



Book of Syllabi

Ciencia de la Computación

*- 2024-II -*

**: June 9, 2025**

# Equipo de Trabajo

**Ciro Nuñez Iturri**

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

**Dr. Javier Solano Salinas**

Director del Departamento de Ciencia de la Computación  
email: *jsolano@uni.edu.pe*

**Yuri Nuñez Medrano**

Profesor Investigador  
Departamento de Ciencia de la Computación  
email: *ynunezm@uni.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*  
<http://socios.spc.org.pe/ecuadros>

# Contents

<b>First Semester</b>	<b>5</b>
1.1 CS100. Introduction to Computer Science . . . . .	5
1.2 CS111. Introduction to Programming . . . . .	19
1.3 MA101FCCS. Linear Algebra . . . . .	27
1.4 MA102FCCS. Integral Calculus . . . . .	30
1.5 FI101FCCS. Physics I . . . . .	33
1.6 CH101FCCS. Chemistry I . . . . .	36
1.7 ID101. Technical and professional English I . . . . .	39
<b>Second Semester</b>	<b>42</b>
2.1 CS112. Computer Science I . . . . .	42
2.2 CS1D1. Discrete Structures I . . . . .	49
2.3 MA103FCCS. Differential Calculus . . . . .	53
2.4 FI201FCCS. Computational Physics . . . . .	56
2.5 BI101FCCS. Biology I . . . . .	59
2.6 ID102. Technical and professional English II . . . . .	62
<b>Third Semester</b>	<b>66</b>
3.1 CS113. Computer Science II . . . . .	66
3.2 CS1D2. Discrete Structures II . . . . .	70
3.3 CS2B1. Platform Based Development . . . . .	74
3.4 MA104FCCS. Advanced Differential and Integral Calculus . . . . .	78
3.5 EC101FCCS. General Economy . . . . .	81
3.6 ID103. Technical and professional English III . . . . .	84
<b>Fourth Semester</b>	<b>88</b>
4.1 CS210. Algorithms and Data Structures . . . . .	88
4.2 CS211. Theory of Computation . . . . .	91
4.3 CS221. Computer Systems Architecture . . . . .	94
4.4 CS271. Data Management . . . . .	101
4.5 MA106FCCS. Numerical Methods . . . . .	106
4.6 ST251FCCS. Probability Calculation . . . . .	109
4.7 ID104. Technical and professional English IV . . . . .	112
<b>Fifth Semester</b>	<b>116</b>
5.1 CS212. Analysis and Design of Algorithms . . . . .	116
5.2 CS272. Databases II . . . . .	120
5.3 CS291. Software Engineering I . . . . .	124

5.4	CS2S1. Operating systems . . . . .	129
5.5	CS342. Compilers . . . . .	137
5.6	ID105. Technical and professional English V . . . . .	142
<b>Sixth Semester</b>		<b>146</b>
6.1	CS231. Networking and Communication . . . . .	146
6.2	CS261. Artificial Intelligence . . . . .	150
6.3	CS2H1. User Experience (UX) . . . . .	159
6.4	CS311. Competitive Programming . . . . .	165
6.5	CS312. Advanced Data Structures . . . . .	169
6.6	MA307. Mathematics applied to computing . . . . .	173
<b>Seventh Semester</b>		<b>176</b>
7.1	CS251. Computer graphics . . . . .	176
7.2	CS292. Software Engineering II . . . . .	183
7.3	CS393. Information systems . . . . .	189
7.4	CS3I1. Computer Security . . . . .	191
7.5	FG211-ACM. Professional Ethics . . . . .	201
7.6	CS262. Machine learning . . . . .	204
<b>Eighth Semester</b>		<b>206</b>
8.1	CS281. Computing in Society . . . . .	206
8.2	CS391. Software Engineering III . . . . .	215
8.3	CS3P1. Parallel and Distributed Computing . . . . .	221
8.4	CS401. Capstone Project I . . . . .	227
8.5	FG120FCCS. General Economy . . . . .	229
8.6	EX301FCCS. Extracurricular Activities . . . . .	232
8.7	CS361. Computational Vision . . . . .	234
<b>Ninth Semester</b>		<b>236</b>
9.1	CS370. Big Data . . . . .	236
9.2	CS400FCCS. Pre-professional internships . . . . .	239
9.3	CS402. Capstone Project II . . . . .	241
9.4	CB309. Bioinformatics . . . . .	243
9.5	CS369. Topics in Artificial Intelligence . . . . .	247
9.6	CS351. Topics in Computer Graphics . . . . .	249
9.7	CS392. Tópicos en Ingeniería de Software . . . . .	251
<b>Tenth Semester</b>		<b>256</b>
10.1	CS353. Quantum Computing . . . . .	256
10.2	CS365. Evolutionary Computing . . . . .	258
10.3	CS3P2. Cloud Computing . . . . .	260
10.4	CS3P3. Internet of Things . . . . .	263
10.5	CS403. Research Workshop . . . . .	269
10.6	FG211. Professional Ethics . . . . .	271



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS100. Introduction to Computer Science (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	CS100. Introduction to Computer Science
<b>2.2 Semester</b>	:	1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course serves as the foundation for understanding the fundamental concepts of computational thinking applicable across various professions.

The course provides, starting from ground zero, a panoramic view of: introductory computational thinking, data storage, computer architecture, operating systems, networks and the Internet, algorithms, sorting methods, software engineering, databases, data structures, software engineering, computer graphics, artificial intelligence among others.

Designed as an introductory course to Computer Science, the concepts are presented in a playful manner and using an Active Learning methodology. Throughout the course, active audience participation is encouraged, akin to a theatrical performance.

The related knowledge areas covered are directly aligned with the Computing Curricula ACM/IEEE-CS.

The course **does not require** any prior knowledge in computer handling topics and can be taken by student from any field.

**5. GOALS**

- Introduce the fundamental concepts of Computational Thinking and Computer Science to students from any professional background.
- Develop their ability to abstract.
- Understand how Computational Thinking is applied in each of their professions.
- Apply advanced concepts in a simplified manner in any career.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

<b>Unit 1: Computational Thinking (Part I) (4 hours)</b>	
<b>Competences Expected: 1,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• General course instructions.</li> <li>• Explanation of the evaluation system.</li> <li>• Definition of Computing.</li> <li>• Computing as a Human-Computer dyad.</li> <li>• Distortions in the definition of computing.</li> <li>• Computing as the automation of abstraction.</li> <li>• Computing and Engineering: similarities and differences.</li> <li>• Algorithmic problem-solving.</li> <li>• Dynamics: Understanding the execution of an algorithm at human speed.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply the fundamental concepts of computing in real-life situations. [Usar]</li> <li>• Identify distortions of Computing in real-life situations. [Usar]</li> <li>• Clearly identify at least 3 contexts of using the word“Engineer” in English. [Evaluar]</li> <li>• Identify the limitations of humans in solving computational problems. [Usar]</li> </ul>
<b>Readings :</b> [BB19]	

<b>Unit 2: Computational Thinking. Part II (4 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Binary vs. decimal numbering.</li> <li>• Character representation: the ASCII table.</li> <li>• Internal representation of colors.</li> <li>• Understanding an image from the inside.</li> <li>• Binary search.</li> <li>• Computational complexity of an algorithm.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply various numbering systems to real-world problems. [Usar]</li> <li>• Understand the internal representation of characters in the ASCII and UTF-8 tables. [Familiarizarse]</li> <li>• Understand the representation of colors in an image. [Familiarizarse]</li> <li>• Apply Divide and Conquer algorithmic strategy. [Usar]</li> <li>• Determine basic analysis of algorithmic complexity. [Usar]</li> </ul>
<b>Readings :</b> [BB19]	

Unit 3: Lógica digital y sistemas digitales (4 hours)	
Competences Expected: 1,2,6	
Topics	Learning Outcomes
<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>• Múltiples 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>
Readings : [BB19]	

Unit 4: Representación de programas (2 hours)	
Competences Expected: 1,2,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Programas que tienen otros programas como entrada tales como interpretes, 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> </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>
Readings : [BB19]	

<b>Unit 5: Criptografía (2 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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>Readings :</b> [BB19]	

<b>Unit 6: Organización y Arquitectura del Sistema de Memoria (4 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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>Readings :</b> [BB19]	



<b>Unit 7: Visión general de Sistemas Operativos (4 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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 contemporáneo 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>Readings :</b> [BB19]	

<b>Unit 8: Introducción a redes (4 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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>Readings :</b> [BB19]	

<b>Unit 9: Entrega confiable de datos (4 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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>Readings :</b> [BB19]	

**Unit 10: Análisis Básico (4 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [BB19]	

**Unit 11: Algoritmos y Estructuras de Datos fundamentales (8 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<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>• 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 en específico [Evaluar]</li></ul>
<b>Readings :</b> [BB19]	

**Unit 12: Sistemas de Bases de Datos (4 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<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 largos 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>
<b>Readings :</b> [BB19]	

**Unit 13: Programación orientada a objetos (4 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>● Diseño orientado a objetos:               <ul style="list-style-type: none"> <li>– Descomposicion en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquia 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 coleccion de clases, iteradores, y otros componentes de la libreria 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 procedurar/funcional- definiendo una función por cada operación con el cuerdo 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>
<b>Readings :</b> [BB19]	

Unit 14: Procesos de Software (4 hours)	
Competences Expected: 1,2,6	
Topics	Learning Outcomes
<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, ágil): <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>• 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> </ul>
Readings : [BB19]	

Unit 15: Cuestiones fundamentales (2 hours)	
Competences Expected: 1,2,6	
Topics	Learning Outcomes
<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>
Readings : [BB19]	

<b>Unit 16: Estrategias de búsquedas básicas (1 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<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>
<b>Readings :</b> [BB19]	

<b>Unit 17: Aprendizaje Automático Básico (1 hours)</b>	
<b>Competences Expected: 1,2,6</b>	
<b>Topics</b>	<b>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 [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>
<b>Readings :</b> [BB19]	

**Unit 18: Conceptos Fundamentales (2 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<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 imagenes 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 ejemplificando 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 básico 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>
<b>Readings :</b> [BB19]	



**Unit 19: Rendering Básico (2 hours)****Competences Expected: 1,2,6**

Topics	Learning Outcomes
<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>• 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 afin [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>
<b>Readings :</b> [BB19]	

Unit 20: Closure Class: How Does a Search Engine Like Google Work? (2 hours)	
Competences Expected: 1,2,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Problem analysis</li> <li>• The index does not grow linearly with the size of the indexed information.</li> <li>• Response time does not depend on the size of the “database.”</li> <li>• Response time does not depend on the number of occurrences found.</li> <li>• Combining various data structures to reach a solution.</li> <li>• Analyzing the scalability of the solution.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the principles under which a search engine is created [Usar]</li> <li>• Correctly apply data structures to solve the problem [Usar]</li> <li>• Apply concepts related to algorithmic complexity in a search engine [Usar].</li> </ul>
Readings : [BB19]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

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



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS111. Introduction to Programming (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS111. Introduction to Programming
<b>2.2 Semester</b>	: 1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	: 4
<b>2.4 Horas</b>	: 2 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This is the first course in the sequence of introductory courses to Computer Science. This course is intended to cover the concepts outlined by the Computing Curricula ACM/IEEE-CS 2013. Programming is one of the pillars of Computer Science; any professional of the area, will need to program to materialize their models and proposals. This course introduces participants to the fundamental concepts of this art. Topics include data types, control structures, functions, lists, recursion, and the mechanics of execution, testing, and debugging.

**5. GOALS**

- Introduce the fundamental concepts of programming.
- Develop the ability of abstraction using programming language

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Historia (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
<b>Readings :</b> [BB19], [Gut13], [Zel10]	

Unit 2: Sistemas de tipos básicos (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 contruídos 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>
<b>Readings :</b> [Gut13], [Zel10]	

**Unit 3: Conceptos Fundamentales de Programación (9 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 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>	<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>• Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [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>Readings :</b> [Gut13], [Zel10]	

**Unit 4: Análisis Básico (2 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 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 [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>Readings :</b> [Gut13], [Zel10]	

**Unit 5: Algoritmos y Estructuras de Datos fundamentales (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>• 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 en específico [Evaluar]</li> <li>• Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Familiarizarse]</li> <li>• Resolver problemas usando algoritmos de grafos, incluyendo camino más corto de una sola fuente y camino más corto de todos los pares, y como mínimo un algoritmo de arbol de expansion minima [Usar]</li> <li>• Trazar y/o implementar un algoritmo de comparación de string [Usar]</li> </ul>
<b>Readings :</b> [Gut13], [Zel10]	

**Unit 6: Programación orientada a objetos (4 hours)****Competences Expected: 1**

Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Object oriented languages and encapsulation: Privacy y visibility of class members.</li><li>• Definición de las categorías, campos, métodos y constructores.</li><li>• Subclasses and inheritance.</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 procedurador/funcional- definiendo una función por cada operación con el uso 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 (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>• Use encapsulation to create objects [Familiarizarse].</li></ul>

**Readings :** [Gut13], [Zel10]



Unit 7: Algoritmos y Diseño (9 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 Program</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>
Readings : [Gut13], [Zel10]	

Unit 8: Métodos de Desarrollo (1 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Gut13], [Zel10]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

### 9. EVALUATION SYSTEM

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

### 10. BASIC BIBLIOGRAPHY

- [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>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

MA101FCCS. Linear Algebra (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: MA101FCCS. Linear Algebra
<b>2.2 Semester</b>	: 1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Linear algebra is fundamental to computer science, providing essential tools for algorithm analysis, computer graphics, machine learning, and many other areas. This course provides a solid foundation in the concepts and techniques of linear algebra, with a focus on its application in computing.

**5. GOALS**

- Understand the fundamental concepts of linear algebra, including vector spaces, matrices, linear transformations, and systems of linear equations.
- Apply linear algebra techniques to solve problems in various computational contexts.
- Develop abstract reasoning and logical thinking skills to address mathematical problems.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C08** Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Assessment)
- AG-C12** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Vector Spaces (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C08</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Definition of vector space and subspace.</li> <li>• Linear combinations, linear independence, and bases.</li> <li>• Dimension and rank.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and give examples of vector spaces and subspaces. [Familiarizarse]</li> <li>• Determine the linear independence of a set of vectors. [Usar]</li> <li>• Calculate bases and the dimension of a vector space. [Evaluar]</li> </ul>
<b>Readings :</b> [Str16], [LLM16]	

<b>Unit 2: Matrices and Systems of Linear Equations (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Matrix operations.</li> <li>• Gaussian elimination and reduced row echelon form.</li> <li>• Solving systems of linear equations.</li> <li>• Inverse matrices and determinants.</li> </ul>	<ul style="list-style-type: none"> <li>• Perform matrix operations. [Familiarizarse]</li> <li>• Solve systems of linear equations using Gaussian elimination. [Usar]</li> <li>• Calculate the inverse of a matrix and its determinant. [Evaluar]</li> </ul>
<b>Readings :</b> [Str16], [LLM16]	

<b>Unit 3: Linear Transformations (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Definition and examples of linear transformations.</li> <li>• Kernel and image of a linear transformation.</li> <li>• Transformation matrices.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and give examples of linear transformations. [Familiarizarse]</li> <li>• Calculate the kernel and image of a linear transformation. [Usar]</li> <li>• Represent linear transformations using matrices. [Evaluar]</li> </ul>
<b>Readings :</b> [Str16], [LLM16]	

<b>Unit 4: Eigenvalues and Eigenvectors (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Definition and calculation of eigenvalues and eigenvectors.</li> <li>• Diagonalization of matrices.</li> <li>• Applications of eigenvalues and eigenvectors.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and calculate eigenvalues and eigenvectors. [Familiarizarse]</li> <li>• Diagonalize matrices. [Usar]</li> <li>• Apply eigenvalues and eigenvectors to solve problems. [Evaluar]</li> </ul>
<b>Readings :</b> [Str16], [LLM16]	

Unit 5: Orthogonality and Least Squares (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Inner product and orthogonality.</li> <li>• Orthogonal projections.</li> <li>• Least squares method.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and calculate inner product and orthogonality. [Familiarizarse]</li> <li>• Calculate orthogonal projections. [Usar]</li> <li>• Apply the least squares method. [Evaluar]</li> </ul>
Readings : [Str16], [LLM16]	

Unit 6: Applications in Computing (8 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Applications in computer graphics.</li> <li>• Applications in machine learning.</li> <li>• Applications in algorithm analysis.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe applications of linear algebra in computer graphics. [Familiarizarse]</li> <li>• Explain how linear algebra is used in machine learning. [Usar]</li> <li>• Analyze the complexity of algorithms using linear algebra concepts. [Evaluar]</li> </ul>
Readings : [Str16], [LLM16]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

MA102FCCS. Integral Calculus (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: MA102FCCS. Integral Calculus
<b>2.2 Semester</b>	: 1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Differential calculus is a fundamental tool in computer science for understanding and modeling change. This course introduces the main concepts of differential calculus, including limits, derivatives, applications of the derivative, and optimization.

**5. GOALS**

- Understand the concept of a limit and its application to calculating derivatives.
- Apply differentiation rules to calculate derivatives of various functions.
- Use the derivative to solve optimization problems, rates of change, and function analysis.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Functions and Limits (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Review of functions.</li> <li>• Definition of a limit.</li> <li>• Properties of limits.</li> <li>• Limits involving infinity.</li> <li>• Continuity.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluate limits graphically and numerically. [Familiarizarse]</li> <li>• Apply the properties of limits to evaluate limits algebraically. [Usar]</li> <li>• Determine the continuity of a function. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 2: The Derivative (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Definition of the derivative.</li> <li>• Geometric interpretation of the derivative.</li> <li>• Derivatives of polynomial and exponential functions.</li> <li>• Differentiation rules: sum, product, quotient, and chain rule.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate the derivative of a function using the definition. [Familiarizarse]</li> <li>• Interpret the derivative as the slope of the tangent line. [Usar]</li> <li>• Apply differentiation rules to find derivatives of functions. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 3: Applications of the Derivative (12 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Related rates.</li> <li>• Maximum and minimum values.</li> <li>• Mean Value Theorem.</li> <li>• Concavity and inflection points.</li> <li>• Optimization.</li> </ul>	<ul style="list-style-type: none"> <li>• Solve related rates problems. [Familiarizarse]</li> <li>• Find maximum and minimum values of a function. [Usar]</li> <li>• Apply the Mean Value Theorem. [Evaluar]</li> <li>• Determine the concavity and inflection points of a function. [Evaluar]</li> <li>• Solve optimization problems. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 4: Transcendental Functions (12 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Inverse trigonometric functions.</li> <li>• Hyperbolic functions.</li> <li>• Derivatives of inverse trigonometric and hyperbolic functions.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluate inverse trigonometric functions. [Familiarizarse]</li> <li>• Define and manipulate hyperbolic functions. [Usar]</li> <li>• Differentiate inverse trigonometric and hyperbolic functions. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

Unit 5: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Algorithm optimization.</li> <li>• Modeling dynamic systems.</li> <li>• Machine learning (e.g., gradient descent).</li> </ul>	<ul style="list-style-type: none"> <li>• Use derivatives to optimize algorithms. [Familiarize]</li> <li>• Model dynamic systems using differential equations. [Usar]</li> <li>• Apply differential calculus in machine learning algorithms. [Evaluar]</li> </ul>
Readings : [Ste15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

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

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





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

FI101FCCS. Physics I (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	FI101FCCS. Physics I
<b>2.2 Semester</b>	:	1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Physics is essential for understanding the world around us, and its principles are fundamental in many areas of computer science, such as computer graphics, physical simulations, and robotics. This course introduces the basic concepts of classical mechanics, including kinematics, dynamics, work, and energy.

**5. GOALS**

- Understand the fundamental laws of classical mechanics.
- Apply these laws to solve problems of motion in one and two dimensions.
- Develop skills to analyze physical systems and model them mathematically.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C09** Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)
- AG-C12** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Kinematics (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Displacement, velocity, and acceleration.</li> <li>• Uniform and uniformly accelerated linear motion.</li> <li>• Projectile motion.</li> <li>• Uniform circular motion.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and calculate displacement, velocity, and acceleration. [Familiarizarse]</li> <li>• Solve problems involving linear motion and projectile motion. [Usar]</li> <li>• Analyze uniform circular motion. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [SJ18]	

<b>Unit 2: Dynamics (10 hours)</b>	
<b>Competences Expected: 1,AG-C07,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Newton's laws of motion.</li> <li>• Forces of friction.</li> <li>• Work and energy.</li> <li>• Work-kinetic energy theorem.</li> <li>• Power.</li> </ul>	<ul style="list-style-type: none"> <li>• State and apply Newton's laws of motion. [Familiarizarse]</li> <li>• Calculate the work done by a force. [Usar]</li> <li>• Apply the work-kinetic energy theorem to solve dynamics problems. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [SJ18]	

<b>Unit 3: Conservation of Energy (8 hours)</b>	
<b>Competences Expected: 1,AG-C07,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Potential energy.</li> <li>• Conservation of mechanical energy.</li> <li>• Conservative and non-conservative forces.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and calculate potential energy. [Familiarizarse]</li> <li>• Apply the principle of conservation of mechanical energy. [Usar]</li> <li>• Distinguish between conservative and non-conservative forces. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [SJ18]	

<b>Unit 4: Systems of Particles and Conservation of Linear Momentum (6 hours)</b>	
<b>Competences Expected: 1,AG-C07,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Center of mass.</li> <li>• Linear momentum.</li> <li>• Conservation of linear momentum.</li> <li>• Collisions.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate the center of mass of a system of particles. [Familiarizarse]</li> <li>• Apply the principle of conservation of linear momentum. [Usar]</li> <li>• Analyze elastic and inelastic collisions. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [SJ18]	

Unit 5: Rotation (8 hours)	
Competences Expected: 1,AG-C07,AG-C09	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Rotational kinematics.</li> <li>• Rotational dynamics.</li> <li>• Moment of inertia.</li> <li>• Torque and rotational kinetic energy.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe rotational motion using angular variables. [Familiarizarse]</li> <li>• Calculate the moment of inertia of simple objects. [Usar]</li> <li>• Apply the laws of rotational dynamics. [Evaluar]</li> </ul>
Readings : [YF18], [SJ18]	

Unit 6: Applications in Computing (8 hours)	
Competences Expected: 1,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Physical simulations.</li> <li>• Computer graphics.</li> <li>• Robotics.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain how the principles of physics are used in physical simulations. [Familiarizarse]</li> <li>• Describe the application of physics in computer graphics. [Usar]</li> <li>• Analyze the use of physics in robotics. [Evaluar]</li> </ul>
Readings : [YF18]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[SJ18] Raymond A. Serway and John W. Jewett. *Physics for Scientists and Engineers with Modern Physics*. Cengage Learning, 2018.

[YF18] Hugh D. Young and Roger A. Freedman. *University Physics with Modern Physics*. Pearson, 2018.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CH101FCCS. Chemistry I (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	CH101FCCS. Chemistry I
<b>2.2 Semester</b>	:	1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Chemistry provides a foundation for understanding the composition, structure, and properties of matter. While not directly related to many aspects of programming, chemistry is relevant to computer science in areas such as materials science (developing new materials for computer components), nanotechnology, and bioinformatics. This course introduces the basic principles of general chemistry.

**5. GOALS**

- Understand the structure of matter at the atomic and molecular level.
- Apply the principles of stoichiometry to perform chemical calculations.
- Understand the different types of chemical bonds and their influence on the properties of substances.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C09)** Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Usage)

**7. TOPICS**

<b>Unit 1: Introduction to Chemistry (6 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Matter and energy.</li> <li>• The scientific method.</li> <li>• Units of measurement.</li> <li>• Classification of matter.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the properties of matter and energy. [Familiarizarse]</li> <li>• Apply the scientific method to solve chemical problems. [Usar]</li> <li>• Perform unit conversions. [Evaluar]</li> </ul>
<b>Readings :</b> [BrownLeMayBursten2017], [CG16]	

<b>Unit 2: Atomic Structure (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Structure of the atom.</li> <li>• Atomic number, mass number, and isotopes.</li> <li>• Quantum model of the atom.</li> <li>• Electron configuration.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the structure of the atom. [Familiarizarse]</li> <li>• Determine the electron configuration of an atom. [Usar]</li> <li>• Relate electron configuration to chemical properties. [Evaluar]</li> </ul>
<b>Readings :</b> [BrownLeMayBursten2017], [CG16]	

<b>Unit 3: Chemical Bonds (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Ionic bonds.</li> <li>• Covalent bonds.</li> <li>• Metallic bonds.</li> <li>• Molecular geometry.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the different types of chemical bonds. [Familiarizarse]</li> <li>• Predict the molecular geometry of a molecule. [Usar]</li> <li>• Relate the type of bond to the properties of substances. [Evaluar]</li> </ul>
<b>Readings :</b> [BrownLeMayBursten2017], [CG16]	

<b>Unit 4: Stoichiometry (8 hours)</b>	
<b>Competences Expected: 1,AG-C07,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Molar mass and mole.</li> <li>• Chemical reactions and chemical equations.</li> <li>• Stoichiometric calculations.</li> <li>• Limiting reactant and percent yield.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate the molar mass of a compound. [Familiarizarse]</li> <li>• Balance chemical equations. [Usar]</li> <li>• Perform stoichiometric calculations to determine the amount of reactants and products. [Evaluar]</li> </ul>
<b>Readings :</b> [BrownLeMayBursten2017], [CG16]	

Unit 5: States of Matter (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Gases.</li> <li>• Liquids.</li> <li>• Solids.</li> <li>• Changes of state.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the properties of the different states of matter. [Familiarizarse]</li> <li>• Explain changes of state and phase diagrams. [Usar]</li> <li>• Apply the ideal gas laws. [Evaluar]</li> </ul>
Readings : [BrownLeMayBursten2017], [CG16]	

Unit 6: Chemistry and Computing (10 hours)	
Competences Expected: 1,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Molecular modeling.</li> <li>• Chemical simulations.</li> <li>• Materials science in computing.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe how chemistry is used in molecular modeling. [Familiarizarse]</li> <li>• Explain the role of chemistry in chemical simulations. [Usar]</li> <li>• Analyze the importance of chemistry in materials science for computing. [Evaluar]</li> </ul>
Readings : [BrownLeMayBursten2017]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

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



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ID101. Technical and professional English I (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : ID101. Technical and professional English I
- 2.2 Semester : 2<sup>nd</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people’s lives. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services English is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

**5. GOALS**

- Know the English language and its grammatical structure.
- Identify situations and employ dialogues related to them.

**6. COMPETENCES**

- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Hello everybody! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Verb To Be.</li><li>• Affirmative sentences , Negatives and Questions.</li><li>• Number Expressions.</li><li>• Objects and Countries.</li><li>• Expressions to greet and make presentations.</li></ul>	<ul style="list-style-type: none"><li>• At the end of the first unit, each student, understanding the grammar of the present tense is able to express a greater quantity of expressions of time and also to use sentences with the verb To Be to express situation and state.</li><li>• That the student is able to analyze and express ideas about dates and numbers in order.</li></ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 2: Meeting people! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Possessive adjectives.</li> <li>• Expressions to find prices.</li> <li>• Possession expressions</li> <li>• Vocabulary of Family, Food and Drinks.</li> <li>• Formal requests.</li> <li>• Informal letters.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the second unit, students having identified how to express orders and make offerings in restaurants use them in various situations. Explain and apply food and drink vocabulary.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 3: The world of work (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Simple present tense. Auxiliaries.</li> <li>• Affirmative sentences, Negatives and Questions.</li> <li>• Common verbs and occupations.</li> <li>• Indications for expressing the time.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the third unit, students having recognized the characteristics of the present simple, use it to make descriptions of various types. Describe people and places and give directions. Express time.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 4: Take it easy! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Present Simple 2.</li> <li>• Affirmative sentences , Negatives and Questions.</li> <li>• Use of Verbs of entertainment.</li> <li>• Free time.</li> <li>• The seasons of the year.</li> <li>• Expressions of social activities.</li> </ul>	<ul style="list-style-type: none"> <li>• At the conclusion of the fourth unit, the students having identified the idea of expressing ideas of free time actions in Simple and Continuous Present. Express ideas of stations and related activities.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 5: Where do you live? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Use There is/There are.</li> <li>• Sentences with Prepositions.</li> <li>• Expressions of Quantity.</li> <li>• Vocabulary of airplanes and places.</li> <li>• Expressions of direction indications.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fifth unit, students, based on the understanding of the present continuous time, will elaborate sentences using ideas of location and place. They will also assimilate the need to express objects in common use. They will acquire vocabulary to describe the parts of a house using expressions to ask for directions.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	



Unit 6: Can you speak English? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Can/cant.</li> <li>• Past of verb to be. Use of Could</li> <li>• Vocabulary of Countries and languages.</li> <li>• Expressions for using the phone</li> <li>• Writing formal letters.</li> <li>• Readings.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the sixth unit, students having learned the fundamentals of using auxiliary mode, will create sentences applied to the appropriate context. They emphasize the difference between languages and nationalities. They describe feelings. Use expressions on the phone.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 7: Then and now! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Past Simple.</li> <li>• Expressions of past tense.</li> <li>• Vocabulary regular and irregular verbs</li> <li>• Expressions to describe the climate.</li> <li>• Writing descriptive paragraphs.</li> <li>• Special occasions.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the seventh unit, students having learned the basics of structuring the Simple Past experience the need to be able to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between irregular and regular verbs. They describe actions with several verbs. They use expressions to describe the climate.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS112. Computer Science I (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS112. Computer Science I
- 2.2 Semester : 2<sup>nd</sup> Semester.
- 2.3 Credits : 5
- 2.4 Horas : 2 HT; 6 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS111. Introduction to Programming. (1<sup>st</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This is the second course in the sequence of introductory courses in computer science. The course will introduce students in the various topics of the area of computing such as: Algorithms, Data Structures, Software Engineering, etc.

**5. GOALS**

- Introduce the student to the foundations of the object orientation paradigm, allowing the assimilation of concepts necessary to develop information systems.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Familiarity)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Overview of Programming Languages (1 hours)	
Competences Expected: 1	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Brief review of programming paradigms.</li><li>• Comparison between functional and imperative programming.</li><li>• History of programming languages (emphasis on C and 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>
Readings : [Str13], [Jos19], [Dei17]	

**Unit 2: Máquinas virtuales (2 hours)****Competences Expected: 1,6**

Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• The concept of a virtual machine.</li><li>• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) .</li><li>• Intermediate languages.</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>Readings :</b> [Str13], [Jos19], [Dei17]	

Unit 3: Sistemas de tipos básicos (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Type systems in programming languages.</li> <li>• Declaration models (linking, visibility, scope, and lifetime).</li> <li>• Overview of type checking.</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>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

<b>Unit 4: Conceptos Fundamentales de Programación (6 hours)</b>	
<b>Competences Expected: 1,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Diseño orientado a objetos: <ul style="list-style-type: none"> <li>– Descomposicion en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquia de clases para modelamiento</li> </ul> </li> <li>• Variables and data types.</li> <li>• Expressions and operators.</li> <li>• Conditional statements.</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>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

<b>Unit 5: Functions (6 hours)</b>	
<b>Competences Expected: 1,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Paso de funciones y parámetros.</li> <li>• Parameter passing.</li> <li>• Function overloading.</li> <li>• Fundamentals of recursion.</li> <li>• Function templates.</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>• Understand and apply the concept of parameter passing to a function, both by value and by reference. [Usar]</li> <li>• Identify and apply the concept of function overloading. [Usar]</li> <li>• Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse]</li> <li>• Design, implement, and apply the concept of templates to create generic functions. [Usar]</li> </ul>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

<b>Unit 6: Arrays, Pointers, and Memory Management (8 hours)</b>	
<b>Competences Expected: 1,6</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Array definition.</li> <li>• Multidimensional arrays.</li> <li>• Pointer fundamentals.</li> <li>• Dynamic memory management (new/delete, stack vs. heap).</li> <li>• Smart pointers (unique_ptr, shared_ptr, weak_ptr).</li> <li>• Advanced pointer concepts (pointers to pointers, pointers to functions).</li> </ul>	<ul style="list-style-type: none"> <li>• Understand and implement one-dimensional arrays. [Familiarizarse]</li> <li>• Design and apply the concept of multidimensional arrays. [Usar]</li> <li>• Understand and apply the concept of references and pointers. [Familiarizarse]</li> <li>• Understand, apply, and evaluate the relationship between pointers and arrays. [Evaluar]</li> <li>• Understand and implement dynamic memory management. Differentiate between heap and stack memory regions. [Evaluar]</li> <li>• Design, implement, and evaluate concepts like pointer-to-pointer, pointer-to-function, among other advanced pointer concepts. [Evaluar]</li> </ul>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

<b>Unit 7: Working with Arrays and Pointers (5 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Arrays as function arguments.</li> <li>• Character arrays and pointers.</li> <li>• Pointers and 2-dimensional arrays.</li> <li>• Pointers and multidimensional arrays.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate the use of pointers with different types of arrays. [Usar]</li> <li>• Demonstrate the layout of an array in memory and how pointers are manipulated within those memory spaces. [Usar]</li> <li>• Demonstrate the use of pointer arithmetic with arrays. [Usar]</li> </ul>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

<b>Unit 8: Pointers and Dynamic Memory (5 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Pointers and dynamic memory - stack vs heap.</li> <li>• Pointers as return values from a function in C/C++.</li> <li>• Pointers to functions in C/C++.</li> <li>• Pointers to functions and callbacks.</li> <li>• Memory leaks in C/C++.</li> </ul>	<ul style="list-style-type: none"> <li>• Show the memory structure within a program and understand how the compiler allocates elements on the stack and heap. [Usar]</li> <li>• Demonstrate the use of functions and operators for dynamic memory allocation and deallocation. [Usar]</li> <li>• Understand the implications of returning pointers from functions. [Usar]</li> <li>• Use pointers to functions as parameters. [Usar]</li> <li>• Understand the implications of dynamic memory usage and memory leaks. [Usar]</li> </ul>
<b>Readings :</b> [Str13], [Jos19], [Dei17]	

Unit 9: Pointers and Classes (5 hours)	
Competences Expected: 1	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Pointers to class members - attributes.</li> <li>• Pointers to class members - methods and calls to method pointers.</li> <li>• Pointers to class members - static methods and calls to static method pointers.</li> <li>• Pointers to classes - example with linked list management.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the use of pointers to different elements of a class. [Usar]</li> <li>• Understand the use of pointers to static members of a class. [Usar]</li> <li>• Introduce the node structure and its use in a simple data structure. [Usar]</li> <li>• Introduce data structures, showing a simple implementation of linked lists.[Usar]</li> </ul>
Readings : [Str13], [Jos19], [Dei17]	

Unit 10: Programación orientada a objetos (8 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Diseño orientado a objetos: <ul style="list-style-type: none"> <li>– Descomposicion en objetos que almacenan estados y poseen comportamiento</li> <li>– Diseño basado en jerarquia 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 coleccion de clases, iteradores, y otros componentes de la libreria 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 procedurar/funcional- definiendo una función por cada operación con el cuerdo 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>
Readings : [Str13], [Jos19], [Dei17]	

Unit 11: Templates and STL (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Class templates.</li> <li>• Basic concepts of the Standard Template Library (STL) including: vector, list, stack, queue.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concepts of class templates. [Familiarizarse]</li> <li>• Implement and create new generic data types. [Usar]</li> <li>• Understand the basic structures of the STL. [Familiarizarse]</li> <li>• Use basic data structures like stack, queue, list, and vector from the STL. [Usar]</li> </ul>
Readings : [Str13], [Jos19], [Dei17]	

Unit 12: Operator Overloading (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Operator overloading definition.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concepts of operator overloading. [Familiarizarse]</li> <li>• Implement the overloading of allowed operators in the programming language. [Usar]</li> </ul>
Readings : [Str13], [Jos19], [Dei17]	

Unit 13: File Handling (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• File input and output (I/O).</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concepts of file manipulation. [Familiarizarse]</li> <li>• Create programs to read and write files. [Usar]</li> </ul>
Readings : [Str13], [Jos19], [Dei17]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS1D1. Discrete Structures I (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS1D1. Discrete Structures I
<b>2.2 Semester</b>	: 1 <sup>st</sup> Semester.
<b>2.3 Credits</b>	: 4
<b>2.4 Horas</b>	: 2 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Discrete structures provide the theoretical foundations necessary for computation. These fundamentals are not only useful to develop computation from a theoretical point of view as it happens in the course of computational theory, but also is useful for the practice of computing; In particular in applications such as verification, cryptography, formal methods, etc.

**5. GOALS**

- Apply Properly concepts of finite mathematics (sets, relations, functions) to represent data of real problems.
- Model real situations described in natural language, using propositional logic and predicate logic.
- Determine the abstract properties of binary relations.
- Choose the most appropriate demonstration method to determine the veracity of a proposal and construct correct mathematical arguments.
- Interpret mathematical solutions to a problem and determine their reliability, advantages and disadvantages.
- Express the operation of a simple electronic circuit using Boolean algebra.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

Unit 1: Funciones, relaciones y conjuntos (22 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Relations: <ul style="list-style-type: none"> <li>– Reflexivity, simmetry, transitivity</li> <li>– Equivalence relations</li> <li>– Partial order relations and sets</li> <li>– Extremal elements of a partially ordered sets</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>
Readings : [Gri03], [Ros07], [Vel06]	

Unit 2: Lógica básica (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Logica 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>
Readings : [Ros07], [Gri03], [Vel06]	

Unit 3: Técnicas de demostración (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 solido [Usar]</li> <li>• Determine que 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>
Readings : [Ros07], [Vel06], [Sch12], [Vel06]	

Unit 4: Data Representation (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Numerical representation: sign-magnitude, floating point.</li> <li>• Representation of other objects: sets, relations, functions.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain numerical representations such as sign-magnitude and floating point. [Evaluar].</li> <li>• Carry out arithmetic operations using different kinds of representations. [Evaluar].</li> <li>• Explain the floating point standard IEEE-754 [Familiarizarse].</li> </ul>
Readings : [Ros07], [Gri03], [Vel06]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Vel06] Daniel J. Velleman. *How to Prove It: A Structured Approach*. Ed. by Cambridge University Pres. 2nd. 2006.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. 2007.
- [Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. 3 ed. 2012.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

MA103FCCS. Differential Calculus (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: MA103FCCS. Differential Calculus
<b>2.2 Semester</b>	: 2 <sup>nd</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: MA102FCCS. Integral Calculus. (1 <sup>st</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Integral calculus is essential in computer science for modeling and solving problems involving accumulation, change, and areas under curves. This course provides the foundations of integral calculus, including integration techniques, applications, and its relationship with differential calculus.

**5. GOALS**

- Understand the concept of definite and indefinite integrals.
- Apply various integration techniques to solve problems.
- Use integral calculus to model and solve problems in scientific and engineering contexts, including applications in computing.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: The Indefinite Integral (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Antiderivatives and the indefinite integral.</li> <li>• Basic integration rules.</li> <li>• Integration by substitution.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate antiderivatives of basic functions. [Familiarizarse]</li> <li>• Apply the basic integration rules. [Usar]</li> <li>• Solve indefinite integrals using the substitution technique. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 2: The Definite Integral (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Riemann sums and the definite integral.</li> <li>• The Fundamental Theorem of Calculus.</li> <li>• Calculating areas.</li> </ul>	<ul style="list-style-type: none"> <li>• Approximate definite integrals using Riemann sums. [Familiarizarse]</li> <li>• Apply the Fundamental Theorem of Calculus to evaluate definite integrals. [Usar]</li> <li>• Calculate areas under curves using definite integrals. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 3: Techniques of Integration (12 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Integration by parts.</li> <li>• Integration of trigonometric functions.</li> <li>• Integration by partial fractions.</li> <li>• Improper integrals.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply the technique of integration by parts. [Familiarizarse]</li> <li>• Integrate trigonometric functions using identities and substitution techniques. [Usar]</li> <li>• Solve integrals using the technique of partial fractions. [Evaluar]</li> <li>• Evaluate improper integrals. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 4: Applications of the Definite Integral (12 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Calculating areas between curves.</li> <li>• Calculating volumes of solids of revolution.</li> <li>• Arc length.</li> <li>• Work, average value, and centroids.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate the area between two curves using definite integrals. [Familiarizarse]</li> <li>• Calculate the volume of solids of revolution using different methods. [Usar]</li> <li>• Calculate the arc length of a curve. [Evaluar]</li> <li>• Apply integrals to calculate work, average value, and centroids. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

Unit 5: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Algorithm analysis (e.g., calculating time complexity).</li> <li>• Signal and image processing (e.g., integral transforms).</li> <li>• Probability and statistics (e.g., probability density functions).</li> </ul>	<ul style="list-style-type: none"> <li>• Use integrals to analyze the time complexity of algorithms. [Familiarizarse]</li> <li>• Apply integral transforms in signal and image processing. [Usar]</li> <li>• Use integrals in the context of probability density functions. [Evaluar]</li> </ul>
Readings : [Ste15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

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

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



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

FI201FCCS. Computational Physics (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |   |
|-----------------------------------|---|
| <b>2.1 Course</b>                 | : FI201FCCS. Computational Physics            |
| <b>2.2 Semester</b>               | : 2 <sup>nd</sup> Semester.                   |
| <b>2.3 Credits</b>                | : 3   |
| <b>2.4 Horas</b>                  | : 2 HT; 2 HP;                                 |
| <b>2.5 Duration of the period</b> | : 16 weeks                                    |
| <b>2.6 Type of course</b>         | : Mandatory                                   |
| <b>2.7 Learning modality</b>      | : Face to face                                |
| <b>2.8 Prerequisites</b>          | : FI101FCCS. Physics I. (1 <sup>st</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course applies the principles of physics to computational problems, with an emphasis on light, wave propagation, collisions, and energy transfer. These concepts are essential in areas such as computer graphics, physical simulations, and video game development.

**5. GOALS**

- Understand the physical principles relevant to computing.
- Apply these principles to solve specific computational problems.
- Implement physics-based algorithms for simulations and computer graphics.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C09** Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)
- AG-C12** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**



<b>Unit 1: Optics and Light Propagation (10 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Nature of light.</li> <li>• Reflection and refraction.</li> <li>• Lenses and mirrors.</li> <li>• Interference and diffraction.</li> <li>• Illumination models (e.g., Phong, Blinn-Phong).</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the properties of light and its propagation. [Familiarizarse]</li> <li>• Apply the laws of reflection and refraction. [Usar]</li> <li>• Implement illumination models in computer graphics. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [Hec17]	

<b>Unit 2: Collisions and Energy Transfer (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Impulse and linear momentum.</li> <li>• Elastic and inelastic collisions.</li> <li>• Conservation of energy in collisions.</li> <li>• Deformation of elastic meshes (e.g., mass-spring model).</li> </ul>	<ul style="list-style-type: none"> <li>• Apply the principles of conservation of linear momentum and energy in collisions. [Familiarizarse]</li> <li>• Model the deformation of elastic meshes due to impact. [Usar]</li> <li>• Implement collision simulations in a computational environment. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [Tay05]	

<b>Unit 3: Rigid Body Mechanics (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Rotation of rigid bodies.</li> <li>• Moment of inertia.</li> <li>• Torque and rotational kinetic energy.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the rotation of rigid bodies. [Familiarizarse]</li> <li>• Calculate the moment of inertia. [Usar]</li> <li>• Apply the equations of rotational dynamics. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18], [Tay05]	

<b>Unit 4: Fluid Dynamics (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Basic principles of fluid dynamics.</li> <li>• Viscosity.</li> <li>• Laminar and turbulent flow.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the properties of fluids. [Familiarizarse]</li> <li>• Explain the concepts of viscosity and laminar/turbulent flow. [Usar]</li> <li>• Solve simple fluid dynamics problems. [Evaluar]</li> </ul>
<b>Readings :</b> [YF18]	

Unit 5: Thermodynamics (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Laws of thermodynamics.</li> <li>• Heat transfer.</li> </ul>	<ul style="list-style-type: none"> <li>• State the laws of thermodynamics. [Familiarizarse]</li> <li>• Describe the mechanisms of heat transfer. [Usar]</li> </ul>
Readings : [YF18]	

Unit 6: Physical Simulation (10 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Numerical methods for physical simulation.</li> <li>• Verlet integration.</li> <li>• Collision detection.</li> <li>• Particle systems.</li> <li>• Constraints and solvers.</li> </ul>	<ul style="list-style-type: none"> <li>• Implement basic numerical methods for physical simulation. [Familiarizarse]</li> <li>• Use Verlet integration to simulate motion. [Usar]</li> <li>• Implement collision detection algorithms. [Evaluar]</li> </ul>
Readings : [Tay05]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

BI101FCCS. Biology I (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: BI101FCCS. Biology I
<b>2.2 Semester</b>	: 2 <sup>nd</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Biology is the science of life, and while it may not seem immediately relevant to computer science, it provides a foundation for interdisciplinary fields such as bioinformatics, computational biology, and biologically inspired artificial intelligence. This course introduces the fundamental concepts of biology, from the cell to evolution.

**5. GOALS**

- Understand the basic principles of cellular and molecular biology.
- Learn the fundamental processes of life, such as DNA replication, transcription, and translation.
- Appreciate the interrelationship between biology and computing in areas like bioinformatics.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- AG-C01)** The Professional and the World: Analyzes and evaluates the impact of solutions to complex computing problems on the sustainable development of society. (Usage)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Usage)

**7. TOPICS**

<b>Unit 1: Introduction to Biology (4 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Characteristics of living organisms.</li> <li>• Levels of biological organization.</li> <li>• The scientific method in biology.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the characteristics of living organisms. [Familiarizarse]</li> <li>• Identify the different levels of biological organization. [Usar]</li> <li>• Apply the scientific method in the context of biology. [Evaluar]</li> </ul>
<b>Readings :</b> [al17], [Rav+17]	

<b>Unit 2: The Cell (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Cell structure: prokaryotes and eukaryotes.</li> <li>• Cell membranes and transport.</li> <li>• Cell metabolism: respiration and photosynthesis.</li> </ul>	<ul style="list-style-type: none"> <li>• Compare and contrast prokaryotic and eukaryotic cells. [Familiarizarse]</li> <li>• Describe the mechanisms of transport across cell membranes. [Usar]</li> <li>• Explain the processes of respiration and photosynthesis. [Evaluar]</li> </ul>
<b>Readings :</b> [al17], [Rav+17]	

<b>Unit 3: Molecular Genetics (12 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• DNA: structure and replication.</li> <li>• RNA: transcription and translation.</li> <li>• Gene regulation.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the structure and replication of DNA. [Familiarizarse]</li> <li>• Explain the processes of transcription and translation. [Usar]</li> <li>• Understand the mechanisms of gene regulation. [Evaluar]</li> </ul>
<b>Readings :</b> [al17], [Rav+17]	

<b>Unit 4: Evolution (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Theory of evolution by natural selection.</li> <li>• Evidence of evolution.</li> <li>• Mechanisms of evolution.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain the theory of evolution by natural selection. [Familiarizarse]</li> <li>• Describe the evidence of evolution. [Usar]</li> <li>• Analyze the different mechanisms of evolution. [Evaluar]</li> </ul>
<b>Readings :</b> [al17], [Rav+17]	

Unit 5: Biology and Computing (8 hours)	
Competences Expected: 1,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Bioinformatics: analysis of biological sequences.</li> <li>• Computational biology: modeling and simulation of biological systems.</li> <li>• Biologically inspired algorithms.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe applications of bioinformatics. [Familiarizarse]</li> <li>• Explain how computation is used in biology. [Usar]</li> <li>• Analyze examples of biologically inspired algorithms. [Evaluar]</li> </ul>
Readings : [al17]	

Unit 6: Ecology and Biodiversity (8 hours)	
Competences Expected: 1,AG-C01,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Ecological interactions.</li> <li>• Biodiversity and conservation.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the interactions between organisms and their environment. [Familiarizarse]</li> <li>• Explain the importance of biodiversity and conservation. [Usar]</li> <li>• Analyze the impact of human activities on biodiversity. [Evaluar]</li> </ul>
Readings : [al17], [Rav+17]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[al17] Lisa A. Urry et al. *Campbell Biology*. Pearson, 2017.

[Rav+17] Peter H. Raven et al. *Biology*. McGraw-Hill Education, 2017.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ID102. Technical and professional English II (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : ID102. Technical and professional English II                       |
| <b>2.2 Semester</b>               | : 3 <sup>rd</sup> Semester.  |
| <b>2.3 Credits</b>                | : 4  |
| <b>2.4 Horas</b>                  | : 2 HT; 4 HP;  |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face   |
| <b>2.8 Prerequisites</b>          | : ID101. Technical and professional English I. (2 <sup>nd</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people's lives. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services english is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

**5. GOALS**

- Develop the ability to fluently speak the language..
- Increase vocabulary and handle simple expressions

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)

**7. TOPICS**

Unit 1: How long ago? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Simple past</li> <li>• Negative sentences with ago.</li> <li>• Conjunctions</li> <li>• Expressions of Time in past</li> <li>• Phonetic relations and symbols</li> <li>• Expressions to give the date</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the eighth unit, each student, understanding the grammar of the past tense is able to express a greater number of expressions of time and also to use prepositions to describe varied places and times. It is also capable of analyzing and expressing ideas about dates and numbers in order.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

Unit 2: Food you like! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Accounting and Non-Accounting Nouns</li> <li>• Expressions with Would like and I'd like</li> <li>• Quantifiers</li> <li>• Meals around the world</li> <li>• Formal requests</li> <li>• Formal letters</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the ninth unit, students having identified how to express orders and make collections, uses in various situations. Express situations and states related to quantities. Explain and apply food and drinks vocabulary.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

Unit 3: The world of work (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Adjectives</li> <li>• Sentences with Comparative Adjectives</li> <li>• Sentences with Superlative Adjectives</li> <li>• Cities and countryside</li> <li>• Directional indications</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the tenth unit, students who have recognized the characteristics of adjectives use these to make comparisons of various types. Describes people and places and directions. They will use conjunctions to unite type ideas.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

Unit 4: Looking good! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Present continuous</li> <li>• Affirmative sentences, Negatives and Questions</li> <li>• Use of Whose</li> <li>• Possessive pronouns</li> <li>• Clothing and colors</li> <li>• Expressions to wear in clothing stores</li> <li>• Phonetic symbols.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the eleventh unit, students having identified the idea of expressing ideas of actions that occur at the time or that relate at any time structure sentences in Present Progressive. They express ideas of possession with regard to clothes and colors.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

Unit 5: Life is an adventure! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Use of going to</li> <li>• Future time sentences</li> <li>• Expressions of Quantity.</li> <li>• Action verbs</li> <li>• Vocabulary of the climate</li> <li>• Expressions of Suggestion</li> <li>• Write a postcard</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the twelfth unit, students, from the understanding of future time, will elaborate sentences using the necessary elements. They will also assimilate the need to express purposeful infinitives. They will acquire vocabulary to describe the climate. Expressions will be presented to make and ask for suggestions.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

Unit 6: You're pretty smart! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Forms of Questions</li> <li>• Adverbs and Adjectives</li> <li>• Vocabulary description of feelings</li> <li>• Expressions for train travel</li> <li>• Writing Short Stories</li> <li>• Readings</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the thirteenth unit, students having learned the fundamentals of structuring various questions, will perform application work in appropriate contexts. They emphasize the difference between adjectives and adverbs. They describe feelings. They use expressions to catch a train. They assume the idea is suffixes and prefixes.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	



Unit 7: Have you ever? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Perfect present</li> <li>• Keywords with never, ever, and yet</li> <li>• Vocabulary verbs in Past participle</li> <li>• Expressions for airplane travel</li> <li>• Writing thank-you letters</li> <li>• Readings</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fourteenth unit, students having known the fundamentals of the structure of the Present Perfect experience the need to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between simple past and perfect present. Describe actions with never, ever, and yet. They use expressions to use at an airport.</li> </ul>
Readings : [Soars021S], [Cambridge06], [MacGrew99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS113. Computer Science II (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : CS113. Computer Science II                       |
| <b>2.2 Semester</b>               | : 3 <sup>rd</sup> Semester.                        |
| <b>2.3 Credits</b>                | : 4  |
| <b>2.4 Horas</b>                  | : 2 HT; 4 HP;                                      |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face                                     |
| <b>2.8 Prerequisites</b>          | : CS112. Computer Science I. (2 <sup>nd</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This is the third course in the sequence of introductory courses in computer science. This course is intended to cover Concepts indicated by the Computing Curriculum IEEE (c) -ACM 2001, under the functional-first approach. The object-oriented paradigm allows us to combat complexity by making models from abstractions of the problem elements and using techniques such as encapsulation, modularity, polymorphism and inheritance. The Dominion of these topics will enable participants to provide computational solutions to design problems simple of the real world.

**5. GOALS**

- Introduce the student in the fundamentals of the paradigm of object orientation, allowing the assimilation of concepts necessary to develop an information system

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)

**7. TOPICS**

<b>Unit 1: Advanced STL (8 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Associative Containers (std::set, std::map, std::unordered_set, std::unordered_map).</li> <li>• Adapters (std::stack, std::queue, std::priority_queue).</li> <li>• Advanced STL Algorithms.</li> <li>• Functors and Predicates.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the use of associative containers. [Usar]</li> <li>• Implement programs that use STL adapters. [Usar]</li> <li>• Apply advanced STL algorithms. [Usar]</li> <li>• Use functors and predicates with the STL. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

<b>Unit 2: Advanced Templates (7 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Template Metaprogramming.</li> <li>• SFINAE (Substitution Failure Is Not An Error).</li> <li>• Perfect Forwarding.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply template metaprogramming to solve complex problems. [Usar]</li> <li>• Understand and use SFINAE for template selection. [Usar]</li> <li>• Use Perfect Forwarding for efficient argument passing. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

<b>Unit 3: Variadic Templates (12 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Introduction to Variadic Templates.</li> <li>• Variadic Template Functions.</li> <li>• Variadic Template Methods.</li> <li>• Variadic Template Classes.</li> <li>• Classes inheriting from variable lists of variadic templates.</li> <li>• Example: Implementing a custom tuple class.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concept of variadic templates. [Familiarizarse]</li> <li>• Implement variadic functions. [Usar]</li> <li>• Design classes with variadic methods. [Usar]</li> <li>• Create classes with variadic templates. [Usar]</li> <li>• Implement inheritance with variable lists of variadic templates. [Usar]</li> <li>• Apply variadic templates to solve real-world problems. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

<b>Unit 4: Move Semantics and Rvalue References (5 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Lvalues and Rvalues.</li> <li>• Rvalue References.</li> <li>• Move Semantics.</li> <li>• Move Constructors and Move Assignment Operators.</li> <li>• Perfect Forwarding (review).</li> </ul>	<ul style="list-style-type: none"> <li>• Explain move semantics and its purpose in C++. [Familiarizarse]</li> <li>• Define and use rvalue references. [Usar]</li> <li>• Analyze the performance implications of using move semantics. [Evaluar]</li> <li>• Implement move constructors and move assignment operators for custom classes. [Usar]</li> <li>• Apply move semantics to optimize resource management in C++ programs. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

<b>Unit 5: Design Patterns (Creational and Structural) (6 hours)</b>	
<b>Competences Expected: 1</b>	
<b>Topics</b>	<b>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>• Understand and apply creational design patterns: Singleton, Factory, Builder. [Usar]</li> <li>• Understand and apply structural design patterns: Adapter, Decorator, Facade. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

<b>Unit 6: Functors (3 hours)</b>	
<b>Competences Expected: 1,3</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Definition of Functors.</li> <li>• Functors and Templates.</li> <li>• Passing Functors to Functions using parameters.</li> <li>• Passing Functors to Functions using templates.</li> <li>• Passing Functors to Classes using parameters.</li> <li>• Passing Functors to Classes using templates.</li> <li>• Examples and Applications.</li> </ul>	<ul style="list-style-type: none"> <li>• Introduction to Functors. [Usar]</li> <li>• Using Functors as parameters to functions and classes. [Usar]</li> <li>• Using Functors in functions and classes through templates. [Usar]</li> </ul>
<b>Readings :</b> [Stroustrup2013], [MJo19], [Deitel17]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[MJo19] Nicolai M.Josuttis. *C++17-The Complete Guide*. 1st. 2019.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS1D2. Discrete Structures II (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	CS1D2. Discrete Structures II
<b>2.2 Semester</b>	:	2 <sup>nd</sup> Semester.
<b>2.3 Credits</b>	:	4
<b>2.4 Horas</b>	:	2 HT; 4 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	CS1D1. Discrete Structures I. (1 <sup>st</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

**5. GOALS**

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student applies efficient travel strategies to be able to search data in an optimal way.
- That the student uses the various counting techniques to solve computational problems.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

**7. TOPICS**

**Unit 1: Digital Logic and Data Representation (10 hours)****Competences Expected:**

<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"><li>• Reticles: Types and properties.</li><li>• Boolean algebras.</li><li>• Boolean Functions and Expressions.</li><li>• Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form.</li><li>• Logical gates.</li><li>• Circuit Minimization.</li></ul>	<ul style="list-style-type: none"><li>• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Evaluar].</li><li>• Explain the algebraic structures of reticulum and its types [Evaluar].</li><li>• Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Evaluar].</li><li>• Explain the properties that satisfies a Boolean algebra [Evaluar].</li><li>• Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Evaluar].</li><li>• Find the canonical forms of a Boolean function [Evaluar].</li><li>• Represent a Boolean function as a Boolean circuit using logic gates [Evaluar].</li><li>• Minimize a Boolean function. [Evaluar].</li></ul>
<b>Readings :</b> [Ros07], [Gri03]	

**Unit 2: Fundamentos de conteo (40 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>Readings :</b> [Gri97]	



Unit 3: Árboles y Grafos (40 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Arboles de expansion/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>• Demostrar 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>
Readings : [Joh99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.
- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS2B1. Platform Based Development (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : CS2B1. Platform Based Development                |
| <b>2.2 Semester</b>               | : 3 <sup>rd</sup> Semester.                        |
| <b>2.3 Credits</b>                | : 3  |
| <b>2.4 Horas</b>                  | : 1 HT; 4 HP;                                      |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face                                     |
| <b>2.8 Prerequisites</b>          | : CS112. Computer Science I. (2 <sup>nd</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The world has changed due to the use of fabric and related technologies, rapid, timely and personalized access to the information, through web technology, ubiquitous and pervasive; they have changed the way we do things, how do we think? and how does the industry develop? Web technologies, ubiquitous and pervasive are based on the development of web services, web applications and mobile applications, which are necessary to understand the architecture, design, and implementation of web services, web applications and mobile applications.

**5. GOALS**

- That the student is able to design and implement services, web applications using tools and languages such as HTML, CSS, JavaScript (including AJAX), back-end scripting and a database, at an intermediate level.
- That the student is able to develop mobile applications, administration of web servers in a Unix system and an introduction to web security, at an intermediate level.

**6. COMPETENCES**

- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

**7. TOPICS**

<b>Unit 1: Introducción (5 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>Readings :</b> [fielding2000fielding], [grove2009web], [annuzzi2013introduction], [Cornez2015]	

<b>Unit 2: Plataformas web (5 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Lenguajes de programación web (e.g., HTML5, Javascript, PHP, CSS)</li> <li>•</li> <li>• Web Platform constraints: Client-Server, Stateless-Stateful, Cache, 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>• Revisar una aplicación web existente con un estándar web actual [Familiarizarse]</li> </ul>
<b>Readings :</b> [fielding2000fielding]	

Unit 3: Desarrollo de servicios y aplicaciones web (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Describe, identify and debug issues related to web application development</li> <li>• Design and development of interactive web applications using HTML5 and Python</li> <li>• Use MySQL for data management and manipulate MySQL with Python</li> <li>• Design and development of asynchronous web applications using Ajax techniques</li> <li>• Using dynamic client side Javascript scripting language and server side python scripting language with Ajax</li> <li>• Apply XML / JSON technologies for data management with Ajax</li> <li>• Use framework, services and Ajax web APIs and apply design patterns to web application development</li> </ul>	<ul style="list-style-type: none"> <li>• Server-side python scripting language: variables, data types, operations, strings, functions, control statements, arrays, files and directory access, maintain state. [Usar]</li> <li>• Web programming approach using embedded python. [Usar]</li> <li>• Accessing and Manipulating MySQL. [Usar]</li> <li>• The Ajax web application development approach. [Usar]</li> <li>• DOM and CSS used in JavaScript. [Usar]</li> <li>• Asynchronous Content Update Technologies. [Usar]</li> <li>• XMLHttpRequest objects use to communicate between clients and servers. [Usar]</li> <li>• XML and JSON. [Usar]</li> <li>• XSLT and XPath as mechanisms for transforming XML documents. [Usar]</li> <li>• Web services and APIs (especially Google Maps). [Usar]</li> <li>• Macros Ajax for the development of contemporary web applications. [Usar]</li> <li>• Design patterns used in web applications. [Usar]</li> </ul>
Readings : [freeman2011head]	

Unit 4: Plataformas móviles (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Lenguajes de Programación para Móviles.</li> <li>• Design Principles: Segregation of Interfaces, Single Responsibility, Separation of concerns, Dependency Inversion.</li> <li>• Desafíos con movilidad 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 el rendimiento vs pérdida de potencia [Familiarizarse]</li> <li>• Compare y contraste la programación móvil con la programación de proposito general [Familiarizarse]</li> </ul>
Readings : [martin2017clean], [annuzzi2013introduction]	

Unit 5: Mobile Applications for Android Handheld Systems (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Students identify necessary software and install it on their personal computers.</li> <li>• Students perform various tasks to familiarize themselves with the Android platform and Environment for development. [Usar]</li> <li>• Students build applications that trace the lifecycle callback methods emitted by the Android platform and demonstrate the behavior of Android when device configuration changes (for example, when the device moves from vertical to horizontal and vice versa ). [Usar]</li> <li>• Students build applications that require starting multiple activities through both standard and custom methods. [Usar]</li> <li>• Students build applications that require standard and custom permissions. [Usar]</li> <li>• Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usar]</li> <li>• Students construct a to-do list manager using the user interface elements discussed in class. The application allows users to create new items and to display them in a ListView. [Usar]</li> <li>• Students build an application that uses location information to collect latitude, length of places they visit. [Usar]</li> </ul>
Readings : [annuzzi2013introduction], [Cornez2015]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

MA104FCCS. Advanced Differential and Integral Calculus (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: MA104FCCS. Advanced Differential and Integral Calculus
<b>2.2 Semester</b>	: 3 <sup>rd</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: MA103FCCS. Differential Calculus. (2 <sup>nd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course extends the concepts of differential and integral calculus to functions of several variables. Topics covered include partial derivatives, multiple integrals, and their applications in computing.

**5. GOALS**

- Understand and apply the concept of partial derivatives.
- Calculate multiple integrals and apply them to problem-solving.
- Apply multivariable calculus concepts in computational contexts.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07**) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12**) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Functions of Several Variables (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Functions of two or more variables.</li> <li>• Graphs and level curves.</li> <li>• Limits and continuity.</li> </ul>	<ul style="list-style-type: none"> <li>• Represent functions of several variables graphically. [Familiarizarse]</li> <li>• Evaluate limits and determine the continuity of multivariable functions. [Usar]</li> <li>• Interpret level curves. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 2: Partial Derivatives (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Partial derivatives.</li> <li>• Geometric interpretation.</li> <li>• Higher-order derivatives.</li> <li>• Chain rule.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate partial derivatives of multivariable functions. [Familiarizarse]</li> <li>• Interpret partial derivatives geometrically. [Usar]</li> <li>• Apply the chain rule for partial derivatives. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 3: Applications of Partial Derivatives (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07,AG-C12</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Tangent plane and linear approximation.</li> <li>• Extreme values.</li> <li>• Lagrange multipliers.</li> </ul>	<ul style="list-style-type: none"> <li>• Find the tangent plane to a surface. [Familiarizarse]</li> <li>• Determine the extreme values of a multivariable function. [Usar]</li> <li>• Apply Lagrange multipliers for constrained optimization. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

<b>Unit 4: Multiple Integrals (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Double integrals over rectangles.</li> <li>• Double integrals over general regions.</li> <li>• Triple integrals.</li> <li>• Change of variables in multiple integrals.</li> </ul>	<ul style="list-style-type: none"> <li>• Calculate double integrals over rectangular regions. [Familiarizarse]</li> <li>• Calculate double integrals over general regions. [Usar]</li> <li>• Evaluate triple integrals. [Evaluar]</li> <li>• Apply changes of variables in multiple integrals. [Evaluar]</li> </ul>
<b>Readings :</b> [Ste15], [LE14]	

Unit 5: Applications in Computing (8 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• 3D graphics and geometric modeling.</li> <li>• Computer vision (e.g., image processing).</li> <li>• Machine learning (e.g., cost function optimization).</li> </ul>	<ul style="list-style-type: none"> <li>• Apply multivariable calculus to the representation of 3D objects. [Familiarizarse]</li> <li>• Use multiple integrals in image processing. [Usar]</li> <li>• Apply partial derivatives in machine learning algorithms. [Evaluar]</li> </ul>
Readings : [Ste15]	

Unit 6: Vector Calculus (8 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Vector fields.</li> <li>• Line integrals.</li> <li>• Surface integrals.</li> <li>• Green's Theorem, Stokes' Theorem, and the Divergence Theorem.</li> </ul>	<ul style="list-style-type: none"> <li>• Visualize and interpret vector fields. [Familiarizarse]</li> <li>• Evaluate line integrals. [Usar]</li> <li>• Calculate surface integrals. [Evaluar]</li> <li>• Apply Green's Theorem, Stokes' Theorem, and the Divergence Theorem. [Evaluar]</li> </ul>
Readings : [Ste15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

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

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





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

EC101FCCS. General Economy (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : EC101FCCS. General Economy
- 2.2 Semester : 3<sup>rd</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 2 HT; 2 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Economics is the study of how societies allocate scarce resources. For computer science students, understanding economic principles is valuable for decision-making in business contexts, evaluating technology projects, and understanding the economic impact of innovations. This course provides an introduction to microeconomic and macroeconomic principles.

**5. GOALS**

- Understand the fundamental principles of microeconomics and macroeconomics.
- Analyze supply, demand, and price formation in markets.
- Evaluate economic policies and their impact on society.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C09)** Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Familiarity)

**7. TOPICS**

<b>Unit 1: Introduction to Economics (6 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• What is economics?</li> <li>• Basic principles of economics.</li> <li>• Microeconomics vs. Macroeconomics.</li> <li>• Economic thinking.</li> </ul>	<ul style="list-style-type: none"> <li>• Define the basic concepts of economics. [Familiarizarse]</li> <li>• Differentiate between microeconomics and macroeconomics. [Usar]</li> <li>• Apply economic thinking to analyze everyday situations. [Evaluar]</li> </ul>
<b>Readings :</b> [Man18], [SN10]	

<b>Unit 2: Microeconomics: Supply and Demand (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• The demand curve.</li> <li>• The supply curve.</li> <li>• Market equilibrium.</li> <li>• Elasticity.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain the law of demand and supply. [Familiarizarse]</li> <li>• Determine market equilibrium. [Usar]</li> <li>• Calculate and interpret the price elasticity of demand. [Evaluar]</li> </ul>
<b>Readings :</b> [Man18], [SN10]	

<b>Unit 3: Microeconomics: Firms and Markets (10 hours)</b>	
<b>Competences Expected: 1,AG-C07,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Production costs.</li> <li>• Perfect competition.</li> <li>• Monopoly.</li> <li>• Oligopoly.</li> </ul>	<ul style="list-style-type: none"> <li>• Analyze the production costs of a firm. [Familiarizarse]</li> <li>• Describe the characteristics of different market types. [Usar]</li> <li>• Compare and contrast perfect competition with monopoly and oligopoly. [Evaluar]</li> </ul>
<b>Readings :</b> [Man18], [SN10]	

<b>Unit 4: Macroeconomics: Measuring Economic Activity (8 hours)</b>	
<b>Competences Expected: 1,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Gross Domestic Product (GDP).</li> <li>• Inflation and unemployment.</li> </ul>	<ul style="list-style-type: none"> <li>• Define and calculate GDP. [Familiarizarse]</li> <li>• Explain how inflation and unemployment are measured. [Usar]</li> <li>• Analyze the relationship between GDP, inflation, and unemployment. [Evaluar]</li> </ul>
<b>Readings :</b> [Man18], [SN10]	

Unit 5: Macroeconomics: Monetary and Fiscal Policy (8 hours)	
Competences Expected: 1,AG-C07,AG-C09	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Monetary policy.</li> <li>• Fiscal policy.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain how monetary policy works. [Familiarizarse]</li> <li>• Describe the instruments of fiscal policy. [Usar]</li> <li>• Analyze the impact of monetary and fiscal policies on the economy. [Evaluar]</li> </ul>
Readings : [Man18], [SN10]	

Unit 6: Economics of Innovation and Technology (8 hours)	
Competences Expected: 1,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• The role of innovation in economic growth.</li> <li>• Economic impact of new technologies.</li> <li>• Digital markets.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain how innovation drives economic growth. [Familiarizarse]</li> <li>• Analyze the economic impact of new technologies. [Usar]</li> <li>• Understand the characteristics of digital markets. [Evaluar]</li> </ul>
Readings : [SN10]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[SN10] Paul A. Samuelson and William D. Nordhaus. *Economics*. McGraw-Hill Education, 2010.

[Man18] N. Gregory Mankiw. *Principles of Economics*. Cengage Learning, 2018.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ID103. Technical and professional English III (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : ID103. Technical and professional English III
- 2.2 Semester : 4<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : ID102. Technical and professional English II. (3<sup>rd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, undoubtedly English is the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services, English is perhaps the most important practical tool that the student must master from the outset as part of his / her integral education

**5. GOALS**

- Train the student to understand and hold a conversation.
- Provide techniques of llation of ideas .

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)

**7. TOPICS**

Unit 1: Getting to know you! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Present, Past, and Future Times.</li><li>• Interrogative sentences with Wh-</li><li>• Words with more than one meaning.</li><li>• Parts of the sentence</li><li>• Expressions for free time</li></ul>	<ul style="list-style-type: none"><li>• At the end of the first unit, each of the students, understanding the grammar of present, past and future times, is able to express a greater number of actions in the form of sentences. He is also able to express ideas in the form of questions. Assume the idea of words with more than one meaning. Use social expressions in entertainment situations.</li></ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 2: The way we live! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Simple present tense.</li> <li>• Present Continuous Time.</li> <li>• Collocations.</li> <li>• Vocabulary of the countries of the world.</li> <li>• Expressions of anger.</li> <li>• Connectors.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the second unit, students having identified the present form of expression recognize the difference between the forms of the same and apply it properly. They describe the countries accurately. They take expressions to show interest. Use connectors to join various ideas.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 3: It all went wrong! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Past simple tense.</li> <li>• Continuous past tense.</li> <li>• Irregular Verbs.</li> <li>• Time expressions.</li> <li>• Connectors of time.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the third unit, students having recognized the characteristics of past times use them properly. They use prefixes and suffixes to create and recognize new words. They describe time in a broad way. They will use conjunctions to unite type ideas.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 4: Let's go shopping! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Expressions of Indefinite Quantity.</li> <li>• Affirmative sentences, Negatives and Questions.</li> <li>• Use of Articles.</li> <li>• Product prices.</li> <li>• Filling of formats and surveys</li> <li>• Expressions for shopping</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fourth unit, students having identified the idea of quantity express different situations that involve it. Recognize and apply articles to nouns. They assume the idea of shopping with the help of expressions. They express money prices and ideas. They fill several formats. They express attitudes.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 5: What do you want to do? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Verbal Patterns I.</li> <li>• Future Intentions.</li> <li>• Verbs of Perception.</li> <li>• Vocabulary of feelings.</li> <li>• Expressions of Plans and Ambitions.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fifth unit, students, from the understanding of the idea of verbal patterns, will elaborate sentences using the necessary elements. They will also assimilate the need to express future intentions. They will acquire vocabulary to describe feelings. Expressions will be presented to describe plans and ambitions.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 6: The best in the world! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• What's it like?.</li> <li>• Adjectives</li> <li>• Comparative and superlative.</li> <li>• Synonyms and antonyms.</li> <li>• Indications of direction .</li> <li>• Readings.</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the sixth unit, students having known the fundamentals of using adjectives, structure sentences with different forms of adjectives in appropriate contexts. They emphasize the difference between types of cities and towns and lifestyles. They use expressions indicating directions.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 7: Fame! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Present Perfect and Simple Past</li> <li>• Expressions for, ever, since</li> <li>• Adverbs</li> <li>• Expressions that come in pairs</li> <li>• Short answers</li> <li>• Celebrities</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the seventh unit, students have learned the fundamentals of structuring the present perfect time and differentiate it from the simple past. They emphasize the difference between forms of adjectives. Describe ideas of music. They use expressions to give short answers. They assume the idea of giving extra explanations of the elements of a sentence.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

**9. EVALUATION SYSTEM**

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

**10. BASIC BIBLIOGRAPHY**



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS210. Algorithms and Data Structures (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS210. Algorithms and Data Structures
<b>2.2 Semester</b>	: 4 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 4
<b>2.4 Horas</b>	: 2 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: CS113. Computer Science II. (3 <sup>rd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The theoretical foundation of all branches of computing rests on algorithms and data structures, this course will provide participants with an introduction to these topics, thus forming a basis that will serve for the following courses in the career.

**5. GOALS**

- Make the student understand the importance of algorithms for solving problems.
  
- Introduce the student to the field of application of data structures.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
  
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
  
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**



<b>Unit 1: Graphs (12 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Graph Concept</li> <li>• Directed Graphs and Non-directed Graphs.</li> <li>• Using Graphs.</li> <li>• Measurement of efficiency ,in time and space.</li> <li>• Adjacency matrices.</li> <li>• Tag adjacent matrices.</li> <li>• Adjacency Lists.</li> <li>• Implementation of graphs using adjacency matrices.</li> <li>• Graph Implementation using adjacency lists</li> <li>• Insertion, search and deletion of nodes and edges.</li> <li>• Graph search algorithms.</li> </ul>	<ul style="list-style-type: none"> <li>• Acquire Dexterity to Perform Correct Implementation. [Usar]</li> <li>• Develop knowledge to decide when it is better to use one implementation technique than another. [Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

<b>Unit 2: Scatter Matrices (8 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Initial concepts.</li> <li>• Dense Matrices</li> <li>• Measurement of Efficiency in Time and Space</li> <li>• Static scatter vs. dynamic matrix creation.</li> <li>• Insert, search, and delete methods.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the use and implementation of scatter matrices.[Evaluuar]</li> </ul>
<b>Readings :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

<b>Unit 3: Balanced Trees (16 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• AVL Trees.</li> <li>• Measurement of Efficiency.</li> <li>• Simple and Composite Rotations</li> <li>• Insertion, deletion and search.</li> <li>• Trees B , B+ B* y Patricia.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the basic functions of these complex structures in order to acquire the capacity for their implementation. [Evaluuar]</li> </ul>
<b>Readings :</b> [Cor+09], [Fag+14], [Knu97], [Knu98]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

## 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

## 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS211. Theory of Computation (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : CS211. Theory of Computation                         |
| <b>2.2 Semester</b>               | : 4 <sup>th</sup> Semester.                            |
| <b>2.3 Credits</b>                | : 4  |
| <b>2.4 Horas</b>                  | : 2 HT; 4 HP;  |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face   |
| <b>2.8 Prerequisites</b>          | : CS1D2. Discrete Structures II. (2 <sup>nd</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course emphasizes formal languages, computer models and computability, as well as the fundamentals of computational complexity and complete NP problems.

**5. GOALS**

- That the student learn the fundamental concepts of the theory of formal languages.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Computabilidad y complejidad básica de autómatas (20 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Máquinas de estado finito.</li> <li>• Expresiones regulares.</li> <li>• Problema de la parada.</li> <li>• Gramáticas libres de contexto.</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>• 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>• Explique porque el problema de la parada no tiene solución algorítmica [Evaluar]</li> <li>• Diseñe una gramática libre de contexto para representar un lenguaje especificado [Evaluar]</li> <li>• Defina las clases P y NP [Evaluar]</li> <li>• Explique el significado de NP-Complejidad [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 [Familiarizarse]</li> </ul>
<b>Readings :</b> [Mar10], [Lin11], [Sip12]	

<b>Unit 2: Complejidad Computacional Avanzada (20 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>• Defina la clase P-Space y su relación con la clase EXP [Evaluar]</li> <li>• Defina 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>
<b>Readings :</b> [Mar10], [Lin11], [Sip12], [HU13]	

Unit 3: Teoría y Computabilidad Avanzada de Autómatas (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• 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> </ul>
Readings : [HU13], [Bro93]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.
- [Mar10] John Martin. *Introduction to Languages and the Theory of Computation*. 4th. McGraw-Hill, 2010.
- [Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones & Bartlett Learning, 2011.
- [Sip12] Michael Sipser. *Introduction to the Theory of Computation*. 3rd. Cengage Learning, 2012.
- [HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS221. Computer Systems Architecture (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	CS221. Computer Systems Architecture
<b>2.2 Semester</b>	:	3 <sup>rd</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	CS1D2. Discrete Structures II. (2 <sup>nd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A computer scientist must have a solid knowledge of the organization and design principles of diverse computer systems, by understanding the limitations of modern systems they could propose next-gen paradigms. This course teaches the basics and principles of Computer Architecture. This class addresses digital logic design, basics of Computer Architecture and processor design (Instruction Set architecture, microarchitecture, out-of-order execution, branch prediction), execution paradigms (superscalar, dataflow, VLIW, SIMD, GPUs, systolic, multithreading) and memory system organization.

**5. GOALS**

- Provide a first approach in Computer Architecture.
- Study the design and evolution of computer architectures, which lead to modern approaches and implementations in computing systems.
- Provide fine-grained details of computer hardware, and its relation with software execution.
- Implement a simple microprocessor using Verilog language.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

Unit 1: Lógica digital y sistemas digitales (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Múltiples 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 language 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 [Usar]</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 [Usar]</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, sintetizar, y simular bloques de construcción (como ALUs, registros, movimiento entre registros) de un computador simple [Familiarizarse]</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>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

**Unit 2: Representación de datos a nivel máquina (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Bits, Bytes y Words.</li><li>• Representación de datos numérica y bases numéricas.</li><li>• Sistemas de punto flotante y punto fijo.</li><li>• Representaciones con signo y complemento a 2.</li><li>• Representación de información no numérica (códigos de caracteres, información gráfica)</li><li>• Representación de registros y arreglos.</li></ul>	<ul style="list-style-type: none"><li>• Explicar porqué en computación todo es datos, inclusive las instrucciones [Evaluar]</li><li>• Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse]</li><li>• Describir cómo los enteros negativos se almacenan con representaciones de bit de signo y complemento a 2 [Usar]</li><li>• Explicar cómo las representaciones de tamaño fijo afectan en la exactitud y la precisión [Usar]</li><li>• Describir la representación interna de datos no numéricos como caracteres, cadenas, registros y arreglos [Usar]</li><li>• Convertir datos numéricos de un formato a otro [Usar]</li></ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	



**Unit 3: Organización de la Máquina a Nivel Ensamblador (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<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></ul>	<ul style="list-style-type: none"><li>• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse]</li><li>• Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse]</li><li>• Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse]</li><li>• Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse]</li><li>• Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar]</li><li>• Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar]</li><li>• Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar]</li><li>• Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse]</li><li>• Escribir segmentos de programa simples en lenguaje ensamblador [Usar]</li><li>• Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]</li></ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 4: Organización funcional (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>	<ul style="list-style-type: none"> <li>• Comparar implementaciones alternativas de ruta de datos [Evaluar]</li> <li>• Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse]</li> <li>• Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar]</li> <li>• Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar]</li> <li>• Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 5: Organización y Arquitectura del Sistema de Memoria (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 [Usar]</li> <li>• Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar]</li> <li>• Explique el efecto del retardo de la memoria en tiempo de ejecución [Usar]</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 [Evaluar]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

<b>Unit 6: Interfaz y comunicación (8 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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 [Usar]</li> <li>• Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar]</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>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

<b>Unit 7: Multiprocesamiento y arquitecturas alternativas (8 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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 [Evaluar]</li> <li>• Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse]</li> <li>• Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar]</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 electricos en paralelo backplane, interconexión memoria procesador y memoria remota via red, sus implicaciones para la latencia de acceso y el impacto en el rendimiento de un programa [Evaluar]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 8: Mejoras de rendimiento (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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> <li>• Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial.</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 [Usar]</li> <li>• Caracterizar los costos y beneficios de la precarga prefetching [Evaluar]</li> <li>• Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar]</li> <li>• Discutir las ventajas de rendimiento ofrecida en una arquitectura de multihebras junto con los factores que hacen difícil dar el máximo beneficio de estas [Evaluar]</li> <li>• Describir la importancia de la escalabilidad en el rendimiento [Evaluar]</li> </ul>
<b>Readings :</b> [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.
- [Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005.
- [PP05] Yale N Patt and Sanjay J Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.
- [HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. 4th. San Mateo, CA: Morgan Kaufman, 2006.
- [PCh06] Pong P.Chu. *RTL Hardware Design Using VHDL*. 1st. Wiley-Interscience, 2006.
- [JAs07] Peter J.Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Morgan Kaufmann, 2007.
- [Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.
- [HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture*. 2nd. Morgan Kaufmann, 2012.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS271. Data Management (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS271. Data Management
- 2.2 Semester : 4<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
  - CS112. Computer Science I. (2<sup>nd</sup> Sem)
- 2.8 Prerequisites :
  - CS1D2. Discrete Structures II. (2<sup>nd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques. It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

**5. GOALS**

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

## 7. TOPICS

Unit 1: Sistemas de Bases de Datos (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 de 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 de 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 largos volúmenes de datos [Usar]</li> </ul>
<b>Readings :</b> [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

<b>Unit 2: Modelado de datos (14 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>
<b>Readings :</b> [SW04], [EN04], [KS02]	

<b>Unit 3: Indexación (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• El impacto de índices en el rendimiento de consultas.</li> <li>• La estructura básica de un índice.</li> <li>• Mantener un buffer de datos en memoria.</li> <li>• Creando índices 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>
<b>Readings :</b> [WM01], [RG03], [ER15], [CJ11], [KS02]	

**Unit 4: Bases de Datos Relacionales (14 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [WM01], [RG03], [ER15], [CJ11], [KS02]	



Unit 5: Lenguajes de Consulta (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Visión general de lenguajes de base de datos.</li> <li>• SQL (definición de datos, formulacion 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>• Evaluar 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 relaciones 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>
Readings : [Die01], [EN04], [Cel05], [KS02]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

## 1. COURSE

MA106FCCS. Numerical Methods (Mandatory)

## 2. GENERAL INFORMATION

2.1 Course	: MA106FCCS. Numerical Methods
2.2 Semester	: 4 <sup>th</sup> Semester.
2.3 Credits	: 3
2.4 Horas	: 2 HT; 2 HP;
2.5 Duration of the period	: 16 weeks
2.6 Type of course	: Mandatory
2.7 Learning modality	: Face to face
2.8 Prerequisites	: MA103FCCS. Differential Calculus. (2 <sup>nd</sup> Sem)

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

Numerical methods are essential in computer science for approximating solutions to mathematical problems that cannot be solved analytically. This course provides an introduction to the most common numerical methods, including equation solving, interpolation, numerical integration, and the solution of differential equations.

## 5. GOALS

- Understand the importance of numerical methods in solving computational problems.
- Apply different numerical methods to approximate solutions to mathematical problems.
- Analyze the accuracy and efficiency of the numerical methods used.

## 6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

## 7. TOPICS

<b>Unit 1: Introduction to Numerical Methods (4 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Computer representation of numbers.</li> <li>• Round-off and truncation errors.</li> <li>• Error propagation.</li> <li>• Analysis of stability and convergence.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain how numbers are represented in a computer and the limitations of this representation. [Familiarizarse]</li> <li>• Differentiate between round-off and truncation errors. [Usar]</li> <li>• Analyze how errors propagate in numerical calculations. [Evaluar]</li> </ul>
<b>Readings :</b> [CC15], [BF10]	

<b>Unit 2: Solving Nonlinear Equations (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Bisection method.</li> <li>• Newton-Raphson method.</li> <li>• Secant method.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply the bisection method to find roots of equations. [Familiarizarse]</li> <li>• Use the Newton-Raphson method to approximate solutions. [Usar]</li> <li>• Implement the secant method to solve nonlinear equations. [Evaluar]</li> </ul>
<b>Readings :</b> [CC15], [BF10]	

<b>Unit 3: Interpolation and Polynomial Approximation (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Lagrange polynomial interpolation.</li> <li>• Newton interpolation.</li> <li>• Splines.</li> </ul>	<ul style="list-style-type: none"> <li>• Construct Lagrange interpolating polynomials. [Familiarizarse]</li> <li>• Apply Newton interpolation. [Usar]</li> <li>• Use splines to approximate functions. [Evaluar]</li> </ul>
<b>Readings :</b> [CC15], [BF10]	

<b>Unit 4: Numerical Integration (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Trapezoidal rule.</li> <li>• Simpson's rule.</li> <li>• Gaussian quadrature.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply the trapezoidal rule to approximate integrals. [Familiarizarse]</li> <li>• Use Simpson's rule to calculate integrals numerically. [Usar]</li> <li>• Apply Gaussian quadrature for numerical integration. [Evaluar]</li> </ul>
<b>Readings :</b> [CC15], [BF10]	

Unit 5: Numerical Solution of Ordinary Differential Equations (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>Euler's method.</li> <li>Runge-Kutta methods.</li> </ul>	<ul style="list-style-type: none"> <li>Apply Euler's method to approximate solutions of ODEs. [Familiarizarse]</li> <li>Implement Runge-Kutta methods to solve ODEs numerically. [Usar]</li> </ul>
Readings : [CC15], [BF10]	

Unit 6: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>Simulation of physical systems.</li> <li>Scientific modeling.</li> <li>Machine learning (e.g., model optimization).</li> </ul>	<ul style="list-style-type: none"> <li>Use numerical methods to simulate physical systems. [Familiarizarse]</li> <li>Apply numerical methods in scientific modeling. [Usar]</li> <li>Implement numerical methods in machine learning algorithms. [Evaluar]</li> </ul>
Readings : [CC15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ST251FCCS. Probability Calculation (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: ST251FCCS. Probability Calculation
<b>2.2 Semester</b>	: 4 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Probability calculus is fundamental in computer science for algorithm analysis, system modeling, and decision-making under uncertainty. This course introduces the basic concepts of probability, including random variables, probability distributions, and important theorems like Bayes' theorem.

**5. GOALS**

- Understand the fundamentals of probability calculus.
- Apply probability rules to solve problems.
- Analyze different probability distributions and their applications.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
  - 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)
- AG-C12)** Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

**7. TOPICS**

<b>Unit 1: Introduction to Probability (6 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Sample spaces and events.</li> <li>• Definition of probability.</li> <li>• Conditional probability and independence.</li> <li>• Bayes' theorem.</li> </ul>	<ul style="list-style-type: none"> <li>• Define sample spaces and events. [Familiarizarse]</li> <li>• Calculate probabilities of events. [Usar]</li> <li>• Apply Bayes' theorem to calculate conditional probabilities. [Evaluar]</li> </ul>
<b>Readings : [Ross2014], [Devore2016]</b>	

<b>Unit 2: Discrete Random Variables (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Discrete random variables.</li> <li>• Discrete probability distributions (Bernoulli, binomial, Poisson).</li> <li>• Expectation and variance.</li> </ul>	<ul style="list-style-type: none"> <li>• Define discrete random variables and their distributions. [Familiarizarse]</li> <li>• Calculate the expectation and variance of discrete random variables. [Usar]</li> <li>• Apply discrete distributions to model problems. [Evaluar]</li> </ul>
<b>Readings : [Ross2014], [Devore2016]</b>	

<b>Unit 3: Continuous Random Variables (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Continuous random variables.</li> <li>• Continuous probability distributions (uniform, exponential, normal).</li> <li>• Expectation and variance.</li> </ul>	<ul style="list-style-type: none"> <li>• Define continuous random variables and their distributions. [Familiarizarse]</li> <li>• Calculate the expectation and variance of continuous random variables. [Usar]</li> <li>• Apply continuous distributions to model problems. [Evaluar]</li> </ul>
<b>Readings : [Ross2014], [Devore2016]</b>	

<b>Unit 4: Joint Probability Distributions (8 hours)</b>	
<b>Competences Expected: 1,6,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Joint distributions of random variables.</li> <li>• Covariance and correlation.</li> <li>• Independence of random variables.</li> </ul>	<ul style="list-style-type: none"> <li>• Define joint distributions of random variables. [Familiarizarse]</li> <li>• Calculate the covariance and correlation between random variables. [Usar]</li> <li>• Determine the independence of random variables. [Evaluar]</li> </ul>
<b>Readings : [Ross2014], [Devore2016]</b>	

Unit 5: Central Limit Theorem and Law of Large Numbers (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Central Limit Theorem.</li> <li>• Law of Large Numbers.</li> </ul>	<ul style="list-style-type: none"> <li>• State and apply the Central Limit Theorem. [Familiarizarse]</li> <li>• Explain the Law of Large Numbers. [Usar]</li> </ul>
Readings : [Ross2014], [Devore2016]	

Unit 6: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Analysis of probabilistic algorithms.</li> <li>• Modeling stochastic systems.</li> <li>• Machine learning (e.g., probabilistic models).</li> </ul>	<ul style="list-style-type: none"> <li>• Use probability in algorithm analysis. [Familiarizarse]</li> <li>• Model systems using stochastic processes. [Usar]</li> <li>• Apply probability calculus in machine learning models. [Evaluar]</li> </ul>
Readings : [Ross2014]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ID104. Technical and professional English IV (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : ID104. Technical and professional English IV
- 2.2 Semester : 5<sup>th</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : ID103. Technical and professional English III. (4<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around all the world. There is no country where it is not spoken. In addition to being vital to your professional career

**5. GOALS**

- Increase the level of conversation in different subjects, in the students. As well as the ability to write and read documentation of all kinds.
- Bring the student to a more intense expression in the language domain.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)

**7. TOPICS**

Unit 1: Do and don't! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Mode Auxiliaries should, must and have got to.</li><li>• Affirmative, negative and interrogative sentences with modals.</li><li>• Terms for formal letters.</li><li>• Parts of short answers.</li><li>• Expressions for occupations.</li></ul>	<ul style="list-style-type: none"><li>• At the end of the eighth unit, each of the students, understanding the grammar of should and must auxiliaries, is able to express a greater number of actions in an obligatory and suggestive way. Also be able to express ideas describing occupations. Assumes the need to write formal letters</li></ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	



Unit 2: Going places! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Present and Future Present Time with Will</li> <li>• First conditional</li> <li>• Collocations</li> <li>• Vocabulary of prepositions of place and time</li> <li>• Expressions of connection of ideas</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the ninth unit, students having identified how to express present recognize the difference between future forms and apply them properly. They describe conditions accurately. They assume expressions to show place location. They use expressions of time and connectors to unite several ideas.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 3: Scared to death! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Infinitive and gerund verb patterns</li> <li>• What + Infinitive</li> <li>• Something + infinitive</li> <li>• Expressions of feelings</li> <li>• Exclamations of surprise</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the tenth unit of students, the chapters recognize and use the patterns of times in the past properly. They use exclamation marks. And describe feelings. They will use conjunctions to unite type ideas.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 4: Things that changed the world! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Passive Voice</li> <li>• Affirmative Prayers, Negatives and Questions</li> <li>• Use of participles, verbs and nouns that go together</li> <li>• Signals. Signs and notes</li> <li>• Summaries</li> <li>• Expressions to indicate prohibition</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the eleventh unit ,the students having identified the idea of passive actions describe actions appropriately in diverse situations that involve it. They recognize and apply participations. They assume the idea of respecting public signs and signals. They express ideas of habits. They make summaries.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 5: Dreams and reality! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Second Conditional</li> <li>• Auxiliar of mode "might"</li> <li>• Phrase Verbs</li> <li>• Social expressions vocabulary</li> <li>• Adverbs</li> <li>• Expressions to give advice</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the twelfth unit, students, starting from understanding the idea of Conditionals and expressing the possibility of elaborating sentences using the necessary elements. They will also assimilate the need for verbal phrases (2 word verbs). They will acquire vocabulary to describe social expressions.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 6: Making a living! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Present Perfect Continuous</li> <li>• Present Continuous</li> <li>• Occupations</li> <li>• Word formation</li> <li>• Adverbs</li> <li>• Expressions of use on the phone</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the thirteenth unit, they structure sentences with actions that include present and past in appropriate contexts. They emphasize the difference between types of occupations. Use appropriate expressions for telephone conversations.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

Unit 7: All you need is love! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Past Perfect and Past Simple</li> <li>• Report Expressions</li> <li>• Expressions of words in different contexts</li> <li>• Short and formal farewells</li> <li>• Love Stories</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fourteenth unit, students having learned the fundamentals of structuring past perfect time, differentiate it from the simple past. They emphasize the difference between words in different contexts. Describe farewell ideas. They use expressions to write love stories. They assume the idea of giving and doing interviews.</li> </ul>
Readings : [Soars022S], [Cambridge06], [MacGrew99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

**9. EVALUATION SYSTEM**

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

**10. BASIC BIBLIOGRAPHY**



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS212. Analysis and Design of Algorithms (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS212. Analysis and Design of Algorithms
- 2.2 Semester : 5<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
  - CS210. Algorithms and Data Structures. (4<sup>th</sup> Sem)
- 2.8 Prerequisites :
  - CS211. Theory of Computation. (4<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

An algorithm is, essentially, a well-defined set of rules or instructions that allow solving a computational problem. The theoretical study of the performance of the algorithms and the resources used by them, usually time and space, allows us to evaluate if an algorithm is suitable for solving a specific problem, comparing it with other algorithms for the same problem or even delimiting the boundary between Viable and impossible. This matter is so important that even Donald E. Knuth defined Computer Science as the study of algorithms. This course will present the most common techniques used in the analysis and design of efficient algorithms, with the purpose of learning the fundamental principles of the design, implementation and analysis of algorithms for the solution of computational problems

**5. GOALS**

- Develop the ability to evaluate the complexity and quality of algorithms proposed for a given problem.
- Study the most representative, introductory algorithms of the most important classes of problems treated in computation.
- Develop the ability to solve algorithmic problems using the fundamental principles of algorithm design learned.
- Be able to answer the following questions when a new algorithm is presented: How good is the performance ?, Is there a better way to solve the problem?

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

<b>Unit 1: Análisis Básico (10 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>• Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial.</li> <li>• Asymptotic Notation</li> <li>• Análisis de algoritmos iterativos y recursivos.</li> <li>• Inductive proofs and correctness of algorithms</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>• Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Evaluar]</li> <li>• Lista y contraste de clases estándares de complejidad [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>• Analyze worst-case running times of algorithms using asymptotic analysis [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> <li>• Argue the correctness of algorithms using inductive proofs [Evaluar]</li> </ul>
<b>Readings :</b> [KT05], [DPV06], [Cor+09], [SF13], [Knu97]	

<b>Unit 2: Estrategias Algorítmicas (30 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<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>
<b>Readings :</b> [KT05], [DPV06], [Cor+09], [Als99]	

Unit 3: Algoritmos y Estructuras de Datos fundamentales (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <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>• Cache oblivious algorithms</li> <li>• Number theory and cryptography</li> </ul>	<ul style="list-style-type: none"> <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 en específico [Evaluar]</li> <li>• Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar]</li> </ul>
<b>Readings :</b> [KT05], [DPV06], [Cor+09], [SW11], [GT09]	

Unit 4: Computabilidad y complejidad básica de autómatas (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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-Completos.</li> <li>• Reductions</li> </ul>	<ul style="list-style-type: none"> <li>• Define las clases P y NP [Familiarizarse]</li> <li>• Explique el significado de NP-Compleitud [Familiarizarse]</li> </ul>
<b>Readings :</b> [KT05], [DPV06], [Cor+09]	

Unit 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Grafos (ej. Ordenamiento Topológico, encontrando componentes fuertemente conectados)</li> <li>• Algoritmos aleatorios.</li> <li>• Análisis amortizado.</li> <li>• Análisis Probabilístico.</li> <li>• Approximation Algorithms</li> <li>• Linear Programming</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 lineales, etc) [Familiarizarse]</li> <li>• Seleccionar y aplicar técnicas avanzadas de análisis (ejemplo, amortizado, probabilístico, etc) para algoritmos [Usar]</li> </ul>
<b>Readings :</b> [KT05], [DPV06], [Cor+09], [Tar83], [Raw92]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

## 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

## 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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*. v. 1. Addison-Wesley, 1997.
- [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.
- [Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009.
- [GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS272. Databases II (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |   |
|-----------------------------------|---|
| <b>2.1 Course</b>                 | : CS272. Databases II                           |
| <b>2.2 Semester</b>               | : 5 <sup>th</sup> Semester.                     |
| <b>2.3 Credits</b>                | : 3   |
| <b>2.4 Horas</b>                  | : 1 HT; 4 HP;                                   |
| <b>2.5 Duration of the period</b> | : 16 weeks                                      |
| <b>2.6 Type of course</b>         | : Mandatory                                     |
| <b>2.7 Learning modality</b>      | : Face to face                                  |
| <b>2.8 Prerequisites</b>          | : CS271. Data Management. (4 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Information Management (IM) plays a leading role in almost every area where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of access and update of stored information, data modeling and abstraction, and physical file storage techniques.

It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which IM methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable constraints, including scalability and Usability.

**5. GOALS**

- To make the student understand the different applications that the databases have, in the different areas of knowledge.
- Show appropriate ways of storing information based on their various approaches and their subsequent retrieval of information.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

**7. TOPICS**



Unit 1: Diseño Físico de Bases de Datos (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Bur04], [Cel05]	

Unit 2: Procesamiento de Transacciones (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Phi97], [Ram04]	

**Unit 3: Almacenamiento y Recuperación de Información (10 hours)****Competences Expected:**

Topics	Learning Outcomes
<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, fuzificación (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, citasiones.</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, integrdad</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 que temas son específicos para una recuperación de la información eficiente [Usar]</li><li>• Da aplicaciones de estrategias alternativas de búsqueda y explica porqué 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>
<b>Readings :</b> [Pet98], [Ram04]	

Unit 4: Bases de Datos Distribuidas (36 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 distinguidos y el método de votación. [Usar]</li> <li>• Describir los tres niveles del software en el modelo cliente servidor [Usar]</li> </ul>
Readings : [M T99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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 Ozsü. *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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS291. Software Engineering I (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS291. Software Engineering I
- 2.2 Semester : 5<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
  - CS113. Computer Science II. (3<sup>rd</sup> Sem)
- 2.8 Prerequisites :
  - CS271. Data Management. (4<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The aim of developing software, except for extremely simple applications, requires the execution of a well-defined development process. Professionals in this area require a high degree of knowledge of the different models and development process, so that they are able to choose the most suitable for each development project. On the other hand, the development of medium and large-scale systems requires the use of pattern and component libraries and the mastery of techniques related to component-based design

**5. GOALS**

- Provide the student with a theoretical and practical framework for the development of software under quality standards.
- Familiarize the student with the software modeling and construction processes through the use of CASE tools.
- Students should be able to select architectures and ad-hoc technology platforms for deployment scenarios
- Applying component-based modeling to ensure variables such as quality, cost, and time-to-market in development processes.
- Provide students with best practices for software verification and validation.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

**Unit 1: Ingeniería de Requisitos (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 elicitación.</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>	<ul style="list-style-type: none"><li>• Enumerar los componentes clave de un caso de uso o una descripción similar de algún comportamiento que es requerido para un sistema [Evaluar]</li><li>• Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento [Evaluar]</li><li>• Interpretar un modelo de requisitos dada por un sistema de software simple [Evaluar]</li><li>• Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos [Evaluar]</li><li>• Enumerar los componentes clave de un modelo de datos (por ejemplo, diagramas de clases o diagramas ER) [Evaluar]</li><li>• Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software [Evaluar]</li><li>• Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos [Evaluar]</li><li>• Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio [Evaluar]</li><li>• Comparar los métodos ágiles y el dirigido por planes para la especificación y validación de requisitos y describir los beneficios y riesgos asociados con cada uno [Evaluar]</li><li>• Usar un método común, no formal para modelar y especificar los requisitos para un sistema de software de tamaño medio [Evaluar]</li><li>• Traducir al lenguaje natural una especificación de requisitos de software (por ejemplo, un contrato de componentes de software) escrito en un lenguaje de especificación formal [Evaluar]</li><li>• Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos [Evaluar]</li><li>• Diferenciar entre el rastreo (<i>tracing</i>) hacia adelante y hacia atrás y explicar su papel en el proceso de validación de requisitos [Evaluar]</li></ul>
<b>Readings :</b> [ES14], [HF03]	

**Unit 2: Diseño de Software (18 hours)****Competences Expected:****Topics****Learning Outcomes**

- 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.
- 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.
- Modelos estructurales y de comportamiento de los diseños de software.
- Diseño de patrones.
- Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.
- Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).
- 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 standard widget set)
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Medición y análisis de la calidad de un diseño.
- Compensaciones entre diferentes aspectos de la calidad.
- Aplicaciones en frameworks.
- 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.
- Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design).
  - Principio de privilegios mínimos
  - Principio de falla segura por defecto
  - Principio de aceptabilidad psicológica

- Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Familiarizarse]
- 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]
- Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]
- 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 [Familiarizarse]
- Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]
- Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar]
- Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Evaluar]
- 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 [Familiarizarse]
- Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (*3-tier*), *pipe-and-filter*, y cliente-servidor [Familiarizarse]
- Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Evaluar]
- Aplicar ejemplos simples de patrones en un diseño de software [Usar]
- Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Familiarizarse]
- Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]
- Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Familiarizarse]
- Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema, [Usar]
- Discutir y seleccionar la arquitectura de software

Unit 3: Construcción de Software (24 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Prácticas de codificación: técnicas, idiomas/patrones, mecanismos para construcción de programas de calidad: <ul style="list-style-type: none"> <li>– Prácticas de codificación defensiva</li> <li>– Prácticas de codificación segura</li> <li>– Utilizando mecanismos de manejo de excepciones para hacer el programa más robusto, tolerante a fallas</li> </ul> </li> <li>• Normas de codificación.</li> <li>• Estrategias de integración.</li> <li>• Desarrollando contexto: "campo verde" frente a la base de código existente : <ul style="list-style-type: none"> <li>– Análisis de cambio impacto</li> <li>– Cambio de actualización</li> </ul> </li> <li>• Los problemas de seguridad potenciales en los programas : <ul style="list-style-type: none"> <li>– Buffer y otros tipos de desbordamientos</li> <li>– Condiciones elemento Race</li> <li>– Inicialización incorrecta, incluyendo la elección de los privilegios</li> <li>– Entrada Comprobación</li> <li>– Suponiendo éxito y corrección</li> <li>– La validación de las hipótesis</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Describir técnicas, lenguajes de codificación y mecanismos de implementación para conseguir las propiedades deseadas, tales como la confiabilidad, la eficiencia y la robustez [Evaluar]</li> <li>• Construir código robusto utilizando los mecanismos de manejo de excepciones [Evaluar]</li> <li>• Describir la codificación segura y prácticas de codificación de defensa [Evaluar]</li> <li>• Seleccionar y utilizar un estándar de codificación definido en un pequeño proyecto de software [Evaluar]</li> <li>• Comparar y contrastar las estrategias de integración incluyendo: de arriba hacia abajo (<i>top-down</i>), de abajo hacia arriba (<i>bottom-up</i>), y la integración Sándwich [Evaluar]</li> <li>• Describir el proceso de analizar e implementar los cambios a la base de código desarrollado para un proyecto específico [Evaluar]</li> <li>• Describir el proceso de analizar e implementar los cambios a la base de código desarrollado para un proyecto específico [Evaluar]</li> <li>• Reescribir un programa sencillo para eliminar vulnerabilidades comunes, tales como desbordamientos de búffer, desbordamientos de enteros y condiciones de carrera [Evaluar]</li> <li>• Escribir un componente de software que realiza alguna tarea no trivial y es resistente a errores en la entrada y en tiempo de ejecución [Evaluar]</li> </ul>
Readings : [ES14], [HF03]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[HF03] Brian Lyons Hans-Erik Eriksson Magnus Penker and Davis Fado. *UML 2 Toolkit*. 2nd. Wiley, Oct. 2003.

[ES14] Bert Bates Eric Freeman Elisabeth Robson and Kathy Sierra. *Head First Design Patterns*. 2nd. O'Reilly Media, Inc, July 2014.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS2S1. Operating systems (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS2S1. Operating systems
- 2.2 Semester : 4<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS221. Computer Systems Architecture. (3<sup>rd</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

An Operating System (OS) manages the computing resources to complete the execution of multiple applications and their associated processes. This course teaches the design of modern operating systems; and introduces their fundamental concepts covering multiple-program execution, scheduling, memory management, file systems, and security. Also, the course includes programming activities on a minimal operating system to solve problems and extend its functionality. Notice that these activities require much time to complete. However, working on them provides valuable insight into operating systems.

**5. GOALS**

- Study the design of modern operating systems.
- Provide a practical experience by designing and implementing a minimal operating system.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Familiarity)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

**7. TOPICS**

Unit 1: Visión general de Sistemas Operativos (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 contemporáneo 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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 2: Principios de Sistemas Operativos (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

**Unit 3: Concurrency (9 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 4: Planificación y despacho (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 5: Manejo de memoria (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>	<ul style="list-style-type: none"> <li>• Explicar la jerarquía de la memoria y costo-rendimiento de intercambio [Familiarizarse]</li> <li>• Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse]</li> <li>• Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar]</li> <li>• Defiende las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse]</li> <li>• Describir el motivo y el uso de memoria caché (rendimiento y proximidad, dimensión diferente de como los caches complican el aislamiento y abstracción en VM) [Familiarizarse]</li> <li>• Estudiar los conceptos de <i>thrashing</i>, tanto en términos de las razones por las que se produce y las técnicas usadas para el reconocimiento y manejo del problema [Familiarizarse]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 6: Seguridad y protección (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Visión general de la seguridad del sistema .</li> <li>• Política / mecanismo de separación.</li> <li>• Métodos de seguridad y dispositivos.</li> <li>• Protección, control de acceso y autenticación.</li> <li>• Las copias de seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>• Articular la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse]</li> <li>• Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse]</li> <li>• Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse]</li> <li>• 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]</li> </ul>
<b>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 7: Máquinas virtuales (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 8: Manejo de dispositivos (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 9: Sistema de archivos (6 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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 bitacora 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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 10: Sistemas empotrados y de tiempo real (6 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

<b>Unit 11: Tolerancia a fallas (3 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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, reinicie 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>Readings :</b> [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 12: Evaluación del desempeño de sistemas (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Tan01] Andrew S. Tanenbaum. *Modern Operating Systems*, 4/E. Prentice Hall, 2001.
- [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.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS342. Compilers (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS342. Compilers
- 2.2 Semester : 5<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS211. Theory of Computation. (4<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

That the student knows and understands the concepts and fundamental principles of the theory of compilation to realize the construction of a compiler

**5. GOALS**

- Know the basic techniques used during the process of intermediate generation, optimization and code generation.
- Learning to implement small compilers.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

**Unit 1: Representación de programas (5 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Programas que tienen otros programas como entrada tales como interpretes, 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 interprete, 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>
<b>Readings :</b> [Lou04b]	

Unit 2: Traducción y ejecución de lenguajes (10 hours)	
Competences Expected:	
Topics	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 llamadas, 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 típicamente 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>
<b>Readings :</b> [Aho+11], [Lou04a], [App02], [TS98]	

Unit 3: Análisis de sintaxis (10 hours)	
Competences Expected:	
Topics	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 - bracktracking, 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>
<b>Readings :</b> [Aho+11], [Lou04a], [App02], [TS98]	

Unit 4: Análisis semántico de compiladores (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 atribuidas.</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>
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 5: Generación de código (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

ID105. Technical and professional English V (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: ID105. Technical and professional English V
<b>2.2 Semester</b>	: 6 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 4
<b>2.4 Horas</b>	: 2 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: ID104. Technical and professional English IV. (5 <sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, undoubtedly English is the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services, English is perhaps the most important practical tool that the student must master from the outset as part of his / her integral education

**5. GOALS**

- Increase the ability and fluency of speaking and understanding the English language.
- That the students interact with greater emphasis in the creation of dialogues.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)

**7. TOPICS**

Unit 1: It's a wonderful world (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Auxiliary verbs</li> <li>• Verb tenses</li> <li>• Negative Questions and Prayers</li> <li>• Short answers</li> <li>• Word formation</li> <li>• Colloquial expressions</li> <li>• Error correction</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the first unit, each student, understanding the grammar of auxiliaries and different types of sentences, is able to express a greater number of expressions of time and also use prepositions to describe varied places and times. He is also able to analyze and express ideas about word formation.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

Unit 2: Happiness! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Simple present</li> <li>• Present continuous</li> <li>• Passive Voice in Present</li> <li>• Verbs for sports and free time</li> <li>• Types of numbers and</li> <li>• Inventions / Modern World</li> <li>• Corrección de errores</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the second unit, students have identified how to express sports and leisure activities. It uses all kinds of numerical expressions. Express situations and states related to present forms. Explain and apply vocabulary of outdoor activities.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

Unit 3: Telling tales! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Simple Past Time</li> <li>• Past Continuous</li> <li>• Passive Voice in Past</li> <li>• Vocabulary of Art and Literature</li> <li>• Expressions to give and ask opinions</li> <li>• Stories and stories</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the third unit, students having recognized the characteristics of the past passive forms, they use these make descriptions of various types. Describe art and literature and give indications of opinion. They will use conjunctions to unite type ideas.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

Unit 4: Doing the right thing! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Mode Auxiliary Verbs I.</li> <li>• Affirmative, Negative and Modals Questions</li> <li>• Use of nationalities and other adjectives</li> <li>• Expressions of orders and offers</li> <li>• Guide to Good Manners</li> <li>• Form Fill</li> <li>• Phonetic symbols</li> </ul>	<ul style="list-style-type: none"> <li>• At the conclusion of the fourth unit, the students having identified the idea of expressing ideas of modes of actions that happen at the moment or that are related at any time, structure sentences in the Present. They express ideas of nationalities and make requests and offers varied.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

Unit 5: On the move! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Future with Will</li> <li>• Future Time Prayers with going to</li> <li>• Use of might for future</li> <li>• Climate Expressions</li> <li>• Vocabulary of the climate</li> <li>• Expressions for hotels and transportation</li> <li>• E-mails</li> </ul>	<ul style="list-style-type: none"> <li>• At the end of the fifth unit, students, from the understanding of future time, will elaborate sentences using the necessary elements. They will also assimilate the need to express ideas of the climate. They will acquire vocabulary to describe use of public transportation. Expressions will be presented to order at hotels.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

Unit 6: I just love it! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Questions with Shapes 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>• At the end of the sixth unit, students having learned the basics of structuring questions with like and with verbal patterns work applied to appropriate contexts. They emphasize the difference between meals, places and people. Describe sights and sounds. They use expressions to compare daily life in different places. They assume the idea of different lifestyles.</li> </ul>
Readings : [Soars023S], [Soars023W], [Soars023T], [Cambridge06], [MacGrew99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.



### **8.3 Practical Sessions**

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

### **9. EVALUATION SYSTEM**

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

### **10. BASIC BIBLIOGRAPHY**



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS231. Networking and Communication (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS231. Networking and Communication
<b>2.2 Semester</b>	: 6 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 1 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: CS2S1. Operating systems . (4 <sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The ever-growing development of communication and information technologies means that there is a marked tendency to establish more computer networks that allow better information management..

In this second course, participants will be introduced to the problems of communication between computers, through the study and implementation of communication protocols such as TCP / IP and the implementation of software on these protocols

**5. GOALS**

- That the student implements and / or modifies a data communication protocols.
- That the student master the data transmission techniques used by the existing network protocols.
- That the student knows the latest trends in networks that are being applied on the Internet.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Familiarity)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

**7. TOPICS**

Unit 1: Introducción a redes (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [KR13]	

Unit 2: Aplicaciones en red (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [KR13]	

Unit 3: Entrega confiable de datos (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [KR13]	

<b>Unit 4: Ruteo y reenvío (12 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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 (direccionamiento 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 direccionamiento jerárquico [Familiarizarse]</li> </ul>
<b>Readings :</b> [KR13]	

<b>Unit 5: Redes de área local (10 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>Readings :</b> [KR13]	

<b>Unit 6: Asignación de recursos (12 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>Readings :</b> [KR13]	

<b>Unit 7: Celulares (5 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>Readings :</b> [KR13], [Cha16]	

Unit 8: Redes sociales (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [KR13], [Kad11]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS261. Artificial Intelligence (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS261. Artificial Intelligence
- 2.2 Semester : 6<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites :
  - CS210. Algorithms and Data Structures. (4<sup>th</sup> Sem)
  - ST251FCCS. Probability Calculation. (4<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

**5. GOALS**

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.
- Build a notion of intelligence that later supports the tasks of your simulation.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Familiarity)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

**7. TOPICS**

**Unit 1: Cuestiones fundamentales (2 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [De 06], [Pon+14]	

<b>Unit 2: Agentes (2 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>
<b>Readings :</b> [Nil01], [RN03], [Pon+14]	



Unit 3: Estrategias de búsquedas básicas (2 hours)	
Competences Expected:	
Topics	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>
<b>Readings :</b> [Nil01], [Pon+14]	

Unit 4: Búsqueda Avanzada (18 hours)	
Competences Expected:	
Topics	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>
<b>Readings :</b> [Gol89], [Nil01], [RN03], [Pon+14]	

Unit 5: Razonamiento Bajo Incertidumbre (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
<b>Readings :</b> [KF09], [RN03]	

Unit 6: Aprendizaje Automático Básico (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 Naive Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]</li> </ul>
<b>Readings :</b> [Mit98], [RN03], [Pon+14]	

**Unit 7: Aprendizaje de máquina avanzado (20 hours)****Competences Expected:**

Topics	Learning Outcomes
<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, area 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>
<b>Readings :</b> [RN03], [KF09], [Mur12]	

**Unit 8: Procesamiento del Lenguaje Natural (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Gramaticas 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>• Metodos 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>– Precision 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>
<b>Readings :</b> [Nil01], [RN03], [Pon+14]	

Unit 9: Visión y percepción por computador (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>– Análisis 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>
<b>Readings :</b> [Nil01], [RN03], [Pon+14]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS2H1. User Experience (UX) (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |   |
|-----------------------------------|---|
| <b>2.1 Course</b>                 | : CS2H1. User Experience (UX)                       |
| <b>2.2 Semester</b>               | : 7 <sup>th</sup> Semester.                         |
| <b>2.3 Credits</b>                | : 3   |
| <b>2.4 Horas</b>                  | : 1 HT; 4 HP;                                       |
| <b>2.5 Duration of the period</b> | : 16 weeks  |
| <b>2.6 Type of course</b>         | : Mandatory   |
| <b>2.7 Learning modality</b>      | : Face to face                                      |
| <b>2.8 Prerequisites</b>          | : CS393. Information systems. (6 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Language has been one of the most significant creations of humanity. From body language and gesture, through verbal and written communication, to iconic symbolic codes and others, it has made possible complex interactions Among humans and facilitated considerably the communication of information. With the invention of automatic and semi-automatic devices, including computers, The need for languages or interfaces to be able to interact with them, has gained great importance. The utility of the software, coupled with user satisfaction and increased productivity, depends on the effectiveness of the User-Computer Interface. So much so, that often the interface is the most important factor in the success and failure of any computer system. The design and implementation of appropriate Human-Computer Interfaces, which in addition to complying with the technical requirements and the transactional logic of the application, consider the subtle psychological implications, sciences and user facilities, It consumes a good part of the life cycle of a software project, and requires specialized skills, both for the construction of the same, and for the performance of usability tests.

**5. GOALS**

- Know and apply criteria of usability and accessibility to the design and construction of human-computer interfaces, always looking for technology to adapt to people and not people to technology.
- That the student has a vision focused on the user experience by applying appropriate conceptual and technological approaches.
- Understand how emerging technology makes possible new styles of interaction.
- Determine the basic requirements at the interface level, hardware and software for the construction of immersive environments.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Familiarity)

5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)

7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Familiarity)

## 7. TOPICS

Unit 1: Fundamentos (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 centrado 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>Readings :</b> [Dix+04], [Sto+05], [RS11]	

Unit 2: Factores Humanos (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>Readings :</b> [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	



**Unit 3: Diseño y Testing centrados en el usuario (16 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

**Unit 4: Diseño de Interacción (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>Readings :</b> [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

**Unit 5: Nuevas Tecnologías Interactivas (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>– Realidad 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>
<b>Readings :</b> [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unit 6: Colaboración y Comunicación (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Discuta 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>
Readings : [Dix+04], [Sto+05], [RS11]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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. Shneiderman. *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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS311. Competitive Programming (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS311. Competitive Programming
- 2.2 Semester : 6<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS212. Analysis and Design of Algorithms. (5<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Competitive Programming combines problem-solving challenges with the fun of competing with others. It teaches participants to think faster and develop problem-solving skills that are in high demand in the industry. This course will teach you to solve algorithmic problems quickly by combining theory of algorithms and data structures with practice solving problems.

**5. GOALS**

- That the student uses techniques of data structures and complex algorithms..
- That the student apply the concepts learned for the application on a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Introduction (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Introduction to Competitive Programming</li> <li>• Computational model</li> <li>• Runtime and space complexity</li> <li>• Recurrence and recursion</li> <li>• Divide and conquer</li> </ul>	<ul style="list-style-type: none"> <li>• Identify and learn how to use the resources in the Random Access Machine (RAM) computational model. [Usar]</li> <li>• Compute the runtime and space complexity for written algorithms. [Usar]</li> <li>• Compute the recurrence relations for recursive algorithms. [Usar]</li> <li>• Solve problems related to searching and sorting. [Usar]</li> <li>• Learning to select the right algorithms for divide-and-conquer problems. [Usar]</li> <li>• Design new algorithms for real-world problem solving.[Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 2: Data structure (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Arrays and strings problems</li> <li>• Linked lists problems</li> <li>• Stacks and queues problems</li> <li>• Trees problems</li> <li>• Hash tables problems</li> <li>• Heaps problems</li> </ul>	<ul style="list-style-type: none"> <li>• Recognize different data structures, their complexities, uses and restrictions.[Usar]</li> <li>• Identify the type of data structure appropriate to the resolution of the problem. [Usar]</li> <li>• Recognize types of problems associated with operations on data structures such as searching, inserting, deleting and updating.[Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 3: Algorithmic Design Paradigms (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Brute force</li> <li>• Divide and conquer</li> <li>• Backtracking</li> <li>• Greedy</li> <li>• Dynamic Programming</li> </ul>	<ul style="list-style-type: none"> <li>• Learning the different algorithmic design paradigms.[Usar]</li> <li>• Learning to select the right algorithms for different problems applying different algorithmic design paradigms.[Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 4: Graphs (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Graphs transversal</li> <li>• Graphs applications</li> <li>• Shortest path</li> <li>• Networks and flows</li> </ul>	<ul style="list-style-type: none"> <li>• Identify problems classified as graph problems. [Usar]</li> <li>• Learn how to select the right algorithms for network problems (transversal, MST, shortest-path, network and flows). [Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 5: Advanced topics (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Number theory</li> <li>• Probabilities and combinations</li> <li>• String algorithms (tries, string hashing, z-algorithm)</li> <li>• Geometric algorithms</li> </ul>	<ul style="list-style-type: none"> <li>• Learning to select the right algorithms for problems in number theory and mathematics as they are important in competitive programming. [Usar]</li> <li>• Learning to select the right algorithms for problems about probabilities and combinations, strings and computational geometry. [Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 6: Domain specific problems (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Latency and throughput</li> <li>• Parallelism</li> <li>• Networks</li> <li>• Storage</li> <li>• High availability</li> <li>• Caching</li> <li>• Proxies</li> <li>• Load balancers</li> <li>• Key-value stores</li> <li>• Replicating and sharing</li> <li>• Leader election</li> <li>• Rate limiting</li> <li>• Logging and monitoring</li> </ul>	<ul style="list-style-type: none"> <li>• Learning to design systems for different domain-specific problems by applying knowledge about networks, distributed computing, high availability, storage and system architecture.[Usar]</li> </ul>
<b>Readings :</b> [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

## 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

## 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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*. ElementsOfProgrammingInterviews.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*. Springer, 2017.
- [Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

## 1. COURSE

CS312. Advanced Data Structures (Mandatory)

## 2. GENERAL INFORMATION

- |                            |   |   |
|----------------------------|---|---|
| 2.1 Course                 | : | CS312. Advanced Data Structures                                 |
| 2.2 Semester               | : | 6 <sup>th</sup> Semester.                                       |
| 2.3 Credits                | : | 4   |
| 2.4 Horas                  | : | 2 HT; 4 HP;   |
| 2.5 Duration of the period | : | 16 weeks  |
| 2.6 Type of course         | : | Mandatory   |
| 2.7 Learning modality      | : | Face to face  |
| 2.8 Prerequisites          | : | CS212. Analysis and Design of Algorithms. (5 <sup>th</sup> Sem) |

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

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. GOALS

- 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. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Familiarity)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

## 7. TOPICS

<b>Unit 1: Multidimensional Data (16 hours)</b>	
<b>Competences Expected: 1,2</b>	
<b>Topics</b>	<b>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 trascendencia 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>Readings :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

<b>Unit 2: Multidimensional Acces Data Structures (16 hours)</b>	
<b>Competences Expected: 1,2</b>	
<b>Topics</b>	<b>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 deestructuras 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>Readings :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

<b>Unit 3: Approximate Access Methods (20 hours)</b>	
<b>Competences Expected: 1,2</b>	
<b>Topics</b>	<b>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>Readings :</b> [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 4: Métodos de Acceso Aproximados (20 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
<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 una factor muy importante.</li> </ul>
Readings : [Sam06], [PI06], [Zez+07]	

Unit 5: Clustering (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
<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>
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 6: Temporal Data Structures (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
<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>
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 7: Final Talks (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
<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>
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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/ecuadros/cursos/pdfs/>. 2004.
- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

MA307. Mathematics applied to computing (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : MA307. Mathematics applied to computing
- 2.2 Semester : 6<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS212. Analysis and Design of Algorithms. (5<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Este curso es importante porque desarrolla tópicos del Álgebra Lineal y de Ecuaciones Diferenciales Ordinarias útiles en todas aquellas áreas de la ciencia de la computación donde se trabaja con sistemas lineales y sistemas dinámicos.

**5. GOALS**

- Que el alumno tenga la base matemática para el modelamiento de sistemas lineales y sistemas dinámicos necesarios en el Área de Computación Gráfica e Inteligencia Artificial.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Espacios vectoriales.</li><li>• Independencia, base y dimensión.</li><li>• Dimensiones y ortogonalidad de los cuatro subespacios.</li><li>• Aproximaciones por mínimos cuadrados.</li><li>• Proyecciones</li><li>• Bases ortogonales y Gram-Schmidt</li></ul>	<ul style="list-style-type: none"><li>• Identificar espacios generados por vectores linealmente independientes[Usar]</li><li>• Construir conjuntos de vectores ortogonales[Usar]</li><li>• Aproximar funciones por polinomios trigonométricos[Usar]</li></ul>
Readings : [Str03], [Apó73]	

<b>Unit 2: (0 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Concepto de transformación lineal.</li> <li>• Matriz de una transformación lineal.</li> <li>• Cambio de base.</li> <li>• Diagonalización y pseudoinversa</li> </ul>	<ul style="list-style-type: none"> <li>• Determinar el núcleo y la imagen de una transformación[Usar]</li> <li>• Construir la matriz de una transformación[Usar]</li> <li>• Determinar la matriz de cambio de base[Usar]</li> </ul>
<b>Readings :</b> [Str03], [Apó73]	

<b>Unit 3: (0 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Diagonalización de una matriz</li> <li>• Matrices simétricas</li> <li>• Matrices definidas positivas</li> <li>• Matrices similares</li> <li>• La descomposición de valor singular</li> </ul>	<ul style="list-style-type: none"> <li>• Encontrar la representación diagonal de una matriz[Usar]</li> <li>• Determinar la similaridad entre matrices[Usar]</li> <li>• Reducir una forma cuadrática real a diagonal[Usar]</li> </ul>
<b>Readings :</b> [Str03], [Apó73]	

<b>Unit 4: (0 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Exponencial de una matriz</li> <li>• Teoremas de existencia y unicidad para sistemas lineales homogéneos con coeficientes constantes</li> <li>• Sistemas lineales no homogéneos con coeficientes constantes.</li> </ul>	<ul style="list-style-type: none"> <li>• Hallar la solución general de un sistema lineal no homogéneo[Usar]</li> <li>• Resolver problemas donde intervengan sistemas de ecuaciones diferenciales[Usar]</li> </ul>
<b>Readings :</b> [Zil02], [Apó73]	

<b>Unit 5: (0 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Sistemas dinámicos</li> <li>• El teorema fundamental</li> <li>• Existencia y unicidad</li> <li>• El flujo de una ecuación diferencial</li> </ul>	<ul style="list-style-type: none"> <li>• Discutir la existencia y la unicidad de una ecuación diferencial[Usar]</li> <li>• Analizar la continuidad de las soluciones[Usar]</li> <li>• Estudiar la prolongación de una solución[Usar]</li> </ul>
<b>Readings :</b> [HS74]	

<b>Unit 6: (0 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Estabilidad</li> <li>• Funciones de Liapunov</li> <li>• Sistemas gradientes</li> </ul>	<ul style="list-style-type: none"> <li>• Analizar la estabilidad de una solución[Usar]</li> <li>• Hallar la función de Liapunov para puntos de equilibrio[Usar]</li> <li>• Trazar el retrato de fase un flujo gradiente[Usar]</li> </ul>
<b>Readings :</b> [Zil02], [HS74]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[Apó73] Tom M Apóstol. *Calculus Vol II*. Editorial Reverté, 1973.

[HS74] Morris W. Hirsh and Stephen Smale. *Differential Equatons, Dynamical Systems, and Linear Álgebra*. Academia Press, 1974.

[Zil02] Dennis G. Zill. *Ecuaciones Diferenciales con Problemas de Valores en la Frontera*. Thomson Learning, 2002.

[Str03] Gilbert Strang. *Introduction to Linear Algebra, 3rd edición*. Wellesley-Cambridge Press, 2003.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS251. Computer graphics (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS251. Computer graphics
<b>2.2 Semester</b>	: 7 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 4
<b>2.4 Horas</b>	: 2 HT; 4 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: MA106FCCS. Numerical Methods. (4 <sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

It offers an introduction to the area of Computer Graphics, which is an important part of Computer Science. The purpose of this course is to investigate the fundamental principles, techniques and tools for this area.

**5. GOALS**

- Bring students to concepts and techniques used in complex 3-D graphics applications.
- Give the student the necessary tools to determine which graphics software and which platform are best suited to develop a specific application.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**



**Unit 1: Conceptos Fundamentales (6 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 píxeles [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 básico de la producción de movimiento continuo a partir de una secuencia de cuadros discretos (algunas veces llamado it flicker fusion ) [Familiarizarse]</li></ul>
<b>Readings :</b> [HB90]	

**Unit 2: Rendering Básico (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 afin [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>
<b>Readings :</b> [HB90], [Hug+13], [Wol11], [Shr+13]	

**Unit 3: Programación de Sistemas Interactivos (2 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [HB90]	

**Unit 4: Modelado Geométrico (15 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [HB90], [Shr+13]	

Unit 5: Renderizado Avanzado (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Tiempo (desenfoco de movimiento), la posición del objetivo (enfoco), 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>
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 6: Animación por computadora (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Cinématica 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>
Readings : [HB90], [Shr+13]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

### 9. EVALUATION SYSTEM

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

### 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS292. Software Engineering II (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : CS292. Software Engineering II                       |
| <b>2.2 Semester</b>               | : 6 <sup>th</sup> Semester.                            |
| <b>2.3 Credits</b>                | : 4  |
| <b>2.4 Horas</b>                  | : 2 HT; 4 HP;  |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face   |
| <b>2.8 Prerequisites</b>          | : CS291. Software Engineering I. (5 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The topics of this course extend the ideas of software design and development from the introduction sequence to programming to encompass the problems encountered in large-scale projects. It is a broader and more complete view of Software Engineering appreciated from a Project point of view.

**5. GOALS**

- Enable students to be part of and define software development teams facing real-world problems.
- familiarize the students with the process of administering a software project in such a way as to be able to create, improve and use tools and metrics that allow them to carry out the estimation and monitoring of a software project
- Create, evaluate and execute a test plan for medium-sized code segments, Distinguish between different types of tests, lay the foundation for creating, improve test procedures and tools for these purposes
- Select with justification an appropriate set of tools to support the development of a range of software products.
- Create, improve and use existing patterns for software maintenance. Disclose features and design patterns for software reuse.
- Identify and discuss different specialized systems, create, improve and use specialized standards for the design, implementation, maintenance and testing of specialized systems.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

**Unit 1: Herramientas y Entornos (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Administración de configuración de software y control de versiones.</li><li>• Administración de despliegues.</li><li>• Análisis de requerimientos y herramientas para modelado del diseño.</li><li>• Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico.</li><li>• Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas)<ul style="list-style-type: none"><li>– Integración continua.</li></ul></li><li>• Mecanismos y conceptos de herramientas de integración.</li></ul>	<ul style="list-style-type: none"><li>• Administración de configuración de software y control de versiones. [Usar]</li><li>• Administración de despliegues. [Usar]</li><li>• Análisis de requerimientos y herramientas para modelado del diseño. [Usar]</li><li>• Herramientas de <i>testing</i> incluyendo herramientas de análisis estático y dinámico. [Usar]</li><li>• Entornos de programación que automatizan el proceso de construcción de partes de programa (ejem., construcciones automatizadas)<ul style="list-style-type: none"><li>– Integración continua.</li></ul>[Usar]</li><li>• Mecanismos y conceptos de herramientas de integración. [Usar]</li></ul>
<b>Readings :</b> [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	



Unit 2: Verificación y Validación de Software (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Verificación y validación de conceptos.</li> <li>• Inspecciones, revisiones, auditorias.</li> <li>• Tipos de pruebas, incluyendo la interfaz 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 envios por defecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Distinguir entre la validación y verificación del programa [Usar]</li> <li>• Describir el papel que las herramientas pueden desempeñar en la validación de software [Usar]</li> <li>• Realizar, como parte de una actividad de equipo, una inspección de un segmento de código de tamaño medio [Usar]</li> <li>• Describir y distinguir entre diferentes tipos y niveles de pruebas (unitaria, integración, sistemas y aceptación) [Usar]</li> <li>• Describir técnicas para identificar casos de prueba representativos para integración, regresión y pruebas del sistema [Usar]</li> <li>• Crear y documentar un conjunto de pruebas para un segmento de código de mediano tamaño [Usar]</li> <li>• Describir cómo seleccionar buenas pruebas de regresión y automatizarlas [Usar]</li> <li>• Utilizar una herramienta de seguimiento de defectos para manejar defectos de software en un pequeño proyecto de software [Usar]</li> <li>• Discutir las limitaciones de las pruebas en un dominio particular [Usar]</li> <li>• Evaluar un banco de pruebas (<i>a test suite</i>) para un segmento de código de tamaño medio [Usar]</li> <li>• Comparar los enfoques estáticos y dinámicos para la verificación [Usar]</li> <li>• Identificar los principios fundamentales de los métodos de desarrollo basado en pruebas y explicar el papel de las pruebas automatizadas en estos métodos [Usar]</li> <li>• Discutir las limitaciones de las pruebas en un dominio particular [Usar]</li> <li>• Describir las técnicas para la verificación y validación de los artefactos de no código [Usar]</li> <li>• Describir los enfoques para la estimación de fallos [Usar]</li> <li>• Estimar el número de fallos en una pequeña aplicación de software basada en la densidad de defectos y siembra de errores [Usar]</li> <li>• Realizar una inspección o revisión del de código fuente de un software para un proyecto de software de tamaño pequeño o mediano [Usar]</li> </ul>
<b>Readings :</b> [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

**Unit 3: Evolución de Software (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>	<ul style="list-style-type: none"><li>• Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Usar]</li><li>• Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar]</li><li>• Usar refactorización en el proceso de modificación de un componente de software [Usar]</li><li>• Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Usar]</li><li>• Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Usar]</li><li>• Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Usar]</li></ul>
<b>Readings :</b> [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 4: Gestión de Proyectos de Software (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>	<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> </ul>
<b>Readings :</b> [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the

different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Blu92] Bruce I. Blum. *Software Engineering: A Holistic View*. 7th. Oxford University Press US, May 1992.
- [Mon96] Carlo Montangero. *Software Process Technology*. Springer, Sept. 1996.
- [Con00] R Conradi. *Software Process Technology*. Springer, Mar. 2000.
- [WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications*. CRC Press, Apr. 2000.
- [Amb01] Vincenzo Ambriola. *Software Process Technology*. Springer, July 2001.
- [PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing*. Marcel Dekker, Jan. 2001.
- [WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process*. Prentice Hall, Aug. 2002.
- [Oqu03] Flavio Oquendo. *Software Process Technology*. Springer, Sept. 2003.
- [Key04] Jessica Keyes. *Software Configuration Management*. CRC Press, Feb. 2004.
- [Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2004.
- [Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, Jan. 2004.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS393. Information systems (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS393. Information systems
- 2.2 Semester : 6<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS291. Software Engineering I. (5<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Analyze techniques for the correct implementation of scalable, robust, reliable and efficient information systems in organizations.

**5. GOALS**

- Implement correctly (scalable, robust, reliable and efficient) Information Systems in organizations.

**6. COMPETENCES**

- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**7. TOPICS**

Unit 1: Introduction (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Introduction to information management.</li><li>• Software for information management.</li><li>• Technology for information management.</li></ul>	<ul style="list-style-type: none"><li>• Correctly apply technology for information management [Evaluar]</li></ul>
Readings : [Som17], [PM15], [LL17]	

<b>Unit 2: Strategy (15 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Strategy for information management.</li> <li>• Strategy for knowledge management</li> <li>• Strategy for information system.</li> </ul>	<ul style="list-style-type: none"> <li>• Apply and evaluate correctly management strategies [Evaluar]</li> </ul>
<b>Readings :</b> [Som17], [PM15]	

<b>Unit 3: Implementation (15 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Management Information Systems Development.</li> <li>• Change management</li> <li>• Information Architecture</li> </ul>	<ul style="list-style-type: none"> <li>• Implement and correctly evaluate implementation strategies [Evaluar]</li> </ul>
<b>Readings :</b> [Som17], [PM15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS311. Computer Security (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |  |
|-----------------------------------|--|
| <b>2.1 Course</b>                 | : CS311. Computer Security                                   |
| <b>2.2 Semester</b>               | : 7 <sup>th</sup> Semester.                                  |
| <b>2.3 Credits</b>                | : 3  |
| <b>2.4 Horas</b>                  | : 1 HT; 4 HP;  |
| <b>2.5 Duration of the period</b> | : 16 weeks   |
| <b>2.6 Type of course</b>         | : Mandatory  |
| <b>2.7 Learning modality</b>      | : Face to face   |
| <b>2.8 Prerequisites</b>          | : CS231. Networking and Communication. (6 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Nowadays, information is one of the most valuable assets in any organization. This course is oriented to be able to provide the student with the security elements oriented to protect the Information of the organization and mainly to be able to foresee the possible problems related to this heading. This subject involves the development of a preventive attitude on the part of the student in all areas related to software development.

**5. GOALS**

- Discuss at an intermediate intermediate level the fundamentals of Computer Security.
- Provide different aspects of the malicious code.
- That the student knows the concepts of cryptography and security in computer networks.
- Discuss and analyze together with the student the aspects of Internet Security.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

**7. TOPICS**

**Unit 1: Fundamentos y Conceptos en Seguridad (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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(Confidenciabilidad, 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 autenticación, 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>
<b>Readings :</b> [WL14]	



**Unit 2: Principios de Diseño Seguro (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [WL14]	

**Unit 3: Programación Defensiva (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 desallorar 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>
<b>Readings :</b> [WL14]	

Unit 4: Ataques y Amenazas (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Atacante metas, capacidades y motivaciones (como economía sumergida, el espionaje digital, la guerra cibernética, 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>
Readings : [WL14]	

Unit 5: Seguridad de Red (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [WL14]	

**Unit 6: Criptografía (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 trampa, 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>
<b>Readings :</b> [WL14]	

**Unit 7: Seguridad en la Web (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [WL14]	

**Unit 8: Seguridad de plataformas (25 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>

**Readings :** [WL14]

<b>Unit 9: Investigación digital (Digital Forensics) (25 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<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>
<b>Readings :</b> [WL14]	

<b>Unit 10: Seguridad en Ingeniería de Software (25 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<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>
<b>Readings :</b> [WL14]	

## **8. WORKPLAN**

### **8.1 Methodology**

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### **8.2 Theory Sessions**

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### **8.3 Practical Sessions**

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## **9. EVALUATION SYSTEM**

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

## **10. BASIC BIBLIOGRAPHY**

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

FG211-ACM. Professional Ethics (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : FG211-ACM. Professional Ethics
- 2.2 Semester : 7<sup>th</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 2 HT; 2 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course introduces the ethical principles and professional responsibilities in computing, based on the ACM Code of Ethics and international standards. Students will analyze real-world cases, evaluate ethical dilemmas, and apply decision-making frameworks in technological contexts, considering social impact, privacy, security, and sustainability.

**5. GOALS**

- Analyze fundamental ethical principles in computing according to ACM/IEEE.
- Critically evaluate ethical dilemmas in technology development.
- Apply the ACM Code of Ethics to real-world case studies.

**6. COMPETENCES**

- 3) Communicate effectively in a variety of professional contexts.. (Familiarity)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- AG-C02**) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)
- AG-C03**) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Usage)
- AG-C04**) Communication: Communicates effectively in complex computing activities. (Familiarity)

**7. TOPICS**

<b>Unit 1: Fundamentals of Computing Ethics (8 hours)</b>	
<b>Competences Expected: 4,AG-C02</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Introduction to professional ethics and morality.</li> <li>• ACM and IEEE Code of Ethics.</li> <li>• Legal and social responsibilities of computing professionals.</li> </ul>	<ul style="list-style-type: none"> <li>• Explain the principles of the ACM Code of Ethics [Familiarizarse].</li> <li>• Discuss professional responsibilities in technological contexts [Usar].</li> <li>• Analyze ethical conflicts in hypothetical cases [Evaluar].</li> </ul>
<b>Readings :</b> [ACM-Ethics2018], [IEEE-Ethics2020]	

<b>Unit 2: Privacy, Security, and Digital Rights (10 hours)</b>	
<b>Competences Expected: 4,AG-C02</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Data privacy regulations (GDPR, Data Protection Laws).</li> <li>• Cybersecurity and ethical hacking.</li> <li>• Intellectual property rights and open-source software.</li> </ul>	<ul style="list-style-type: none"> <li>• Compare global privacy regulations [Usar].</li> <li>• Evaluate ethical implications of cybersecurity vulnerabilities [Evaluar].</li> <li>• Debate software licensing and open access [Usar].</li> </ul>
<b>Readings :</b> [GDPR2018], [ACM-Code2018]	

<b>Unit 3: Ethics in AI and Algorithms (10 hours)</b>	
<b>Competences Expected: 5,AG-C03</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Algorithmic bias and discrimination.</li> <li>• Transparency and accountability in autonomous systems.</li> <li>• Social impact of automation.</li> </ul>	<ul style="list-style-type: none"> <li>• Identify biases in datasets and algorithms [Usar].</li> <li>• Propose solutions for ethical AI systems [Evaluar].</li> <li>• Debate the impact of automation on employment [Usar].</li> </ul>
<b>Readings :</b> [AI-Ethics2021], [Bostrom2014]	

<b>Unit 4: Case Studies and Workshops (20 hours)</b>	
<b>Competences Expected: 3,AG-C04</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Analysis of historical cases (e.g., Cambridge Analytica, data breaches).</li> <li>• Role-playing: ethical decision-making in teams.</li> <li>• Workshop on writing ethical reports.</li> </ul>	<ul style="list-style-type: none"> <li>• Resolve complex cases using the ACM ethical framework [Evaluar].</li> <li>• Collaborate in teams to propose ethical solutions [Usar].</li> <li>• Draft professional reports on ethical dilemmas [Evaluar].</li> </ul>
<b>Readings :</b> [ACM-Cases2020], [IEEE-Cases2019]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

## **8.2 Theory Sessions**

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

## **8.3 Practical Sessions**

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## **9. EVALUATION SYSTEM**

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

## **10. BASIC BIBLIOGRAPHY**



National University of Engineering (UNI)  
School of Computer Science  
Syllabus 2024-II

1. COURSE

CS262. Machine learning (Elective)

2. GENERAL INFORMATION

- 2.1 Course : CS262. Machine learning
- 2.2 Semester : 7<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Elective
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS261. Artificial Intelligence. (6<sup>th</sup> Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Write justification for this course here ...

5. GOALS

- Write your first goal here.
- Write your second goal here.
- Just in case you need more goals write them here

6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Familiarity)

7. TOPICS

Unit 1: title for the unit goes here (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Topic1</li><li>• Topic2</li><li>• Topic3</li></ul>	<ul style="list-style-type: none"><li>• Learning outcome1 [Leveforthislearningoutcome].</li><li>• Apply computing in complex problems [Usar].</li><li>• Create a search engine [Evaluar].</li><li>• Study data structures [Familiarizarse].</li></ul>
Readings : [Bibitem1], [Bibitem2]	

<b>Unit 2: another unit goes here (1 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Topic1</li> </ul>	<ul style="list-style-type: none"> <li>• Learning outcome xyz [Levelforthislearningoutcome].</li> </ul>
<b>Readings : [Bibitem3], [Bibitem1]</b>	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS281. Computing in Society (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: CS281. Computing in Society
<b>2.2 Semester</b>	: 8 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 2
<b>2.4 Horas</b>	: 2 HT;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

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. GOALS**

- 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. COMPETENCES**

- 3) Communicate effectively in a variety of professional contexts.. (Familiarity)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

**7. TOPICS**

<b>Unit 1: Historia (2 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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 [Familiarizarse]</li> </ul>
<b>Readings :</b> [LL04], [McL00]	

**Unit 2: Contexto Social (4 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>• Discuta 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>
<b>Readings :</b> [LL04], [McL00]	



**Unit 3: Herramientas de Análisis (2 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>Readings :</b> [LL04], [McL00]	

Unit 4: Ética Profesional (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 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>
<b>Readings :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

**Unit 5: Propiedad Intelectual (4 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>Readings :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

**Unit 6: Privacidad y Libertades Civiles (4 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>
<b>Readings :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

**Unit 7: Políticas de seguridad, Leyes y crímenes computacionales (2 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 ciberterrorismo 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>
<b>Readings :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 8: Economía de la Computación (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>• Analisis costo/beneficio de trabajos con consideraciones para manufactura, hardware, software e implicaciones de ingeniería.</li> <li>• Costo estimado versus costo actual in relacion 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>Readings :</b> [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

## 1. COURSE

CS391. Software Engineering III (Mandatory)

## 2. GENERAL INFORMATION

<b>2.1 Course</b>	: CS391. Software Engineering III
<b>2.2 Semester</b>	: 7 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: CS292. Software Engineering II. (6 <sup>th</sup> Sem)

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

Software development requires the use of best development practices, IT project management, equipment management And efficient and rational use of quality assurance frameworks, these elements are key and transversal during the whole productive process. The construction of software contemplates the implementation and use of processes, methods, models and tools that allow to achieve the realization of the quality attributes of a product.

## 5. GOALS

- Understand and implement the fundamental concepts of project management and software equipment management.
- Understand the fundamentals of project management, including its definition, scope, and need for project management in the modern organization.
- Students have to understand the fundamental concepts of CMMI, PSP, TSP to be adopted in software projects.
- Describe and understand quality assurance models as a key framework for the success of IT projects.

## 6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

## 7. TOPICS

**Unit 1: Evolución de Software (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>	<ul style="list-style-type: none"><li>• Identificar los problemas principales asociados con la evolución del software y explicar su impacto en el ciclo de vida del software [Familiarizarse]</li><li>• Estimar el impacto del cambio de requerimientos en productos existentes de tamaño medio [Usar]</li><li>• Usar refactorización en el proceso de modificación de un componente de software [Usar]</li><li>• Estudiar los desafíos de mejorar sistemas en un entorno cambiante [Familiarizarse]</li><li>• Perfilar los procesos de pruebas de regresión y su rol en el manejo de versiones [Familiarizarse]</li><li>• Estudiar las ventajas y desventajas de diferentes tipos de niveles de confiabilidad [Familiarizarse]</li></ul>
<b>Readings :</b> [PM15], [Som17]	



**Unit 2: Gestión de Proyectos de Software (10 hours)****Competences Expected:**

Topics	Learning Outcomes
<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> </ul>	<ul style="list-style-type: none"> <li>• Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Familiarizarse]</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 [Familiarizarse]</li> <li>• Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Familiarizarse]</li> <li>• Describir las diferentes categorías de riesgo en los sistemas de software [Familiarizarse]</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 [Familiarizarse]</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 necesita 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) [Familiarizarse]</li> </ul>
<b>Readings :</b> [PM15], [Som17]	

**Unit 3: Gestión de Proyectos de Software (8 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"> <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>	<ul style="list-style-type: none"> <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 [Evaluar]</li> <li>• Identificar riesgos y describir enfoques para manejar riesgos (evitar, aceptar, transferir, mitigar) y caracterizar fortalezas y defectos para cada uno [Familiarizarse]</li> <li>• Explicar cómo el riesgo afecta las decisiones en el proceso de desarrollo de software [Usar]</li> <li>• Identificar los riesgos de seguridad para un sistema de software [Usar]</li> <li>• Demostrar un enfoque sistemático para la tarea de identificar los peligros y riesgos en una situación particular [Usar]</li> <li>• Aplicar los principios básicos del manejo de riesgos en una variedad de escenarios simples incluyendo una situación de seguridad [Usar]</li> <li>• Dirigir un análisis de costo/beneficio para el enfoque de mitigación de riesgos [Usar]</li> <li>• Identificar y analizar alguno de los riesgos para un sistema entero que surgen de aspectos distintos del software [Usar]</li> </ul>
<b>Readings :</b> [PM15], [Som17]	

**Unit 4: Procesos de Software (12 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 [Usar]</li> <li>• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar]</li> <li>• Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Usar]</li> <li>• Diferenciar entre las fases de desarrollo de software [Usar]</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 [Usar]</li> <li>• Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar]</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 [Usar]</li> <li>• Describir el objetivo y similitudes fundamentales entre los enfoques de mejora de procesos [Usar]</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>• Evaluar un esfuerzo de desarrollo y recomendar cambios potenciales al participar en la mejora de procesos (usando un modelo como PSP) o involucración en una retrospectiva de un proyecto [Usar]</li> <li>• Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Usar]</li> <li>• Describir varias métricas de procesos para la evaluación y el control de un proyecto [Usar]</li> <li>• Usar las medidas en proyecto para describir el estado actual de un proyecto [Usar]</li> </ul>

**Readings :** [PM15], [Som17]

Unit 5: Estándares ISO/IEC (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• ISO 9001:2001.</li> <li>• ISO 9000-3.</li> <li>• ISO/IEC 9126.</li> <li>• ISO/IEC 12207.</li> <li>• ISO/IEC 15939.</li> <li>• ISO/IEC 14598.</li> <li>• ISO/IEC 15504-SPICE.</li> <li>• IT Mark.</li> <li>• SCRUM.</li> <li>• SQuaRE.</li> <li>• CISQ.</li> </ul>	<ul style="list-style-type: none"> <li>• Learn and apply correctly standards and international standards . [Usar]</li> </ul>
Readings : [Som17], [PM15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS3P1. Parallel and Distributed Computing (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS3P1. Parallel and Distributed Computing
- 2.2 Semester : 8<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
  - CS212. Analysis and Design of Algorithms. (5<sup>th</sup> Sem)
- 2.8 Prerequisites :
  - CS231. Networking and Communication. (6<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The last decade has brought explosive growth in computing with multiprocessors, including Multi-core processors and distributed data centers. As a result, computing parallel and distributed has become a widely elective subject to be one of the main components in the mesh studies in computer science undergraduate. Both parallel and distributed computing the simultaneous execution of multiple processes, whose operations have the potential to intercalate in a complex way. Parallel and distributed computing builds on foundations in many areas, including understanding the fundamental concepts of systems, such as: concurrency and parallel execution, consistency in state / memory manipulation, and latency. The communication and coordination between processes has its foundations in the passage of messages and models of shared memory of computing and algorithmic concepts like atomicity, consensus and conditional waiting. Achieving acceleration in practice requires an understanding of parallel algorithms, strategies for decomposition problem, systems architecture, implementation strategies and analysis of performance. Distributed systems highlight the problems of security and tolerance to Failures, emphasize the maintenance of the replicated state and introduce additional problems in the field of computer networks.

**5. GOALS**

- That the student is able to create parallel applications of medium complexity by efficiently leveraging machines with multiple cores.
- That the student is able to compare sequential and parallel applications.
- That the student is able to convert, when the situation warrants, sequential applications to parallel efficiently

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

**Unit 1: Fundamentos de paralelismo (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<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>Readings :</b> [Pac11], [Mat14], [quinnz], [Geo10]	

<b>Unit 2: Arquitecturas paralelas (12 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Procesadores mutlinúcleo.</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>
<b>Readings :</b> [Pac11], [KH13], [SK10], [Geo10]	

<b>Unit 3: Descomposición en paralelo (18 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>• Descomposición 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>
<b>Readings :</b> [Pac11], [Mat14], [Qui03], [Geo10]	

**Unit 4: Comunicación y coordinación (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Memoria Compartida.</li> <li>• La consistencia, y su papel en los lenguaje de programación garantías 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) barreras, 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 mutua 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 semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]</li> </ul>
<b>Readings :</b> [Pac11], [Mat14], [Qui03], [Geo10]	



**Unit 5: Análisis y programación de algoritmos paralelos (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 Algoritmicos 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 analogo secuencial [Usar]</li> <li>• Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) via 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>
<b>Readings :</b> [Mat14], [Qui03], [Geo10]	

Unit 6: Desempeño en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
<b>Readings :</b> [Pac11], [Mat14], [KH13], [SK10], [Geo10]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003.
- [Geo10] Gerhard Wellein Georg Hager. *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>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

### 1. COURSE

CS401. Capstone Project I (Mandatory)

### 2. GENERAL INFORMATION

2.1 Course	:	CS401. Capstone Project I
2.2 Semester	:	8 <sup>th</sup> Semester.
2.3 Credits	:	4
2.4 Horas	:	1 HT; 5 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Mandatory
2.7 Learning modality	:	Face to face
2.8 Prerequisites	:	CS369. Topics in Artificial Intelligence. (9 <sup>th</sup> Sem)

### 3. PROFESSORS

Meetings after coordination with the professor

### 4. INTRODUCTION TO THE COURSE

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. GOALS

- 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 concluido donde se contenga, opcionalmente, los resultados experimentales de la(s) técnica(s) estudiada(s).

### 6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)

---

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

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

- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

## 7. TOPICS

Unit 1: (60 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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 acorde con las características que las conferencias de esta área exigen[Usar]</li> </ul>
Readings : [IEE08], [Ass08], [Cit08]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Ass08] Association for Computing Machinery. *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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

FG120FCCS. General Economy (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	: FG120FCCS. General Economy
<b>2.2 Semester</b>	: 8 <sup>th</sup> Semester.
<b>2.3 Credits</b>	: 3
<b>2.4 Horas</b>	: 2 HT; 2 HP;
<b>2.5 Duration of the period</b>	: 16 weeks
<b>2.6 Type of course</b>	: Mandatory
<b>2.7 Learning modality</b>	: Face to face
<b>2.8 Prerequisites</b>	: None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Understanding Peruvian reality is essential for future professionals, including those in computer science, so they can contribute to the country's development in an informed and responsible manner. This course analyzes the historical, social, economic, and political aspects of Peru, with an emphasis on current challenges and opportunities.

**5. GOALS**

- Analyze the historical processes that have shaped Peruvian national reality.
- Understand the main social, economic, and political challenges facing Peru.
- Identify development opportunities and the role of technology in building a better future for the country.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

**AG-C01)** The Professional and the World: Analyzes and evaluates the impact of solutions to complex computing problems on the sustainable development of society. (Assessment)

**AG-C07)** Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)

**7. TOPICS**

<b>Unit 1: History of Peru (8 hours)</b>	
<b>Competences Expected: 3,AG-C01,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Pre-Hispanic Peru.</li> <li>• Colonial era.</li> <li>• Republican Peru: 19th and 20th centuries.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the main periods of Peruvian history. [Familiarizarse]</li> <li>• Analyze the key events that have shaped national identity. [Usar]</li> <li>• Explain the influence of the past on Peru's current reality. [Evaluar]</li> </ul>
<b>Readings :</b> [KlarÁfn2012], [Cotler2005]	

<b>Unit 2: Cultural and Social Diversity (8 hours)</b>	
<b>Competences Expected: 3,AG-C01,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Cultural diversity of Peru.</li> <li>• Ethnic and linguistic groups.</li> <li>• Social inequality and exclusion.</li> </ul>	<ul style="list-style-type: none"> <li>• Identify the different cultural groups that make up Peru. [Familiarizarse]</li> <li>• Analyze the causes and consequences of social inequality. [Usar]</li> <li>• Propose solutions to promote social inclusion. [Evaluar]</li> </ul>
<b>Readings :</b> [Portocarrero2001], [Manrique1988]	

<b>Unit 3: Peruvian Economy (8 hours)</b>	
<b>Competences Expected: 1,AG-C01,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Economic structure of Peru.</li> <li>• Economic sectors: mining, agriculture, industry, services.</li> <li>• International trade.</li> <li>• Economic challenges and sustainable development.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the economic structure of Peru. [Familiarizarse]</li> <li>• Analyze the importance of the different economic sectors. [Usar]</li> <li>• Evaluate economic policies and their impact on sustainable development. [Evaluar]</li> </ul>
<b>Readings :</b> [Dancourt2013], [Hunt2018]	

<b>Unit 4: Peruvian Political System (8 hours)</b>	
<b>Competences Expected: 1,4,AG-C07</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Organization of the Peruvian State.</li> <li>• Democracy and governance.</li> <li>• Corruption and transparency.</li> </ul>	<ul style="list-style-type: none"> <li>• Describe the organization of the Peruvian State. [Familiarizarse]</li> <li>• Analyze the challenges to democracy and governance in Peru. [Usar]</li> <li>• Evaluate the importance of transparency and the fight against corruption. [Evaluar]</li> </ul>
<b>Readings :</b> [Tanaka2018], [Quiroz2013]	

Unit 5: Technology and Development in Peru (8 hours)	
Competences Expected: 1,AG-C01	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• The role of technology in Peru's development.</li> <li>• Digital divide and digital inclusion.</li> <li>• Technological innovation and entrepreneurship.</li> </ul>	<ul style="list-style-type: none"> <li>• Analyze the importance of technology for Peru's economic and social development. [Familiarizarse]</li> <li>• Propose solutions to reduce the digital divide. [Usar]</li> <li>• Evaluate the potential of technological innovation and entrepreneurship in Peru. [Evaluar]</li> </ul>
Readings : [PlanNacionaldeCompetitividad2019]	

Unit 6: Challenges and Opportunities for the Future (8 hours)	
Competences Expected: 1,AG-C01	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Sustainable development.</li> <li>• Peru's vision for 2050.</li> <li>• The role of professionals in building the future.</li> </ul>	<ul style="list-style-type: none"> <li>• Analyze the challenges Peru faces in achieving sustainable development. [Familiarizarse]</li> <li>• Understand the long-term vision for the country's development. [Usar]</li> <li>• Reflect on the role of professionals in building a better future for Peru. [Evaluar]</li> </ul>
Readings : [CEPLAN2021]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

EX301FCCS. Extracurricular Activities (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : EX301FCCS. Extracurricular Activities
- 2.2 Semester : 8<sup>th</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 2 HT; 2 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course records the student's participation in activities that complement their academic training, such as workshops, conferences, social outreach, volunteering, or technical skills. Its objective is to promote comprehensive development, soft skills, and social commitment.

**5. GOALS**

- Participate in activities that complement their professional training.
- Develop communication and teamwork skills.
- Contribute to initiatives with a social or technical impact.

**6. COMPETENCES**

- 3) Communicate effectively in a variety of professional contexts.. (Familiarity)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.  
(Usage)
- AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams.  
(Usage)
- AG-C04) Communication: Communicates effectively in complex computing activities. (Familiarity)

**7. TOPICS**



<b>Unit 1: Extracurricular Activities (48 hours)</b>	
<b>Competences Expected: 3,5,AG-C03,AG-C04</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Technical or interdisciplinary workshops.</li> <li>• Attendance at conferences, seminars, or talks.</li> <li>• Social outreach (volunteering, mentoring).</li> <li>• Academic skills (hackathons, science fairs).</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate participation in at least three activities [Familiarity].</li> <li>• Prepare reflective reports on experiences [Usage].</li> <li>• Collaborate in teams for social or technical projects [Evaluat].</li> </ul>
<b>Readings :</b> [Com19], [Com22]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[Com19] Association for Computing Machinery (ACM). *Guidelines for Extracurricular Activities in Computing*. Tech. rep. 2019. URL: <https://www.acm.org/education/curricula-recommendations>.

[Com22] ABET Computing Accreditation Commission. *ABET Criteria for Student Professional Development*. Tech. rep. 2022. URL: <https://www.abet.org/accreditation/>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS361. Computational Vision (Elective)

**2. GENERAL INFORMATION**

- 2.1 Course : CS361. Computational Vision
- 2.2 Semester : 8<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Elective
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS262. Machine learning. (7<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course covers fundamental techniques for automated analysis of digital images, essential for applications like medical diagnosis, autonomous vehicles, and surveillance systems. Aligns with ACM/IEEE-CS standards for computer vision.

**5. GOALS**

- Implement feature extraction and object recognition algorithms using OpenCV/Python.
- Evaluate deep learning methods for semantic segmentation (e.g., Mask R-CNN).

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

**AG-C08) Problem Analysis:** Identifies, formulates, and analyzes complex computing problems. (Usage)

**AG-C09) Solution Design and Development:** Designs, implements, and evaluates solutions for complex computing problems. (Usage)

**AG-C11) Tool Usage:** Applies modern computing tools in problem-solving. (Familiarity)

**7. TOPICS**

<b>Unit 1: Digital Image Fundamentals (16 hours)</b>	
<b>Competences Expected: 1,6,AG-C08,AG-C11</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Spatial filtering (Gaussian, Sobel)</li> <li>• Geometric and morphological transformations</li> <li>• Color spaces (RGB, HSV, LAB)</li> </ul>	<ul style="list-style-type: none"> <li>• Apply basic image processing operations [Usar]</li> <li>• Calibrate filter parameters for real-world cases [Evaluuar]</li> </ul>
<b>Readings :</b> [Sze10], [GW18]	

<b>Unit 2: Epipolar Geometry and Reconstruction (16 hours)</b>	
<b>Competences Expected: 2,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Fundamental and essential matrices</li> <li>• Triangulation and structure-from-motion</li> <li>• Point clouds with Open3D</li> </ul>	<ul style="list-style-type: none"> <li>• Implement 3D reconstruction pipelines [Usar]</li> <li>• Document technical results in reports [Evaluuar]</li> </ul>
<b>Readings :</b> [HZ04], [Forsyth22]	

<b>Unit 3: Neural Networks for Vision (16 hours)</b>	
<b>Competences Expected: 2,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• CNN architectures (ResNet, YOLO)</li> <li>• Transfer learning with TensorFlow</li> <li>• Semantic segmentation (U-Net)</li> </ul>	<ul style="list-style-type: none"> <li>• Train models for image classification [Usar]</li> <li>• Collaborate in teams for integrated projects [Usar]</li> </ul>
<b>Readings :</b> [Goodfellow16], [He+17]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [HZ04] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. DOI: 10.1017/CB09780511811685.
- [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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS370. Big Data (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS370. Big Data
- 2.2 Semester : 9<sup>th</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 1 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
  - CS272. Databases II. (5<sup>th</sup> Sem)
- 2.8 Prerequisites :
  - CS3P1. Parallel and Distributed Computing . (8<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Nowadays, knowing scalable approaches to processing and storing large volumes of information (terabytes, petabytes and even exabytes) is fundamental in computer science courses. Every day, every hour, every minute generates a large amount of information which needs to be processed, stored, analyzed.

**5. GOALS**

- That the student is able to create parallel applications to process large volumes of information
- That the student is able to compare the alternatives for the processing of big data
- That the student is able to propose architectures for a scalable application

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Introducción a Big Data (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Overview on Cloud Computing</li><li>• Distributed File System Overview</li><li>• Overview of the MapReduce programming model</li></ul>	<ul style="list-style-type: none"><li>• Explain the concept of Cloud Computing from the point of view of Big Data[Familiarizarse]</li><li>• Explain the concept of Distributed File System [Familiarizarse]</li><li>• Explain the concept of the MapReduce programming model[Familiarizarse]</li></ul>
Readings : [Cou+11]	

Unit 2: Hadoop (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Hadoop overview.</li> <li>• History.</li> <li>• Hadoop Structure.</li> <li>• HDFS, Hadoop Distributed File System.</li> <li>• Programming Model MapReduce</li> </ul>	<ul style="list-style-type: none"> <li>• Understand and explain the Hadoop suite [Familiarizarse]</li> <li>• Implement solutions using the MapReduce programming model. [Usar]</li> <li>• Understand how data is saved in the HDFS. [Familiarizarse]</li> </ul>
Readings : [HDF11], [BVS13]	

Unit 3: Procesamiento de Grafos en larga escala (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Pregel: A System for Large-scale Graph Processing.</li> <li>• Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud.</li> <li>• Apache Giraph is an iterative graph processing system built for high scalability.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand and explain the architecture of the Pregel project. [Familiarizarse]</li> <li>• Understand the GraphLab project architecture. [Familiarizarse]</li> <li>• Understand the architecture of the Giraph project. [Familiarizarse]</li> <li>• Implement solutions using Pregel, GraphLab or Giraph. [Usar]</li> </ul>
Readings : [Low+12], [Mal+10], [Bal+08]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Bal+08] Shumeet Baluja et al. "Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph". In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, 2008, pp. 895–904. DOI: 10.1145/1367497.1367618. URL: <http://doi.acm.org/10.1145/1367497.1367618>.
- [Mal+10] Grzegorz Malewicz et al. "Pregel: A System for Large-scale Graph Processing". In: SIGMOD '10 (2010), pp. 135–146. DOI: 10.1145/1807167.1807184. URL: <http://doi.acm.org/10.1145/1807167.1807184>.
- [Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011.
- [HDF11] Kai Hwang, Jack Dongarra, and Geoffrey C. Fox. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

- [Low+12] Yucheng Low et al. “Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud”. In: *Proc. VLDB Endow.* 5.8 (Apr. 2012), pp. 716–727. DOI: 10.14778/2212351.2212354. URL: <http://dx.doi.org/10.14778/2212351.2212354>.
- [BVS13] Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi. *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS400FCCS. Pre-professional internships (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	CS400FCCS. Pre-professional internships
<b>2.2 Semester</b>	:	9 <sup>th</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course enables students to apply knowledge acquired during their academic training in a real work environment, under the supervision of a company and the university. Professional internships are essential to develop technical, ethical, and teamwork competencies, ensuring students gain hands-on experience in projects related to Computing.

**5. GOALS**

- Evaluate student performance in a real work environment, applying Computing principles.
- Develop professional, ethical, and teamwork skills in an industry setting.
- Ensure internship activities align with the program's learning outcomes.

**6. COMPETENCES**

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
  - 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Assessment)
- AG-C02) Ethics:** Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)
- AG-C03) Individual and Teamwork:** Performs effectively as an individual and as a member or leader in diverse teams. (Assessment)

**7. TOPICS**

Unit 1: Performance Evaluation in Professional Settings (48 hours)	
Competences Expected: 4,5,AG-C02,AG-C03	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Integration into a professional work team.</li> <li>• Application of technical skills in real-world projects.</li> <li>• Assessment of ethical and professional conduct.</li> <li>• Reporting and feedback from the company and university.</li> </ul>	<ul style="list-style-type: none"> <li>• Demonstrate technical skills in a professional environment [Evaluar].</li> <li>• Apply ethical and professional principles in their work [Evaluar].</li> <li>• Collaborate effectively in teams and communicate professionally [Evaluar].</li> </ul>
Readings : [ACMInternshipGuide], [IEEEProfessionalPractices]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS402. Capstone Project II (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS402. Capstone Project II
- 2.2 Semester : 9<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 1 HT; 5 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS401. Capstone Project I. (8<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course aims to allow the student to carry out a study of the state of the art of a topic chosen by the student for his thesis.

**5. GOALS**

- That the student carries out an initial investigation in a specific subject realizing the study of the state of the art of the chosen subject.
- That the student shows mastery in the subject of the line of investigation chosen
- That the student choose a teacher who dominates the research chosen as an advisor.
- The deliverables of this course are:

**Avance parcial:** Solid bibliography and progress of a Technical Reporto.

**Final:** Technical Report with preliminary comparative experiments that demonstrate that the student already knows the existing techniques in the area of his project and choose a teacher who dominates the area of his project as an adviser of his project.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

## 7. TOPICS

Unit 1: Lifting the state of the art (60 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Perform an in-depth study of the state of the art in a certain topic in the area of Computation.</li> <li>• Writing technical articles in computing.</li> </ul>	<ul style="list-style-type: none"> <li>• Make a bibliographical survey of the state of the art of the chosen subject (this probably means 1 or 2 chapters of theoretical framework in addition to the introduction that is chapter I of the thesis) [Usar]</li> <li>• Writing a latex document in paper format with higher quality than Project I (master tables, figures, equations, indices, bibtex, cross references, citations, pstricks) [Usar]</li> <li>• Try to make presentations using prosper [Usar]</li> <li>• Show basic experiments [Usar]</li> <li>• Choose an advisor who dominates the research area [Usar]</li> </ul>
Readings : [IEE08], [Ass08], [Cit08]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Ass08] Association for Computing Machinery. *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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CB309. Bioinformatics (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CB309. Bioinformatics
- 2.2 Semester : 9<sup>th</sup> Semester.
- 2.3 Credits : 2
- 2.4 Horas : 1 HT; 2 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites :
  - CS212. Analysis and Design of Algorithms. (5<sup>th</sup> Sem)
  - MA307. Mathematics applied to computing. (6<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.

In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.

This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

**5. GOALS**

- That the student has a solid knowledge of molecular biological problems that challenge computing.
- That the student is able to abstract the essence of the various biological problems to pose solutions using their knowledge of Computer Science

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

**7. TOPICS**

<b>Unit 1: Introduction to Molecular Biology (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Review of organic chemistry: molecules and macromolecules, sugars, nucleic acids, nucleotides, RNA, DNA, proteins, amino acids and levels of structure in proteins.</li> <li>• The Dogma of Life: From DNA to Proteins, Transcription, Translation, Protein Synthesis.</li> <li>• Genome study: Maps and sequences, specific techniques</li> </ul>	<ul style="list-style-type: none"> <li>• Achieve a general knowledge of the most important topics in Molecular Biology. [Familiarizarse]</li> <li>• Understand that biological problems are a challenge to the computational world. [Evaluar]</li> </ul>
<b>Readings :</b> [CB00], [SM97]	

<b>Unit 2: Sequence Comparison (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Sequences of nucleotides and amino acid sequences.</li> <li>• Sequence alignment, paired alignment problem, exhaustive search, Dynamic programming, global alignment, local alignment, gaps penalty</li> <li>• Comparison of multiple sequences: sum of pairs, complexity analysis by dynamic programming, alignment heuristics, star algorithm, progressive alignment algorithms.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand and solve the problem of aligning a pair of sequences. [Usar]</li> <li>• Understand and solve the problem of multiple sequence alignment. [Usar]</li> <li>• Know the various algorithms for aligning existing sequences in the literature . [Familiarizarse]</li> </ul>
<b>Readings :</b> [CB00], [SM97], [Pev00]	

<b>Unit 3: Phylogenetic Trees (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Phylogeny: Introduction and phylogenetic relations</li> <li>• Phylogenetic trees: definition, type of trees, problem of search and reconstruction of trees</li> <li>• Reconstruction methods: parsimony methods, distance methods, maximum likelihood methods, confidence of reconstructed trees</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concept of phylogeny, phylogenetic trees and the methodological difference between biology and molecular biology. [Familiarizarse]</li> <li>• Understand the problem of the reconstruction of phylogenetic trees, to know and apply the main algorithms for the reconstruction of phylogenetic trees. [Evaluar]</li> </ul>
<b>Readings :</b> [CB00], [SM97], [Pev00]	

<b>Unit 4: DNA Sequence Assembling (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Biological basis: ideal case, difficulties, alternative methods for DNA sequencing</li> <li>• Formal Assembly Models: Shortest Common Superstring, Reconstruction, Multicontig</li> <li>• Algorithms for sequence assembly: representation of overlaps, paths to create superstrings, voracious algorithm, acyclic graphs.</li> <li>• Assembly heuristics: search for overlays, ordering fragments, alignments and consensus.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the computational challenge of the Sequence Assembly problem. [Familiarizarse]</li> <li>• Understand the principle of formal model for assembly. [Evaluar]</li> <li>• Know the main heuristics for the problem of assembly of DNA sequences[Usar]</li> </ul>
<b>Readings :</b> [SM97], [Alu06]	

<b>Unit 5: Secondary and tertiary structures (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Molecular structures: primary, secondary, tertiary, quaternary.</li> <li>• Prediction of secondary structures of RNA: formal model, pair energy, structures with independent bases, solution with Dynamic Programming, structures with loops.</li> <li>• <i>Protein folding</i>: Estructuras en proteínas, problema de protein folding.</li> <li>• <i>Protein Threading</i>: Definitions, Branch Bound Algorithm, Branch Bound for protein threading.</li> <li>• <i>Structural Alignment</i>: Definitions, DALI algorithm</li> </ul>	<ul style="list-style-type: none"> <li>• Know the protein structures and the necessity of computational methods for the prediction of the geometry. [Familiarizarse]</li> <li>• Know the algorithms for solving prediction problems of secondary structures RNA, and structures in proteins. [Evaluar]</li> </ul>
<b>Readings :</b> [SM97], [CB00], [Alu06]	

<b>Unit 6: Probabilistic Models in Molecular Biology (4 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Probability: Random Variables, Markov Chains, Metropoli-Hasting Algorithm, Markov Random Fields, and Gibbs Sampler, Maximum Likelihood.</li> <li>• Hidden Markov Models (HMM), parameter estimation, Viterbi algorithm and Baul-Welch method, Application in paired and multiple alignments, Motifs detection in proteins, in eukaryotic DNA, in sequences families.</li> <li>• Probabilistic phylogeny: probabilistic models of evolution, likelihood of alignments, likelihood for inference, comparison of probailistic and non-probabilistic methods</li> </ul>	<ul style="list-style-type: none"> <li>• Review concepts of Probabilistic Models and understand their importance in Computational Molecular Biology. [Evaluar]</li> <li>• Know and apply Hidden Markov Models for various analyzes in Molecular Biology.. [Usar]</li> <li>• Know the application of probabilistic models in Phylogeny and to compare them with non-probabilistic models[Evaluar]</li> </ul>
<b>Readings :</b> [Dur+98], [CB00], [Alu06], [Kro+94]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Kro+94] Anders Krogh et al. "Hidden Markov Models in Computational Biology, Applications to Protein Modeling". In: *J Mol. Biol* 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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS369. Topics in Artificial Intelligence (Elective)

**2. GENERAL INFORMATION**

- 2.1 Course : CS369. Topics in Artificial Intelligence
- 2.2 Semester : 9<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Elective
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS261. Artificial Intelligence. (6<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Provides a set of tools to solve problems that are difficult to address using traditional algorithmic methods. Includes heuristics, planning, knowledge representation and reasoning formalisms, machine learning techniques, methods for action-reaction problems (e.g., reinforcement learning), as well as applications in natural language processing, computer vision, and robotics.

**5. GOALS**

- Complete an advanced Artificial Intelligence course recommended by the ACM/IEEE Computing Curricula.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**AG-C08) Problem Analysis:** Identifies, formulates, and analyzes complex computing problems. (Usage)

**AG-C09) Solution Design and Development:** Designs, implements, and evaluates solutions for complex computing problems. (Usage)

**AG-C11) Tool Usage:** Applies modern computing tools in problem-solving. (Assessment)

**7. TOPICS**

<b>Unit 1: (60 hours)</b>	
<b>Competences Expected: 1,2,6,AG-C08,AG-C09,AG-C11</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"><li>• Advanced topics in AI</li></ul>	<ul style="list-style-type: none"><li>• Develop expertise in diverse Artificial Intelligence techniques [Usar]</li></ul>
<b>Readings :</b> [RN03], [Hay99], [Gol89]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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 Artifical: Un enfoque moderno*. Prentice Hall, 2003.





**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS351. Topics in Computer Graphics (Elective)

**2. GENERAL INFORMATION**

- 2.1 Course : CS351. Topics in Computer Graphics
- 2.2 Semester : 9<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Elective
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS251. Computer graphics . (7<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

Advanced course covering real-time rendering techniques, 3D modeling, and physical simulation, with applications in entertainment industries and scientific visualization. Aligns with ACM/IEEE Computing Curricula guidelines.

**5. GOALS**

- Implement advanced graphics algorithms (ray tracing, GPU computing).

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- AG-C01**) The Professional and the World: Analyzes and evaluates the impact of solutions to complex computing problems on the sustainable development of society. (Usage)
- AG-C05**) Project Management: Applies project management principles in computing to manage projects. (Usage)

**7. TOPICS**

<b>Unit 1: (60 hours)</b>	
<b>Competences Expected: 1,3</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"><li>• Hardware-accelerated ray tracing</li><li>• Procedural texturing</li></ul>	<ul style="list-style-type: none"><li>• Integrate offline/real-time rendering techniques [Usar]</li><li>• Evaluate computational costs [Evaluar]</li></ul>
<b>Readings :</b> [AH08], [SM20]	

**8. WORKPLAN**

**8.1 Methodology**

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### **8.2 Theory Sessions**

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### **8.3 Practical Sessions**

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## **9. EVALUATION SYSTEM**

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

## **10. BASIC BIBLIOGRAPHY**

[AH08] Tomas Akenine-Möller and Eric Haines. *Real-Time Rendering*. AK Peters, 2008.

[SM20] Peter Shirley and Steve Marschner. *Fundamentals of Computer Graphics*. CRC Press, 2020.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

### 1. COURSE

CS392. Tópicos en Ingeniería de Software (Elective)

### 2. GENERAL INFORMATION

2.1 Course	:	CS392. Tópicos en Ingeniería de Software
2.2 Semester	:	9 <sup>th</sup> Semester.
2.3 Credits	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Elective
2.7 Learning modality	:	Face to face
2.8 Prerequisites	:	CS391. Software Engineering III. (7 <sup>th</sup> Sem)

### 3. PROFESSORS

Meetings after coordination with the professor

### 4. INTRODUCTION TO THE COURSE

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 elementos 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. GOALS

- 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. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 3) Communicate effectively in a variety of professional contexts.. (Usage)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Usage)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

## 7. TOPICS

**Unit 1: Diseño de Software (18 hours)****Competences Expected:****Topics****Learning Outcomes**

- 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.
- 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.
- Modelos estructurales y de comportamiento de los diseños de software.
- Diseño de patrones.
- Relaciones entre los requisitos y diseños: La transformación de modelos, el diseño de los contratos, invariantes.
- Conceptos de arquitectura de software y arquitecturas estándar (por ejemplo, cliente-servidor, n-capas, transforman centrados, tubos y filtros).
- 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 standard widget set)
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Calidad del diseño interno, y modelos para: eficiencia y desempeño, redundancia y tolerancia a fallos, trazabilidad de los requerimientos.
- Medición y análisis de la calidad de un diseño.
- Compensaciones entre diferentes aspectos de la calidad.
- Aplicaciones en frameworks.
- 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.
- Principales diseños de seguridad y codificación (cross-reference IAS/Principles of secure design).
  - Principio de privilegios mínimos
  - Principio de falla segura por defecto
  - Principio de aceptabilidad psicológica

- Formular los principios de diseño, incluyendo la separación de problemas, ocultación de información, acoplamiento y cohesión, y la encapsulación [Usar]
- 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]
- Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo [Usar]
- 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]
- Para un sistema simple adecuado para una situación dada, discutir y seleccionar un paradigma de diseño apropiado [Usar]
- Crear modelos apropiados para la estructura y el comportamiento de los productos de software desde la especificaciones de requisitos [Usar]
- Explicar las relaciones entre los requisitos para un producto de software y su diseño, utilizando los modelos apropiados [Usar]
- 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]
- Dado un diseño de alto nivel, identificar la arquitectura de software mediante la diferenciación entre las arquitecturas comunes de software, tales como 3 capas (*3-tier*), *pipe-and-filter*, y cliente-servidor [Usar]
- Investigar el impacto de la selección arquitecturas de software en el diseño de un sistema simple [Usar]
- Aplicar ejemplos simples de patrones en un diseño de software [Usar]
- Describir una manera de refactorar y discutir cuando esto debe ser aplicado [Usar]
- Seleccionar componentes adecuados para el uso en un diseño de un producto de software [Usar]
- Explicar cómo los componentes deben ser adaptados para ser usados en el diseño de un producto de software [Usar]
- Diseñar un contrato para un típico componente de software pequeño para el uso de un dado sistema [Usar]
- Discutir y seleccionar la arquitectura de software adecuada para un sistema de software simple para un dado escenario [Usar]

**Unit 2: Gestión de Proyectos de Software (14 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 necesita 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>

Unit 3: (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Som17], [PM15]	

Unit 4: (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
Readings : [Som17], [PM15]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS353. Quantum Computing (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS353. Quantum Computing
- 2.2 Semester : 10<sup>th</sup> Semester.
- 2.3 Credits : 4
- 2.4 Horas : 2 HT; 3 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites :
  - CS221. Computer Systems Architecture. (4<sup>th</sup> Sem)
  - MA307. Mathematics applied to computing. (6<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course introduces fundamental principles of quantum computing including qubits, superposition, entanglement and quantum algorithms. Students will learn to contrast classical and quantum models while exploring applications in cryptography, optimization and physical system simulation using frameworks like Qiskit or Cirq.

**5. GOALS**

- Understand quantum mechanics postulates applied to computing
- Implement basic quantum circuits using modern tools
- Analyze ethical and technical impacts of quantum computing

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
  - 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
  - 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Familiarity)
- AG-C02) Ethics:** Applies ethical principles and commits to professional ethics and standards of computing practice. (Familiarity)
- AG-C08) Problem Analysis:** Identifies, formulates, and analyzes complex computing problems. (Usage)
- AG-C09) Solution Design and Development:** Designs, implements, and evaluates solutions for complex computing problems. (Usage)

**7. TOPICS**



Unit 1: Quantum Mechanics Fundamentals (16 hours)	
Competences Expected: 1,AG-C08	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Qubits and Bloch sphere representation</li> <li>• Quantum mechanics postulates</li> <li>• Quantum gates (Hadamard, CNOT)</li> </ul>	<ul style="list-style-type: none"> <li>• Model a qubit mathematically [Usar]</li> </ul>
Readings : [NC10], [Tea23]	

Unit 2: Quantum Algorithms (16 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Deutsch-Jozsa algorithm</li> <li>• Grover's search</li> <li>• Shor's factorization</li> </ul>	<ul style="list-style-type: none"> <li>• Implement Grover's algorithm on a simulator [Usar]</li> </ul>
Readings : [Pre18], [Tea23]	

Unit 3: Ethics and Applications (16 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Post-quantum cryptography</li> <li>• Quantum supremacy</li> <li>• Ethical responsibility in quantum development</li> </ul>	<ul style="list-style-type: none"> <li>• Analyze ethical risks of RSA breaking [Familiarize]</li> </ul>
Readings : [Tea19]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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>.
- [Tea23] IBM Quantum Team. *Qiskit Textbook*. Guía práctica con ejemplos en Qiskit. 2023. URL: <https://qiskit.org/textbook>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

## 1. COURSE

CS365. Evolutionary Computing (Mandatory)

## 2. GENERAL INFORMATION

- |                            |   |  |
|----------------------------|---|--|
| 2.1 Course                 | : | CS365. Evolutionary Computing                  |
| 2.2 Semester               | : | 10 <sup>th</sup> Semester.                     |
| 2.3 Credits                | : | 4  |
| 2.4 Horas                  | : | 2 HT; 4 HP;                                    |
| 2.5 Duration of the period | : | 16 weeks                                       |
| 2.6 Type of course         | : | Mandatory                                      |
| 2.7 Learning modality      | : | Face to face                                   |
| 2.8 Prerequisites          | : | CS262. Machine learning. (7 <sup>th</sup> Sem) |

## 3. PROFESSORS

Meetings after coordination with the professor

## 4. INTRODUCTION TO THE COURSE

This course introduces biologically-inspired algorithms for solving complex optimization and design problems. It covers genetic algorithms, evolutionary strategies, and genetic programming, with applications in engineering, logistics, and machine learning. Students will implement solutions using modern frameworks like DEAP.

## 5. GOALS

- Model complex problems using evolutionary paradigms.
- Implement bio-inspired algorithms in Python.
- Analyze performance of different evolutionary techniques.

## 6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

**AG-C08) Problem Analysis:** Identifies, formulates, and analyzes complex computing problems. (Usage)

**AG-C09) Solution Design and Development:** Designs, implements, and evaluates solutions for complex computing problems. (Usage)

**AG-C11) Tool Usage:** Applies modern computing tools in problem-solving. (Assessment)

## 7. TOPICS

<b>Unit 1: Genetic Algorithms (16 hours)</b>	
<b>Competences Expected: 1,AG-C08</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Chromosomal representation (binary, real-valued)</li> <li>• Selection operators (roulette, tournament)</li> <li>• Crossover and mutation</li> </ul>	<ul style="list-style-type: none"> <li>• Design representations for discrete problems [Usar]</li> <li>• Compare selection operators [Evaluar]</li> </ul>
<b>Readings :</b> [Gol18], [Tea23]	

<b>Unit 2: Evolutionary Strategies (16 hours)</b>	
<b>Competences Expected: 2,AG-C09</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• <math>(\mu + \lambda)</math> and <math>(\mu, \lambda)</math> algorithms</li> <li>• Parameter self-adaptation</li> <li>• Continuous optimization applications</li> </ul>	<ul style="list-style-type: none"> <li>• Implement strategies for real-world problems [Usar]</li> <li>• Tune self-adaptation parameters [Evaluar]</li> </ul>
<b>Readings :</b> [Bey01], [PythonEC]	

<b>Unit 3: Advanced Applications (16 hours)</b>	
<b>Competences Expected: 6,AG-C11</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Genetic programming</li> <li>• Co-evolution</li> <li>• Multi-objective optimization (NSGA-II)</li> </ul>	<ul style="list-style-type: none"> <li>• Develop multi-objective solutions [Usar]</li> <li>• Evaluate Pareto front trade-offs [Evaluar]</li> </ul>
<b>Readings :</b> [ES15], [Aut22]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS3P2. Cloud Computing (Mandatory)

**2. GENERAL INFORMATION**

- 2.1 Course : CS3P2. Cloud Computing
- 2.2 Semester : 10<sup>th</sup> Semester.
- 2.3 Credits : 3
- 2.4 Horas : 1 HT; 4 HP;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : CS370. Big Data. (9<sup>th</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

To understand advanced computational techniques, students must have a strong knowledge of various discrete structures, structures that will be implemented and used in the laboratory with the programming language.

**5. GOALS**

- Students will be able to model computer science problems using graphs and trees related to data structures.
- Students will efficiently apply traversal strategies to search for data optimally.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

Unit 1: Theoretical Foundations of Cloud Computing (12 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"><li>• Introduction to Cloud Computing</li><li>• Cloud Computing Service Models</li><li>• Cloud Computing Deployment Models</li><li>• Infrastructure and Data Centers</li><li>• Research Trends in Cloud Computing</li></ul>	<ul style="list-style-type: none"><li>• Understand the concepts related to Cloud Computing.</li><li>• Understand the infrastructure and components of a Data Center.</li><li>• Understand service models and deployment types in Cloud Computing.</li><li>• Be familiar with research trends in the area of Cloud Computing.</li></ul>
Readings : [aboveTheCloud], [surveySecurity], [mobileCloud]	

Unit 2: Data Processing (15 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Introduction to the Hadoop framework.</li> <li>• Hadoop Distributed File System.</li> <li>• Introduction to the MapReduce programming model.</li> <li>• Introduction to the Spark framework.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concepts related to the Hadoop framework.</li> <li>• Understand the concepts related to the Hadoop Distributed File System.</li> <li>• Understand and apply the MapReduce programming model.</li> <li>• Understand the concepts related to the Spark framework.</li> </ul>
Readings : [mapreduce], [spark], [yarn]	

Unit 3: Virtualization, Containerization (15 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Introduction to Containerization.</li> <li>• Evolution of Containerization.</li> <li>• Differences between Containerization and Virtualization.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand the concept of Containerization.</li> <li>• Create and use containers.</li> <li>• Understand the differences between Containerization and Virtualization.</li> </ul>
Readings : [CborgOmegaKubernetes], [borg], [ContainerizationPaaSCloud], [VirtualizationContainerization]	

Unit 4: Trends in Cloud Computing (12 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Autoscaling.</li> <li>• Infrastructure as Code.</li> <li>• Serverless Computing.</li> </ul>	<ul style="list-style-type: none"> <li>• Understand different forms of autoscaling.</li> <li>• Use different tools for Infrastructure as Code in the cloud.</li> <li>• Understand the Serverless Computing paradigm.</li> </ul>
Readings : [Cormen2009], [Preparata], [Berg]	

Unit 5: Distributed Systems (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Distributed System Faults</li> <li>• Distributed Algorithms</li> <li>• Distributed System Architectures</li> <li>• Distributed Services</li> <li>• Core Distributed System Concepts</li> </ul>	<ul style="list-style-type: none"> <li>• Distinguish between different types of distributed system faults [Familiarizarse]</li> <li>• Explain the challenges of distributed systems [Familiarizarse]</li> <li>• Write distributed algorithms [Usar]</li> <li>• Measure the performance of distributed systems [Usar]</li> <li>• Explain the rationale behind different distributed system designs [Familiarizarse]</li> <li>• Implement a distributed system [Usar]</li> <li>• Explain the trade-offs in distributed system design [Familiarizarse]</li> <li>• Describe different distributed system architectures [Familiarizarse]</li> <li>• Give examples of distributed systems [Usar]</li> </ul>
Readings : [Cou+11]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

[Cou+11] George Coulouris et al. *Distributed Systems: Concepts and Design*. 5th. USA: Addison-Wesley Publishing Company, 2011.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS3P3. Internet of Things (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |   |
|-----------------------------------|---|
| <b>2.1 Course</b>                 | : CS3P3. Internet of Things   |
| <b>2.2 Semester</b>               | : 10 <sup>th</sup> Semester.  |
| <b>2.3 Credits</b>                | : 3   |
| <b>2.4 Horas</b>                  | : 1 HT; 4 HP;   |
| <b>2.5 Duration of the period</b> | : 16 weeks  |
| <b>2.6 Type of course</b>         | : Mandatory   |
| <b>2.7 Learning modality</b>      | : Face to face  |
| <b>2.8 Prerequisites</b>          | : CS3P1. Parallel and Distributed Computing . (8 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

The last decade has an explosive growth in multiprocessor computing, including multi-core processors and distributed data centers. As a result, parallel and distributed computing has evolved from a broadly elective subject to be one of the major components in mesh studies in undergraduate computer science. Both parallel computing and distribution involve the simultaneous execution of multiple processes on different devices that change position.

**5. GOALS**

- That the student is able to create parallel applications of medium complexity by efficiently taking advantage of different mobile devices.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

**7. TOPICS**

**Unit 1: Fundamentos de paralelismo (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<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 más 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 más alto nivel [Familiarizarse]</li></ul>
<b>Readings :</b> [Pac11], [Mat14], [Qui03]	



<b>Unit 2: Arquitecturas paralelas (12 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Procesadores mutlinúcleo.</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>
<b>Readings :</b> [Pac11], [KH13], [SK10]	

<b>Unit 3: Descomposición en paralelo (18 hours)</b>	
<b>Competences Expected:</b>	
<b>Topics</b>	<b>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>• Descomposición 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>
<b>Readings :</b> [Pac11], [Mat14], [Qui03]	

**Unit 4: Comunicación y coordinación (18 hours)****Competences Expected:**

Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Memoria Compartida.</li> <li>• La consistencia, y su papel en los lenguaje de programación garantías 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) barreras, 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 mutua 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 semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]</li> </ul>
<b>Readings :</b> [Pac11], [Mat14], [Qui03]	

**Unit 5: Análisis y programación de algoritmos paralelos (18 hours)**

**Competences Expected:**

Topics	Learning Outcomes
<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 Algoritmicos 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 analogo secuencial [Usar]</li> <li>• Descomponer un problema (por ejemplo, contar el número de ocurrencias de una palabra en un documento) via 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>
<p><b>Readings :</b> [Mat14], [Qui03]</p>	

Unit 6: Desempeño en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
<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>
<b>Readings :</b> [Pac11], [Mat14], [KH13], [SK10]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [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>.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

CS403. Capstone Project III (Mandatory)

**2. GENERAL INFORMATION**

- |                                   |   |
|-----------------------------------|---|
| <b>2.1 Course</b>                 | : CS403. Capstone Project III                       |
| <b>2.2 Semester</b>               | : 10 <sup>th</sup> Semester.                        |
| <b>2.3 Credits</b>                | : 4   |
| <b>2.4 Horas</b>                  | : 1 HT; 5 HP;                                       |
| <b>2.5 Duration of the period</b> | : 16 weeks  |
| <b>2.6 Type of course</b>         | : Mandatory   |
| <b>2.7 Learning modality</b>      | : Face to face                                      |
| <b>2.8 Prerequisites</b>          | : CS402. Capstone Project II. (9 <sup>th</sup> Sem) |

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

This course aims at the student to conclude his thesis project.

**5. GOALS**

- That the student is in the capacity to formally present his thesis project with the theoretical framework and complete bibliographic survey.
- That the student master the state of the art of his area of research.
- The deliverables of this course are:

**Avance parcial:** Thesis plan progress including motivation and context, problem definition, objectives, schedule of activities up to the final thesis project and the state of the art of the topic addressed.

**Final:** Complete thesis plan and advancement of Thesis including theoretical framework chapters, related works and preliminary (formal or statistical) results oriented to your thesis topic.

**6. COMPETENCES**

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Assessment)
- 2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (Assessment)
- 3) Communicate effectively in a variety of professional contexts.. (Assessment)
- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)
- 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (Assessment)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Assessment)

## 7. TOPICS

Unit 1: Thesis project (30 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Thesis project.</li> </ul>	<ul style="list-style-type: none"> <li>• Description of the format used by the University for the thesis[Evaluat</li> <li>• Conclude the thesis project plan[Evaluat</li> <li>• Present the state of the art thesis topic(50%)[Evaluat</li> </ul>
Readings : [IEE08], [Ass08], [Cit08]	

Unit 2: Thesis progress (30 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Thesis Progress.</li> </ul>	<ul style="list-style-type: none"> <li>• Description of the format used by the University for the thesis[Evaluat</li> <li>• Conclude the chapter of the theoretical framework of the Thesis[Evaluat</li> <li>• Complete the chapter on related works(35%)[Evaluat</li> <li>• Plan, develop and present results (formal or statistical) of experiments oriented to your thesis topic (35%)[Evaluat</li> </ul>
Readings : [IEE08], [Ass08], [Cit08]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Ass08] Association for Computing Machinery. *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.



**National University of Engineering (UNI)**  
School of Computer Science  
Syllabus 2024-II

**1. COURSE**

FG211. Professional Ethics (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Course</b>	:	FG211. Professional Ethics
<b>2.2 Semester</b>	:	7 <sup>th</sup> Semester.
<b>2.3 Credits</b>	:	3
<b>2.4 Horas</b>	:	2 HT; 2 HP;
<b>2.5 Duration of the period</b>	:	16 weeks
<b>2.6 Type of course</b>	:	Mandatory
<b>2.7 Learning modality</b>	:	Face to face
<b>2.8 Prerequisites</b>	:	None

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

La ética es una parte constitutiva inherente al ser humano, y como tal debe plasmarse en el actuar cotidiano y profesional de la persona humana. Es indispensable que la persona asuma su rol activo en la sociedad pues los sistemas económico-industrial, político y social no siempre están en función de valores y principios, siendo éstos en realidad los pilares sobre los que debería basarse todo el actuar de los profesionales.

**5. GOALS**

- Que el alumno amplíe sus propios criterios personales de discernimiento moral en el quehacer profesional, de forma que no sólo tome en cuenta los criterios técnicos pertinentes sino que incorpore a sí mismo cuestionamientos de orden moral y se adhiera a una ética profesional correcta, de forma que sea capaz de aportar positivamente en el desarrollo económico y social de la ciudad, región, país y comunidad global.[Usar]

**6. COMPETENCES**

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)
- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)
- 7) Develop computational technology for the well-being of all, contributing with human formation, scientific, technological and professional skills to solve social problems of our community. (Usage)

**7. TOPICS**

Unit 1: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Ser profesional y ser moral.</li> <li>• La objetividad moral y la formulación de principios morales.</li> <li>• El profesional y sus valores.</li> <li>• La conciencia moral de la persona.</li> <li>• El aporte de la DSI en el quehacer profesional.</li> <li>• El bien común y el principio de subsidiaridad.</li> <li>• Principios morales y propiedad privada.</li> <li>• Justicia: Algunos conceptos básicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Presentar al alumno la importancia de tener principios y valores en la sociedad actual.[Usar]</li> <li>• Presentar algunos de los principios de podrían contribuir en la sociedad de ser aplicados y vividos día a día. [Usar]</li> <li>• Presentar a los alumnos el aporte de la Doctrina Social de la Iglesia en el quehacer profesional. [Usar]</li> </ul>
<b>Readings :</b> [Com92], [Sch95], [Loz00], [Arg06]	

Unit 2: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• La responsabilidad individual del trabajador en la empresa.</li> <li>• Liderazgo y ética profesional en el entorno laboral.</li> <li>• Principios generales sobre la colaboración en hechos inmorales.</li> <li>• El profesional frente al soborno: ¿víctima o colaboración?</li> </ul>	<ul style="list-style-type: none"> <li>• Presentar al alumno el rol de la responsabilidad social individual y del liderazgo en la empresa. [Familiarizarse]</li> <li>• Conocer el juicio de la ética frente a la corrupción y sobornos como forma de relación laboral. [Familiarizarse]</li> <li>• Presentar la profesión como una forma de realización personal, y como consecuencia. []</li> </ul>
<b>Readings :</b> [Com92], [Man07], [Sch95], [Pér98], [Nie03]	

Unit 3: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• La ética profesional frente a la ética general.</li> <li>• Trabajo y profesión en los tiempos actuales.</li> <li>• Ética, ciencia y tecnología.</li> <li>• Valores éticos en organizaciones relacionadas con el uso de la información.</li> <li>• Valores éticos en la era de la Sociedad de la Información.</li> </ul>	<ul style="list-style-type: none"> <li>• Presentar al alumno las interrelaciones entre ética y las disciplinas de la última era tecnológica.[Familiarizarse]</li> </ul>
<b>Readings :</b> [Com92], [IEE04], [Her06]	



Unit 4: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> <li>• Ética informática. <ul style="list-style-type: none"> <li>– Ética y software.</li> <li>– El software libre.</li> </ul> </li> <li>• Regulación y ética de telecomunicaciones. <ul style="list-style-type: none"> <li>– Ética en Internet.</li> </ul> </li> <li>• Derechos de autor y patentes.</li> <li>• Ética en los servicios de consultoría.</li> <li>• Ética en los procesos de innovación tecnológica.</li> <li>• Ética en la gestión tecnológica y en empresas de base tecnológica.</li> </ul>	<ul style="list-style-type: none"> <li>• Presentar al alumno algunos aspectos que confrontan la ética con el quehacer de las disciplinas emergentes en la sociedad de la información. [Familiarizarse]</li> </ul>
<b>Readings :</b> [Com02], [Her06], [Com92]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

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

## 10. BASIC BIBLIOGRAPHY

- [Com92] Association for Computing Machinery (ACM). "ACM Code of Ethics and Professional Conduct". In: (1992). URL: <http://www.acm.org/about/code-of-ethics>.
- [Sch95] E. Schmidt. *Ética y Negocios para América Latina*. Universidad del Pacífico, 1995.
- [Pér98] J. A. Pérez López. *Liderazgo y Ética en la Dirección de Empresas*. Bilbao, 1998.
- [Loz00] C Loza. "El aporte de la Doctrina Social de la Iglesia a la Toma de Decisiones Empresariales". In: *Separata ofrecida por el profesor* (2000).
- [Com02] Pontificio Consejo para las Comunicaciones Sociales. *Ética en Internet*. 2002.
- [Nie03] R. Nieburh. *El Yo Responsable. Ensayo de Filosofía Moral Cristiana*. Bilbao, 2003.
- [IEE04] IEEE. "IEEE Code of Ethics". In: *IEE* (2004). URL: <http://www.ieee.org/about/corporate/governance/p7-8.html>.
- [Arg06] Argandoña. "La identidad Cristiana del Directivo de Empresa". In: *IESE* (2006).
- [Her06] A. Hernández. *Ética Actual y Profesional. Lecturas para la Convivencia Global en el Siglo XXI*. Ed. Thomson, 2006.
- [Man07] G. Manzone. *La Responsabilidad de la Empresa, Business Ethics y Doctrina Social de la Iglesia en Diálogo*. Universidad Católica San Pablo, 2007.