>reduce</i> [Usar]</li> <li>• Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar]</li> <li>• Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar]</li> <li>• Implementar un algoritmo de matriz paralela [Usar]</li> <li>• Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]</li> </ul>

**Lecturas :** [Mat14], [Qui03]

<b>Unidad 6: Desempeño en paralelo (18 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Equilibrio de carga.</li> <li>• La medición del desempeño.</li> <li>• Programación y contención.</li> <li>• Evaluación de la comunicación de arriba.</li> <li>• Gestión de datos: <ul style="list-style-type: none"> <li>– Costos de comunicación no uniforme debidos a proximidad</li> <li>– Efectos de Cache (p.e., false sharing)</li> <li>– Manteniendo localidad espacial</li> </ul> </li> <li>• Consumo de energía y gestión.</li> </ul>	<ul style="list-style-type: none"> <li>• Detectar y corregir un desbalanceo de carga [Usar]</li> <li>• Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar]</li> <li>• Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse]</li> <li>• Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar]</li> <li>• Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse]</li> <li>• Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse]</li> <li>• Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]</li> </ul>

**Lecturas :** [Pac11], [Mat14], [KH13], [SK10]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003.
- [SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010.
- [Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011.
- [KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013.
- [Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: <http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.

## Capítulo 10

### Décimo Semestre

## 1. CURSO

CS3P2-CS2023. Cloud Computing (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS3P2-CS2023. Cloud Computing
<b>2.2 Semestre</b>	:	10 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS370. Big Data. (9 <sup>no</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso cubre los fundamentos y tecnologías modernas de computación en la nube, incluyendo virtualización, containerización, procesamiento distribuido y arquitecturas serverless. Los estudiantes aprenderán a implementar soluciones escalables usando plataformas como AWS, Azure y GCP, con enfoque en prácticas actuales de la industria.

## 5. OBJETIVOS

- Diseñar arquitecturas cloud siguiendo mejores prácticas (Well-Architected Framework)
- Implementar soluciones escalables usando servicios gestionados (AWS Lambda, Azure Functions)
- Administrar clusters de contenedores con Kubernetes y Docker
- Optimizar costos y rendimiento en entornos cloud

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (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. (Familiarity)

## 7. TEMAS

<b>Unidad 1: Fundamentos de Cloud Computing (12 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Modelos de servicio (IaaS, PaaS, SaaS, FaaS)</li> <li>• Proveedores globales (AWS, Azure, GCP)</li> <li>• Arquitecturas resilientes (Multi-AZ, Region Failover)</li> <li>• Seguridad e identidad (IAM, RBAC, Zero-Trust)</li> </ul>	<ul style="list-style-type: none"> <li>• Comparar modelos de servicio cloud [Usar]</li> <li>• Diseñar redes VPC/subnets [Evaluar]</li> <li>• Implementar políticas de seguridad [Usar]</li> </ul>
<b>Lecturas :</b> [VC23], [Ser23]	

<b>Unidad 2: Procesamiento de Datos a Escala (15 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Spark 3.0 y Delta Lake</li> <li>• Procesamiento stream (Kafka, Flink)</li> <li>• Data Lakes vs Warehouses (Snowflake, Redshift)</li> <li>• ETL serverless (Glue, Dataflow)</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar pipelines batch/streaming [Usar]</li> <li>• Optimizar consultas distribuidas [Evaluar]</li> <li>• Gestionar metastores de datos [Usar]</li> </ul>
<b>Lecturas :</b> [Dam+23], [KK18]	

<b>Unidad 3: Virtualización y Contenerización (15 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Docker avanzado (buildx, compose)</li> <li>• Kubernetes (ARK, Operators)</li> <li>• Service Mesh (Istio, Linkerd)</li> <li>• Serverless Containers (AWS Fargate, Cloud Run)</li> </ul>	<ul style="list-style-type: none"> <li>• Desplegar clusters EKS/GKE [Usar]</li> <li>• Implementar CI/CD para contenedores [Evaluar]</li> <li>• Monitorizar con Prometheus/Grafana [Usar]</li> </ul>
<b>Lecturas :</b> [BB23], [Mic23]	

<b>Unidad 4: Tendencias en Cloud Computing (12 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Infraestructura como código (Terraform, Pulumi)</li> <li>• GitOps (ArgoCD, Flux)</li> <li>• Edge Computing (AWS Outposts, Azure Arc)</li> <li>• Quantum Computing en la nube (Braket, Qiskit)</li> </ul>	<ul style="list-style-type: none"> <li>• Automatizar despliegues con IaC [Usar]</li> <li>• Implementar patrones GitOps [Evaluar]</li> <li>• Evaluar arquitecturas híbridas [Usar]</li> </ul>
<b>Lecturas :</b> [Clo23], [Cen23]	

<b>Unidad 5: Sistemas distribuídos (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>● Fallos: <ul style="list-style-type: none"> <li>– Fallos basados en red (incluyendo particiones) y fallos basados en nodos</li> <li>– Impacto en garantías a nivel de sistema (p.e., disponibilidad)</li> </ul> </li> <li>● Envío de mensajes distribuido: <ul style="list-style-type: none"> <li>– Conversión y transmisión de datos</li> <li>– Sockets</li> <li>– Secuenciamiento de mensajes</li> <li>– Almacenando <i>Buffering</i>, reenviando y desechando mensajes</li> </ul> </li> <li>● Compensaciones de diseño para Sistemas Distribuidos: <ul style="list-style-type: none"> <li>– Latencia versus rendimiento</li> <li>– Consistencia, disponibilidad, tolerancia de particiones</li> </ul> </li> <li>● Diseño de Servicio Distribuido: <ul style="list-style-type: none"> <li>– Protocolos y servicios Stateful versus stateless</li> <li>– Diseños de Sesión (basados en la conexión)</li> <li>– Diseños reactivos (provocados por E/S) y diseños de múltiples hilos</li> </ul> </li> <li>● Algoritmos de Distribución de Núcleos: <ul style="list-style-type: none"> <li>– Elección, descubrimiento</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● Distinguir las fallas de red de otros tipos de fallas [Familiarizarse]</li> <li>● Explicar por qué estructuras de sincronización como cerraduras simples (<i>locks</i>) no son útiles en la presencia de fallas distribuidas [Familiarizarse]</li> <li>● Escribir un programa que realiza cualquier proceso de <i>marshalling</i> requerido y la conversión en unidades de mensajes, tales como paquetes, para comunicar datos importantes entre dos <i>hosts</i> [Usar]</li> <li>● Medir el rendimiento observado y la latencia de la respuesta a través de los <i>hosts</i> en una red dada [Usar]</li> <li>● Explicar por qué un sistema distribuido no puede ser simultáneamente Consistente (<i>Consistent</i>), Disponible (<i>Available</i>) y Tolerante a fallas (<i>Partition tolerant</i>). [Familiarizarse]</li> <li>● Implementar un servidor sencillo - por ejemplo, un servicio de corrección ortográfica [Usar]</li> <li>● Explicar las ventajas y desventajas entre: <i>overhead</i>, escalabilidad y tolerancia a fallas entre escojer un diseño sin estado (<i>stateless</i>) y un diseño con estado (<i>stateful</i>) para un determinado servicio [Familiarizarse]</li> <li>● Describir los desafíos en la escalabilidad, asociados con un servicio creciente para soportar muchos clientes, así como los asociados con un servicio que tendrá transitoriamente muchos clientes [Familiarizarse]</li> <li>● Dar ejemplos de problemas donde algoritmos de consenso son requeridos, por ejemplo, la elección de líder [Usar]</li> </ul>

**Lecturas :** [Cou+11]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- 
- [Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. Addison-Wesley, 2011.
- [KK18] Holden Karau and Andy Konwinski. *Learning Spark*. O'Reilly, 2018.
- [BB23] Brendan Burns and Joe Beda. *Kubernetes: Up and Running*. 3rd. O'Reilly, 2023.
- [Cen23] IBM Cloud Architecture Center. *Hybrid Cloud Strategies*. Tech. rep. IBM, 2023.
- [Clo23] Google Cloud. *Serverless Architectures with Cloud Functions*. Tech. rep. Google Cloud, 2023.
- [Dam+23] Jules Damji et al. *Learning Spark: Lightning-Fast Data Analytics*. 2nd. O'Reilly, 2023.
- [Mic23] Microsoft. *Containerized Applications on Azure*. Tech. rep. Microsoft Docs, 2023.
- [Ser23] Amazon Web Services. *AWS Well-Architected Framework*. Tech. rep. Official cloud best practices guide. AWS, 2023.
- [VC23] Diego Villegas and Mark Cresswell. *Cloud Native Architectures*. Covers multi-cloud patterns with case studies. Packt, 2023.

## 1. CURSO

CS404. Taller de Investigación (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS404. Taller de Investigación
<b>2.2 Semestre</b>	:	10 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	3
<b>2.4 Horas</b>	:	1 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	CS403. Proyecto de tesis 2. (9 <sup>no</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso tiene por objetivo que el alumno logre finalizar adecuadamente su borrador de tesis.

## 5. OBJETIVOS

- Que el alumno complete este curso con su tesis elaborada en calidad suficiente como para una inmediata sustentación.
- Que el alumno presente formalmente el borrador de tesis ante las autoridades de la facultad.
- Los entregables de este curso son:

**Parcial:** Avance del proyecto de tesis incluyendo en el documento: introducción, marco teórico, estado del arte, propuesta, análisis y/o experimentos y bibliografía actualizada.

**Final:** Documento de tesis completo y listo para sustentar en un plazo no mayor de quince días.

## 6. RESULTADOS DEL ESTUDIANTE

1) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

2) Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Assessment)

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

4) Reconocer las responsabilidades profesionales y tomar decisiones informadas en la práctica de la computación basadas en principios legales y éticos. (Assessment)

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

5) Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. (Assessment)

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

6) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

**AG-C12)** Aplica la teoría de la ciencia de la computación y los fundamentos de desarrollo de software para producir soluciones basadas en computadora. (Assessment)

## 7. TEMAS

<b>Unidad 1: Escritura del borrador del trabajo de final de carrera (tesis) (60 horas)</b>	
<b>Resultados esperados: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>Redacción y corrección del trabajo de final de carrera</li></ul>	<ul style="list-style-type: none"><li>Parte experimental concluída (si fuese adecuado al proyecto) [Evaluar]</li><li>Verificar que el documento cumpla con el formato de tesis de la carrera [Evaluar]</li><li>Entrega del borrador de tesis finalizado y considerado listo para una sustentación pública del mismo (requisito de aprobación) [Evaluar]</li></ul>
<b>Lecturas :</b> [IEE08], [ACM08], [Cit08]	

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

[ACM08] ACM. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.

[Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.

[IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.

## 1. CURSO

CS367-CS2023. Tópicos en Inteligencia Artificial (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS367-CS2023. Tópicos en Inteligencia Artificial
<b>2.2 Semestre</b>	:	10 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI264. Aprendizaje Profundo. (7 <sup>mo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Provee una serie de herramientas para resolver problemas que son difíciles de solucionar con los métodos algorítmicos tradicionales. Incluyendo heurísticas, planeamiento, formalismos en la representación del conocimiento y del razonamiento, técnicas de aprendizaje en máquinas, técnicas aplicables a los problemas de acción y reacción: así como el aprendizaje de lenguaje natural, visión artificial y robótica entre otros.

## 5. OBJETIVOS

- Realizar algún curso avanzado de Inteligencia Artificial sugerido por el currículo de la ACM/IEEE.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

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

## 7. TEMAS

<b>Unidad 1: (60 horas)</b>	
<b>Resultados esperados: 2,6,AG-C09</b>	
Temas	Objetivos de Aprendizaje ( <i>Learning Outcomes</i> )
<ul style="list-style-type: none"><li>• Tópicos avanzados en IA</li></ul>	<ul style="list-style-type: none"><li>• Profundizar en diversas técnicas relacionadas a la Inteligencia Artificial [Usar]</li></ul>
<b>Lecturas :</b> [RN03], [Hay99], [Gol89]	

---

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [Hay99] Simon Haykin. *Neural networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [RN03] Stuart Russell and Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Prentice Hall, 2003.

## 1. CURSO

CS368-CS2023. Computación Evolutiva (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	CS368-CS2023. Computación Evolutiva
<b>2.2 Semestre</b>	:	10 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	AI264. Aprendizaje Profundo. (7 <sup>mo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

Este curso introduce algoritmos inspirados en la evolución biológica para resolver problemas complejos de optimización y diseño. Cubre algoritmos genéticos, estrategias evolutivas y programación genética, con aplicaciones en ingeniería, logística y machine learning. Los estudiantes implementarán soluciones usando frameworks modernos como DEAP.

## 5. OBJETIVOS

- Modelar problemas complejos usando paradigmas evolutivos.
- Implementar algoritmos bio-inspirados en Python.
- Analizar el desempeño de diferentes técnicas evolutivas.

## 6. RESULTADOS DEL ESTUDIANTE

**6)** Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

**2)** Diseñar, implementar y evaluar una solución basada en la computación para satisfacer un conjunto dado de requisitos de computación en el contexto de la disciplina del programa. (Usage)

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

**1)** Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Familiarity)

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

## 7. TEMAS

<b>Unidad 1: Algoritmos Genéticos (16 horas)</b> <b>Resultados esperados: 1,AG-C08</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Representación cromosómica (binaria, real)</li> <li>• Operadores de selección (ruleta, torneo)</li> <li>• Cruce y mutación</li> </ul>	<ul style="list-style-type: none"> <li>• Diseñar representaciones para problemas discretos [Usar]</li> <li>• Comparar operadores de selección [Evaluar]</li> </ul>

<b>Unidad 2: Estrategias Evolutivas (16 horas)</b> <b>Resultados esperados: 2,AG-C09</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Algoritmos <math>(\mu + \lambda)</math> y <math>(\mu, \lambda)</math></li> <li>• Auto-adaptación de parámetros</li> <li>• Aplicaciones en optimización continua</li> </ul>	<ul style="list-style-type: none"> <li>• Implementar estrategias para problemas reales [Usar]</li> <li>• Sintonizar parámetros de auto-adaptación [Evaluar]</li> </ul>

<b>Unidad 3: Aplicaciones Avanzadas (16 horas)</b> <b>Resultados esperados: 6</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Programación genética</li> <li>• Co-evolución</li> <li>• Optimización multi-objetivo (NSGA-II)</li> </ul>	<ul style="list-style-type: none"> <li>• Desarrollar soluciones multi-objetivo [Usar]</li> <li>• Evaluar compensaciones en frontes de Pareto [Evaluar]</li> </ul>

**Lecturas :** [Bey01]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [Bey01] Hans-Georg Beyer. *The Theory of Evolution Strategies*. Fundamentos matemáticos. Springer, 2001.
- [ES15] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Visión general actualizada. Springer, 2015.
- [Gol18] David E. Goldberg. *Algoritmos Genéticos*. Clásico en algoritmos genéticos. Addison-Wesley, 2018.
- [Aut22] Various Authors. “Evolutionary Computation in Industry”. In: *Nature Reviews* 3 (2022). Aplicaciones industriales modernas. URL: <https://www.nature.com/articles/s42254-022-00490-y>.
- [Tea23] DEAP Team. *Documentación de DEAP*. Framework para computación evolutiva en Python. 2023. URL: <https://deap.readthedocs.io>.

## 1. CURSO

FG350. Liderazgo y Desempeño (Obligatorio)

## 2. INFORMACIÓN GENERAL

<b>2.1 Curso</b>	:	FG350. Liderazgo y Desempeño
<b>2.2 Semestre</b>	:	10 <sup>mo</sup> Semestre
<b>2.3 Créditos</b>	:	2
<b>2.4 Horas</b>	:	2 HT;
<b>2.5 Duración del periodo</b>	:	16 semanas
<b>2.6 Condición</b>	:	Obligatorio
<b>2.7 Modalidad de aprendizaje</b>	:	Presencial
<b>2.8 Prerrequisitos</b>	:	FG106. Teatro. (8 <sup>vo</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL CURSO

En la actualidad las diferentes organizaciones en el mundo exigen a sus integrantes el ejercicio de liderazgo, esto significa asumir los retos asignados con eficacia y afán de servicio, siendo estas exigencias necesarias para la búsqueda de una sociedad más justa y reconciliada. Este desafío, pasa por la necesidad de formar a nuestros alumnos con un recto conocimiento de sí mismos, con capacidad de juzgar objetivamente la realidad y de proponer orientaciones que busquen modificar positivamente el entorno.

## 5. OBJETIVOS

- Desarrollar conocimientos, criterios, capacidades y actitudes para ejercer liderazgo, con el objeto de lograr la eficacia y servicio en los retos asignados, contribuyendo así en la construcción de una mejor sociedad.

## 6. RESULTADOS DEL ESTUDIANTE

3) Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

## 7. TEMAS

<b>Unidad 1: Primera Unidad: Fundamentos del liderazgo (15 horas)</b>	
<b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"><li>• Teorías de Liderazgo:</li><li>• Definición de Liderazgo.</li><li>• Fundamentos de Liderazgo.</li><li>• Visión integral del Ser Humano y Motivos de la acción.</li><li>• La práctica de la Virtud en el ejercicio de Liderazgo.</li></ul>	<ul style="list-style-type: none"><li>• Analizar y comprender las bases teóricas del ejercicio de Liderazgo.[Familiarizarse]</li><li>• En base a lo comprendido, asumir la actitud correcta para llevarlo a la práctica.[Familiarizarse]</li><li>• Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo.[Familiarizarse]</li></ul>
<b>Lecturas :</b> [CL02], [FA09], [Dia09], [Ant], [Sonnenfeld]	

<b>Unidad 2: (15 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• Teoría de las Competencias</li> <li>• Reconocimiento de Competencias</li> <li>• Plan de Desarrollo</li> <li>• Modelos Mentales</li> <li>• Necesidades Emocionales</li> <li>• Perfiles Emocionales</li> <li>• Vicios Motivacionales</li> </ul>	<ul style="list-style-type: none"> <li>• Conocer y Desarrollar competencias de Liderazgo, centradas en lograr la eficacia, sin dejar de lado el deber de servicio con los demás.[Familiarizarse]</li> <li>• Reconocer las tendencias personales y grupales necesarias para el ejercicio de Liderazgo.[Familiarizarse]</li> </ul>

**Lecturas :** [CW09], [Hue08], [CL02], [CM07]

<b>Unidad 3: (18 horas)</b> <b>Resultados esperados:</b>	
<b>Temas</b>	<b>Objetivos de Aprendizaje (Learning Outcomes)</b>
<ul style="list-style-type: none"> <li>• La relación personal con el equipo</li> <li>• Liderazgo integral</li> <li>• Acompañamiento y discipulado</li> <li>• Fundamentos de unidad</li> </ul>	<ul style="list-style-type: none"> <li>• Desarrollar habilidades para el trabajo en equipo[Familiarizarse]</li> </ul>

**Lecturas :** [Gol12], [CP], [HBJ98], [Hun10], [Haw12], [GS10]

## 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

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA

- [HBJ98] P. Hersey, K. H. Blanchard, and D. E. Johnson. *Administración del comportamiento organizacional: liderazgo situacional*. 1998.
- [CL02] Pablo Cardona and Pilar García Lombardi. *Cómo desarrollar las Competencias de Liderazgo*. PAD Lima- Perú, Tercera Edición, 2002.
- [CM07] Nuria Chinchilla and Maruja Moragas. *Dueños de Nuestro Destino*. Editorial Ariel, 2007.
- [Hue08] Luis Huete. *Construye tu Sueño*. LID Editorial Empresarial, 2008.
- [CW09] Pablo Cardona and Helen Wilkinson. *Creciendo como Líder*. Ediciones Universidad de Navarra S.A (EUNSA), Primera Edición, 2009.
- [Dia09] Alexandre Dianne-Havard. *Perfil del Líder. Hacia un Liderazgo Virtuoso*. Ediciones Urano S.A, 2009.

- 
- [FA09] Pablo Ferreiro and Manuel Alcázar. *Gobierno de Personas en la Empresa*. Ediciones Universidad de Navarra EUNSA, 2009.
- [GS10] G. Ginebra and G. G. Serrabou. *Gestión de incompetentes. Libros de Cabecera*. 2010.
- [Hun10] Phil Hunsaker. *El nuevo arte de gestionar equipos: Un enfoque actual para guiar y motivar con éxito*. 2010.
- [Gol12] D. Goleman. *Inteligencia emocional*. Editorial Kairós, 2012.
- [Haw12] Peter Hawkins. *Coaching y liderazgo de equipos: coaching para un liderazgo con capacidad de transformación*. Ediciones Granica, 2012.
- [Ant] SJ Anthony D' Souza. *Descubre tu Liderazgo*. Editorial Sal Terrae.
- [CP] P. Cardona and C. R. Peña. *Dirección por misiones: Cómo generar empresas de alto rendimiento*.