

**National University of Engineering (UNI)
Faculty of Sciences
Department of Computer Science
School of Computer Science**



Book of Syllabi

– 2026-I –

Lima: December 2, 2025

Equipo de Trabajo

Dr. Ciro Nuñez Iturri

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

Dr. Javier Solano Salinas

Director del Departamento de Ciencia de la Computación
e-mail: jsolano@uni.edu.pe

MSc Yuri Nuñez Medrano

Profesor Investigador
Departamento de Ciencia de la Computación
email: ynunezm@uni.edu.pe

Dr. José Luis Segovia-Juárez

PhD en Ciencia de la Computación. Wayne State University (Detroit, USA)
(2001)

Profesor del Departamento de Ciencia de la Computacion. UNI
email: jsegoviaj@uni.edu.pe
<http://www.tecnociencia9.com/jlsegovia/es>

Dr. Marcos Antonio Alania Vicente

PhD en Ciencias Universidad de Amberes. Bélgica (2017)
Profesor del Departamento de Ciencia de la Computacion. UNI
Desarrollador de algoritmos de IA para procesos complejos interdisciplinarios
email: alania.vicente@gmail.com

MSc. Eduardo Yauri Lozano

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

MSc. Ronald Ricardo Martinez Chunga

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

Prof. Miguel Arrunategui Angulo

Docente del Departamento de Ciencia de la Computación
email: garrunateguia@uni.edu.pe

Dr. 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

Contents

1	First Semester	7
1.1	CS111. Introduction to Programming	8
1.2	BMA101. Linear Algebra	16
1.3	BMA102. Differential Calculus	19
1.4	BCH101. Chemistry I	22
1.5	FG001. General Elective	25
1.6	BEI101. Technical and professional English I	27
2	Second Semester	31
2.1	CS100. Introduction to Computer Science	32
2.2	CS112. Objects oriented programming I	46
2.3	CS1D1. Discrete Structures	53
2.4	BMA103. Integral Calculus	59
2.5	BFI101. Physics I	62
2.6	BEI102. Technical and professional English II	65
3	Third Semester	69
3.1	CS113. Objects oriented programming II	70
3.2	CS2B1. Platform Based Development	74
3.3	AI161. Applied AI	79
3.4	BMA104. Advanced Differential and Integral Calculus	83
3.5	ST251. Probability Calculation	87
3.6	BEI201. Technical and professional English III	91
4	Fourth Semester	95
4.1	CS210. Algorithms and Data Structures	96
4.2	CS211. Theory of Computation	99
4.3	CS221. Computer Systems Architecture	102
4.4	CS271. Data Management	108
4.5	CS401. Research Methodology	113
4.6	MA202. Numerical Methods	115
4.7	BEI202. Technical and professional English IV	118
5	Fifth Semester	123
5.1	CS212. Analysis and Design of Algorithms	124
5.2	CS261. Artificial Intelligence	128
5.3	CS272. Databases II	137
5.4	CS291. Software Engineering I	141

5.5	CS2S1. Operating systems	152
5.6	BEI203. Technical and professional English V	160
6	Sixth Semester	165
6.1	CS231. Networking and Communication	166
6.2	CS311. Competitive Programming	170
6.3	CS312. Advanced Data Structures	174
6.4	CS342. Compilers	178
6.5	AI263. Introduction to Machine Learning	183
6.6	FI201. Computational Physics	186
7	Seventh Semester	189
7.1	CS251. Computer graphics	190
7.2	CS292. Software Engineering II	197
7.3	CS2H1. User Experience (UX)	203
7.4	AI264. Deep Learning	209
7.5	BBI101. Biology	211
7.6	FG211. Professional Ethics	215
8	Eighth Semester	219
8.1	CS281. Computing in Society	220
8.2	CS3I1. Computer Security	229
8.3	CS3P1. Parallel and Distributed Computing	239
8.4	CS402. Capstone Project I	245
8.5	FG106. Theater	247
8.6	EX301. Extracurricular Activities	251
8.7	AI268. Computational Vision	253
8.8	CS391. Software Engineering III	255
8.9	CS393. Information systems	262
9	Ninth Semester	265
9.1	CS370. Big Data	266
9.2	CS400. Pre-professional internships	269
9.3	CS403. Capstone Project II	271
9.4	AI365. Advanced Generative AI Models	274
9.5	CB309. Bioinformatics	276
9.6	CS351. Topics in Computer Graphics	280
9.7	CS353. Quantum Computing	283
9.8	AI369. Robotics	286
9.9	CS392. Tópicos en Ingeniería de Software	288
9.10	CS3P3. Internet of Things	293
10	Tenth Semester	299
10.1	CS3P2. Cloud Computing	300
10.2	CS404. Research Workshop	304
10.3	AI367. Topics in Artificial Intelligence	306
10.4	AI368. Evolutionary Computing	308
10.5	FG350. Leadership and Performance	310

Chapter 1

First Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS111-CS2023. Introduction to Programming (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS111-CS2023. Introduction to Programming
2.2 Semester	: 1 st 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

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

- 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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Historia (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Pre-historia – El mundo antes de 1946. • Historia del hardware, software, redes. • Pioneros de la Computación. • Historia de Internet.	• Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Evaluar]
Readings : [BB19], [Gut13], [Zel10]	

Unit 2: Sistemas de tipos básicos (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Tipos como conjunto de valores junto con un conjunto de operaciones. – Tipos primitivos (p.e. números, booleanos) – Composición de tipos contruídos de otros tipos (p.e., registros, uniones, arreglos, listas, funciones, referencias) • Asociación de tipos de variables, argumentos, resultados y campos. • Tipo de seguridad y los errores causados por el uso de valores de manera incompatible dadas sus tipos previstos.	• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse] • 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] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar]
Readings : [Gut13], [Zel10]	

Unit 3: Conceptos Fundamentales de Programación (9 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Sintaxis y semántica básica de un lenguaje de alto nivel. • Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos) • Expresiones y asignaciones. • Operaciones básicas I/O incluyendo archivos I/O. • Estructuras de control condicional e iterativas. • Paso de funciones y parámetros. • Concepto de recursividad.	• 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] • Identifica y describe el uso de tipos de datos primitivos [Familiarizarse] • Escribe programas que usan tipos de datos primitivos [Usar] • Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar] • 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] • Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar] • Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Familiarizarse] • Describe el concepto de recursividad y da ejemplos de su uso [Evaluar] • Identifica el caso base y el caso general de un problema basado en recursividad [Familiarizarse]
Readings : [Gut13], [Zel10]	

Unit 4: Análisis Básico (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Uso de la notación Big O. • Análisis de algoritmos iterativos y recursivos.	• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Familiarizarse] • Indique la definición formal de Big O [Familiarizarse] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar]
Readings : [Gut13], [Zel10]	

Unit 5: Algoritmos y Estructuras de Datos fundamentales (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. - Algoritmos de búsqueda secuencial y binaria. - Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) - Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) - Tablas Hash, incluyendo estrategias para evitar y resolver colisiones. - Árboles de búsqueda binaria: - Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles. - Grafos y algoritmos en grafos: - Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) - Recorrido en profundidad y amplitud - Montículos (Heaps) - Grafos y algoritmos en grafos: - Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) - Árbol de expansión mínima (algoritmos de Prim y Kruskal) - Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)	- Implementar algoritmos numéricos básicos [Usar] - Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] - Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] - Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse] - Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse] - 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] - Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse] - Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar] - 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] - Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Familiarizarse] - 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] - Trazar y/o implementar un algoritmo de comparación de string [Usar]
Readings : [Gut13], [Zel10]	

Unit 6: Programación orientada a objetos (4 hours)	
Competences Expected: 1	
Topics	Learning Outcomes
• Object poriented languages and encapsulation: Privacy y visibility of class members. • Definición de las categorías, campos, métodos y constructores. • Subclasses and inheritance. • Asignación dinámica: definición de método de llamada.	• Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Familiarizarse] • 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 [Familiarizarse] • 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] • Use encapsulation to create objects [Familiarizarse].
Readings : [Gut13], [Zel10]	

Unit 7: Algoritmos y Diseño (9 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Conceptos y propiedades de los algoritmos – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación	• Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse] • Crea algoritmos para resolver problemas simples [Usar] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar]
Readings : [Gut13], [Zel10]	

Unit 8: Métodos de Desarrollo (1 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Entornos modernos de programación: – Búsqueda de código. – Programación usando librería de componentes y sus APIs.	• Construir y depurar programas que utilizan las bibliotecas estándar disponibles con un lenguaje de programación elegido [Familiarizarse]
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 2026-I

1. COURSE

MA101FCCS. Linear Algebra (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA101FCCS. Linear Algebra
2.2 Semester	: 1 st Semester
2.3 Credits	: 4
2.4 Horas	: 3 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

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)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Vector Spaces (8 hours)	
Competences Expected: 1,6,AG-C07,AG-C08	
Topics	Learning Outcomes
• Definition of vector space and subspace. • Linear combinations, linear independence, and bases. • Dimension and rank.	• Define and give examples of vector spaces and subspaces. [Familiarizarse] • Determine the linear independence of a set of vectors. [Usar] • Calculate bases and the dimension of a vector space. [Evaluar]
Readings : [Str16], [LLM16]	

Unit 2: Matrices and Systems of Linear Equations (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Matrix operations. • Gaussian elimination and reduced row echelon form. • Solving systems of linear equations. • Inverse matrices and determinants.	• Perform matrix operations. [Familiarizarse] • Solve systems of linear equations using Gaussian elimination. [Usar] • Calculate the inverse of a matrix and its determinant. [Evaluar]
Readings : [Str16], [LLM16]	

Unit 3: Linear Transformations (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Definition and examples of linear transformations. • Kernel and image of a linear transformation. • Transformation matrices.	• Define and give examples of linear transformations. [Familiarizarse] • Calculate the kernel and image of a linear transformation. [Usar] • Represent linear transformations using matrices. [Evaluar]
Readings : [Str16], [LLM16]	

Unit 4: Eigenvalues and Eigenvectors (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Definition and calculation of eigenvalues and eigenvectors. • Diagonalization of matrices. • Applications of eigenvalues and eigenvectors.	• Define and calculate eigenvalues and eigenvectors. [Familiarizarse] • Diagonalize matrices. [Usar] • Apply eigenvalues and eigenvectors to solve problems. [Evaluar]
Readings : [Str16], [LLM16]	

Unit 5: Orthogonality and Least Squares (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Inner product and orthogonality. • Orthogonal projections. • Least squares method.	• Define and calculate inner product and orthogonality. [Familiarizarse] • Calculate orthogonal projections. [Usar] • Apply the least squares method. [Evaluar]
Readings : [Str16], [LLM16]	

Unit 6: Applications in Computing (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Applications in computer graphics. • Applications in machine learning. • Applications in algorithm analysis.	• Describe applications of linear algebra in computer graphics. [Familiarizarse] • Explain how linear algebra is used in machine learning. [Usar] • Analyze the complexity of algorithms using linear algebra concepts. [Evaluar]
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 2026-I

1. COURSE

MA102FCCS. Differential Calculus (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA102FCCS. Differential Calculus
2.2 Semester	: 1 st Semester
2.3 Credits	: 5
2.4 Horas	: 4 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

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)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Functions and Limits (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Review of functions. • Definition of a limit. • Properties of limits. • Limits involving infinity. • Continuity.	• Evaluate limits graphically and numerically. [Familiarizarse] • Apply the properties of limits to evaluate limits algebraically. [Usar] • Determine the continuity of a function. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 2: The Derivative (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Definition of the derivative. • Geometric interpretation of the derivative. • Derivatives of polynomial and exponential functions. • Differentiation rules: sum, product, quotient, and chain rule.	• Calculate the derivative of a function using the definition. [Familiarizarse] • Interpret the derivative as the slope of the tangent line. [Usar] • Apply differentiation rules to find derivatives of functions. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 3: Applications of the Derivative (12 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Related rates. • Maximum and minimum values. • Mean Value Theorem. • Concavity and inflection points. • Optimization.	• Solve related rates problems. [Familiarizarse] • Find maximum and minimum values of a function. [Usar] • Apply the Mean Value Theorem. [Evaluar] • Determine the concavity and inflection points of a function. [Evaluar] • Solve optimization problems. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 4: Transcendental Functions (12 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Inverse trigonometric functions. • Hyperbolic functions. • Derivatives of inverse trigonometric and hyperbolic functions.	• Evaluate inverse trigonometric functions. [Familiarizarse] • Define and manipulate hyperbolic functions. [Usar] • Differentiate inverse trigonometric and hyperbolic functions. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 5: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Algorithm optimization. • Modeling dynamic systems. • Machine learning (e.g., gradient descent).	• Use derivatives to optimize algorithms. [Familiar-izarse] • Model dynamic systems using differential equations. [Usar] • Apply differential calculus in machine learning algorithms. [Evaluar]
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 2026-I

1. COURSE

CH101FCCS. Chemistry I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CH101FCCS. Chemistry I
2.2 Semester	: 1 st Semester
2.3 Credits	: 5
2.4 Horas	: 4 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

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. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (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. (Familiarity)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Introduction to Chemistry (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Matter and energy. • The scientific method. • Units of measurement. • Classification of matter.	• Describe the properties of matter and energy. [Familiarizarse] • Apply the scientific method to solve chemical problems. [Usar] • Perform unit conversions. [Evaluar]
Readings : [Bro+17], [CG16]	

Unit 2: Atomic Structure (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Structure of the atom. • Atomic number, mass number, and isotopes. • Quantum model of the atom. • Electron configuration.	• Describe the structure of the atom. [Familiarizarse] • Determine the electron configuration of an atom. [Usar] • Relate electron configuration to chemical properties. [Evaluar]
Readings : [Bro+17], [CG16]	

Unit 3: Chemical Bonds (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Ionic bonds. • Covalent bonds. • Metallic bonds. • Molecular geometry.	• Describe the different types of chemical bonds. [Familiarizarse] • Predict the molecular geometry of a molecule. [Usar] • Relate the type of bond to the properties of substances. [Evaluar]
Readings : [Bro+17], [CG16]	

Unit 4: Stoichiometry (8 hours)	
Competences Expected: 1,AG-C07,AG-C09	
Topics	Learning Outcomes
• Molar mass and mole. • Chemical reactions and chemical equations. • Stoichiometric calculations. • Limiting reactant and percent yield.	• Calculate the molar mass of a compound. [Familiarizarse] • Balance chemical equations. [Usar] • Perform stoichiometric calculations to determine the amount of reactants and products. [Evaluar]
Readings : [Bro+17], [CG16]	

Unit 5: States of Matter (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Gases. • Liquids. • Solids. • Changes of state.	• Describe the properties of the different states of matter. [Familiarizarse] • Explain changes of state and phase diagrams. [Usar] • Apply the ideal gas laws. [Evaluar]
Readings : [Bro+17], [CG16]	

Unit 6: Chemistry and Computing (10 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Molecular modeling. • Chemical simulations. • Materials science in computing.	• Describe how chemistry is used in molecular modeling. [Familiarizarse] • Explain the role of chemistry in chemical simulations. [Usar] • Analyze the importance of chemistry in materials science for computing. [Evaluar]
Readings : [Bro+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

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

FG001FCCS. General Elective (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: FG001FCCS. General Elective
2.2 Semester	: 1 st 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

General Education Elective

5. GOALS

- General Education Elective courses.

6. COMPETENCES

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02 Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

- 3) Communicate effectively in a variety of professional contexts.. (Familiarity)

AG-C04 Communication: Communicates effectively in complex computing activities. (Familiarity)

7. TOPICS

Unit 1: First Unit (48 hours)	
Competences Expected:	
Topics	Learning Outcomes
• General Education Elective courses.	• Develop skills and knowledge from courses outside their technical training in the degree program [Usar].
Readings : [Rob04]	

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

ID101. Technical and professional English I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: ID101. Technical and professional English I
2.2 Semester	: 1 st Semester
2.3 Credits	: 2
2.4 Horas	: 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.. (Assessment)

AG-C04) Communication: Communicates effectively in complex computing activities. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Familiarity)

7. TOPICS

Unit 1: Hello everybody! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Verb To Be. • Affirmative sentences , Negatives and Questions. • Number Expressions. • Objects and Countries. • Expressions to greet and make presentations.	• 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. • That the student is able to analyze and express ideas about dates and numbers in order.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Meeting people! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Possessive adjectives. • Expressions to find prices. • Possession expressions • Vocabulary of Family, Food and Drinks. • Formal requests. • Informal letters.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: The world of work (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Simple present tense. Auxiliaries. • Affirmative sentences, Negatives and Questions. • Common verbs and occupations. • Indications for expressing the time.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Take it easy! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present Simple 2. • Affirmative sentences , Negatives and Questions. • Use of Verbs of entertainment. • Free time. • The seasons of the year. • Expressions of social activities.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Where do you live? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Use There is/There are. • Sentences with Prepositions. • Expressions of Quantity. • Vocabulary of airplanes and places. • Expressions of direction indications.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Can you speak English? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Can/cant. • Past of verb to be. Use of Could • Vocabulary of Countries and languages. • Expressions for using the phone • Writing formal letters. • Readings.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Then and now! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Past Simple. • Expressions of past tense. • Vocabulary regular and irregular verbs • Expressions to describe the climate. • Writing descriptive paragraphs. • Special occasions.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

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

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

Chapter 2

Second Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS100-CS2023. Introduction to Computer Science (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS100-CS2023. Introduction to Computer Science
2.2 Semester	: 2 nd 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 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

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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (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. (Familiarity)

7. TOPICS

Unit 1: Computational Thinking (Part I) (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• General course instructions. • Explanation of the evaluation system. • Definition of Computing. • Computing as a Human-Computer dyad. • Distortions in the definition of computing. • Computing as the automation of abstraction. • Computing and Engineering: similarities and differences. • Algorithmic problem-solving. • Dynamics: Understanding the execution of an algorithm at human speed.	• Apply the fundamental concepts of computing in real-life situations. [Usar] • Identify distortions of Computing in real-life situations. [Usar] • Clearly identify at least 3 contexts of using the word“Engineer” in English. [Evaluar] • Identify the limitations of humans in solving computational problems. [Usar]
Readings : [BB19]	

Unit 2: Computational Thinking. Part II (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Binary vs. decimal numbering. • Character representation: the ASCII table. • Internal representation of colors. • Understanding an image from the inside. • Binary search. • Computational complexity of an algorithm.	• Apply various numbering systems to real-world problems. [Usar] • Understand the internal representation of characters in the ASCII and UTF-8 tables. [Familiarizarse] • Understand the representation of colors in an image. [Familiarizarse] • Apply Divide and Conquer algorithmic strategy. [Usar] • Determine basic analysis of algorithmic complexity. [Usar]
Readings : [BB19]	

Unit 3: Lógica digital y sistemas digitales (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial. • Múltiples representaciones / Capas de interpretación (El hardware es solo otra capa)	• 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] • 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] • 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] • 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] • 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]
Readings : [BB19]	

Unit 4: Representación de programas (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Programas que tienen otros programas como entradas tales como interpretes, compiladores, revisores de tipos y generadores de documentación. • Árboles de sintaxis abstracta, para contrastar la sintaxis correcta. • Estructuras de datos que representan código para ejecución, traducción o transmisión.	• Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse] • Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Usar] • Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse] • 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]
Readings : [BB19]	

Unit 5: Criptografía (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• 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. • Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos.	• Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse] • Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse]
Readings : [BB19]	

Unit 6: Organización y Arquitectura del Sistema de Memoria (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Sistemas de Almacenamiento y su Tecnología. • Jerarquía de Memoria: importancia de la localización temporal y espacial. • Organización y Operaciones de la Memoria Principal. • Latencia, ciclos de tiempo, ancho de banda e intercalación. • Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento) • Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core. • Memoria virtual (tabla de página, TLB) • Manejo de Errores y confiabilidad. • Error de codificación, compresión de datos y la integridad de datos.	• Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse]
Readings : [BB19]	

Unit 7: Visión general de Sistemas Operativos (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Los mecanismos de apoyo modelos cliente-servidor, dispositivos de mano. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.	• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Usar] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]
Readings : [BB19]	

Unit 8: Introducción a redes (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Técnicas de Switching (por ejemplo, de circuitos, de paquetes) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Principios de capas (encapsulación, multiplexación) • Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)	• Articular la organización de la Internet [Familiarizarse] • Listar y definir la terminología de red apropiada [Familiarizarse] • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse] • Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]
Readings : [BB19]	

Unit 9: Entrega confiable de datos (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Control de errores (técnicas de retransmisión, temporizadores) • El control de flujo (agradecimientos, ventana deslizante) • Problemas de rendimiento (pipelining) • TCP	• Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse] • Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse] • Diseñar e implementar un protocolo confiable simple [Usar]
Readings : [BB19]	

Unit 10: Análisis Básico (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. • Análisis asintótico de complejidad de cotas superior y esperada. • Definición formal de la Notación Big O. • Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. • Medidas empíricas de desempeño. • Compensación entre espacio y tiempo en los algoritmos. • Uso de la notación Big O.	• Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Familiarizarse] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar] • Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Usar] • Indique la definición formal de Big O [Familiarizarse] • Lista y contraste de clases estándares de complejidad [Familiarizarse] • 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] • Da ejemplos que ilustran las compensaciones entre espacio y tiempo que se dan en los algoritmos [Familiarizarse] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Familiarizarse]
Readings : [BB19]	

Unit 11: Algoritmos y Estructuras de Datos fundamentales (8 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
- Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. - Algoritmos de búsqueda secuencial y binaria. - Algoritmos de ordenamiento de peor caso cuadrático (selección, inserción) - Algoritmos de ordenamiento con peor caso o caso promedio en $O(N \lg N)$ (Quicksort, Heapsort, Mergesort) - Tablas Hash, incluyendo estrategias para evitar y resolver colisiones. - Árboles de búsqueda binaria: - Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles. - Grafos y algoritmos en grafos: - Representación de grafos (ej., lista de adyacencia, matriz de adyacencia) - Recorrido en profundidad y amplitud - Montículos (Heaps) - Grafos y algoritmos en grafos: - Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) - Árbol de expansión mínima (algoritmos de Prim y Kruskal) - Búsqueda de patrones y algoritmos de cadenas/texto (ej. búsqueda de subcadena, búsqueda de expresiones regulares, algoritmos de subsecuencia común más larga)	- Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse] - Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar] - Implementar algoritmos numéricos básicos [Usar] - Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar] - Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar] - Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse] - Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse] - 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] - Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse] - 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]
Readings : [BB19]	

Unit 12: Sistemas de Bases de Datos (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Enfoque y Evolución de Sistemas de Bases de Datos. • Componentes del Sistema de Bases de Datos. • Diseño de las funciones principales de un DBMS. • Arquitectura de base de datos e independencia de datos. • Uso de un lenguaje de consulta declarativa. • Sistemas de apoyo a contenido estructurado y / o corriente. • Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce).	• Describe los enfoques principales para almacenar y procesar largos volúmenes de datos [Familiarizarse] • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse] • Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Familiarizarse] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse] • Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Familiarizarse] • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse] • Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar] • Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Familiarizarse]
Readings : [BB19]	

Unit 13: Programación orientada a objetos (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Diseño orientado a objetos: – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Definición de las categorías, campos, métodos y constructores. • Las subclases, herencia y método de alteración temporal. • Asignación dinámica: definición de método de llamada. • Subtipificación: – Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos. – Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos. – Relación entre subtipos y la herencia. • Lenguajes orientados a objetos para la encapsulación: – privacidad y la visibilidad de miembros de la clase – Interfaces revelan único método de firmas – clases base abstractas • Uso de colección de clases, iteradores, y otros componentes de la librería estándar.	• Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar] • Comparar y contrastar (1) el enfoque procedur/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 [Evaluar] • Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma más natural por cada lenguaje [Usar]
Readings : [BB19]	

Unit 14: Procesos de Software (4 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno. • Introducción a modelos del proceso de software (e.g., cascada, incremental, ágil): – Actividades con ciclos de vida de software. • Programación a gran escala versus programación individual. • Evaluación de modelos de proceso de software. • Conceptos de calidad de software. • Mejoramiento de procesos. • Modelos de madurez de procesos de software. • Mediciones del proceso de software.	• 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] • Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Familiarizarse] • Diferenciar entre las fases de desarrollo de software [Familiarizarse] • 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] • Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Familiarizarse] • 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] • Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Familiarizarse]
Readings : [BB19]	

Unit 15: Cuestiones fundamentales (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? – El Test de Turing – Razonamiento Racional versus No Racional	• Describir el test de Turing y el experimento pensado cuarto chino” (<i>Chinese Room</i>) [Familiarizarse]
Readings : [BB19]	

Unit 16: Estrategias de búsquedas básicas (1 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda. • Factored representation (factoring state hacia variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*) • El espacio y el tiempo de la eficiencia de búsqueda. • Dos jugadores juegos (introducción a la búsqueda minimax). • Satisfacción de restricciones (backtracking y métodos de búsqueda local).	• 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] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Familiarizarse] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Familiarizarse] • Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar] • Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar] • Evalúa si una heurística dada para un determinado problema es admisible/puede garantizar una solución óptima [Evaluar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Familiarizarse]
Readings : [BB19]	

Unit 17: Aprendizaje Automático Básico (1 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación. • Aprendizaje inductivo • Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión. • El problema exceso de ajuste. • Medición clasificada con exactitud.	• Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Familiarizarse] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Familiarizarse] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Familiarizarse]
Readings : [BB19]	

Unit 18: Conceptos Fundamentales (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual. • 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 • El uso de las API estándar para la construcción de interfaces de usuario y visualización de formatos multimedia estándar • Formatos estándar, incluyendo formatos sin pérdidas y con pérdidas. • Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores. • Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes. • Animación como una secuencia de imágenes fijas. • Almacenamiento doble.	• Identificar usos comunes de presentaciones digitales de humanos (por ejemplo, computación gráfica,sonido) [Familiarizarse] • 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] • Explicar cómo las limitaciones en la percepción humana afectan la selección de la representación digital de señales analógicas [Usar] • 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] • Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse] • 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] • 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] • Describir cómo el doble buffer puede eliminar el parpadeo de la animación [Familiarizarse]
Readings : [BB19]	

Unit 19: Rendering Básico (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización) • Representación poligonal • Radiometría básica, triángulos similares y modelos de proyecciones • Afinamiento y Transformaciones de Sistemas de coordenadas • <i>Ray tracing</i> • Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos. • Representación de la ecuación de adelante hacia atrás. • Rasterización triangular simple. • Mapeo de texturas, incluyendo minificación y magnificación (e.g., MIP-mapping trilineal) • Aplicación de la representación de estructuras de datos espaciales. • Muestreo y anti-aliasing. • Gráficos en escena y la canalización de gráficos.	• 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] • Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse] • Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar] • Derivar la perspectiva lineal de triángulos semejantes por conversión de puntos (x,y,z) a puntos (x/z, y/z, 1) [Usar] • Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afin [Usar] • 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] • Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse] • Implementar simples procedimientos que realicen la transformación y las operaciones de recorte de imágenes simples en 2 dimensiones [Usar] • Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar] • Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]
Readings : [BB19]	

Unit 20: Closure Class: How Does a Search Engine Like Google Work? (2 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Problem analysis • The index does not grow linearly with the size of the indexed information. • Response time does not depend on the size of the “database.” • Response time does not depend on the number of occurrences found. • Combining various data structures to reach a solution. • Analyzing the scalability of the solution.	• Understand the principles under which a search engine is created [Usar] • Correctly apply data structures to solve the problem [Usar] • Apply concepts related to algorithmic complexity in a search engine [Usar].
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 2026-I

1. COURSE

CS112-CS2023. Objects oriented programming I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS112-CS2023. Objects oriented programming I
2.2 Semester	: 2 nd 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	: CS111-CS2023. Introduction to Programming. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the second course in the introductory Computer Science programming sequence.

It incorporates a programming language transition as a pedagogical strategy and focuses on core Object-Oriented Programming (OOP) concepts, expanding on the foundations from the first course while introducing low-level programming principles.

The course provides a solid foundation for advanced topics covered in the subsequent course of the sequence.

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

- 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)

- 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)

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

Unit 1: Overview of Programming Languages (1 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Brief review of programming paradigms. • Comparison between functional and imperative programming. • History of programming languages (emphasis on C and C++).	• Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse]
Readings : [Str13], [Jos19], [Dei17]	

Unit 2: Máquinas virtuales (2 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• The concept of a virtual machine. • Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) . • Intermediate languages.	• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 3: Sistemas de tipos básicos (6 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Type systems in programming languages. • Declaration models (linking, visibility, scope, and lifetime). • Overview of type checking.	• Tanto para tipo primitivo y un tipo compuesto, describir de manera informal los valores que tiene dicho tipo [Familiarizarse] • 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] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar] • Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Familiarizarse] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar] • Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Familiarizarse] • Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar] • Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Familiarizarse] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar] • Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Familiarizarse] • Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Familiarizarse]
Readings : [Str13], [Jos19], [Dei17]	

Unit 4: Conceptos Fundamentales de Programación (6 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Diseño orientado a objetos: – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Variables and data types. • Expressions and operators. • Conditional statements.	• 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] • Identifica y describe el uso de tipos de datos primitivos [Familiarizarse] • Escribe programas que usan tipos de datos primitivos [Usar] • Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar] • 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] • Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Evaluar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 5: Functions (6 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Paso de funciones y parámetros. • Parameter passing. • Function overloading. • Fundamentals of recursion. • Function templates.	• 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] • Understand and apply the concept of parameter passing to a function, both by value and by reference. [Usar] • Identify and apply the concept of function overloading. [Usar] • Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse] • Design, implement, and apply the concept of templates to create generic functions. [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 6: Arrays, Pointers, and Memory Management (8 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Array definition. • Multidimensional arrays. • Pointer fundamentals. • Dynamic memory management (new/delete, stack vs. heap). • Smart pointers (unique_ptr, shared_ptr, weak_ptr). • Advanced pointer concepts (pointers to pointers, pointers to functions).	• Understand and implement one-dimensional arrays. [Familiarizarse] • Design and apply the concept of multidimensional arrays. [Usar] • Understand and apply the concept of references and pointers. [Familiarizarse] • Understand, apply, and evaluate the relationship between pointers and arrays. [Evaluar] • Understand and implement dynamic memory management. Differentiate between heap and stack memory regions. [Evaluar] • Design, implement, and evaluate concepts like pointer-to-pointer, pointer-to-function, among other advanced pointer concepts. [Evaluar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 7: Working with Arrays and Pointers (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Arrays as function arguments. • Character arrays and pointers. • Pointers and 2-dimensional arrays. • Pointers and multidimensional arrays.	• Demonstrate the use of pointers with different types of arrays. [Usar] • Demonstrate the layout of an array in memory and how pointers are manipulated within those memory spaces. [Usar] • Demonstrate the use of pointer arithmetic with arrays. [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 8: Pointers and Dynamic Memory (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Pointers and dynamic memory - stack vs heap. • Pointers as return values from a function in C/C++. • Pointers to functions in C/C++. • Pointers to functions and callbacks. • Memory leaks in C/C++.	• Show the memory structure within a program and understand how the compiler allocates elements on the stack and heap. [Usar] • Demonstrate the use of functions and operators for dynamic memory allocation and deallocation. [Usar] • Understand the implications of returning pointers from functions. [Usar] • Use pointers to functions as parameters. [Usar] • Understand the implications of dynamic memory usage and memory leaks. [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 9: Pointers and Classes (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Pointers to class members - attributes. • Pointers to class members - methods and calls to method pointers. • Pointers to class members - static methods and calls to static method pointers. • Pointers to classes - example with linked list management.	• Understand the use of pointers to different elements of a class. [Usar] • Understand the use of pointers to static members of a class. [Usar] • Introduce the node structure and its use in a simple data structure. [Usar] • Introduce data structures, showing a simple implementation of linked lists.[Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 10: Programación orientada a objetos (8 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Diseño orientado a objetos: – Descomposición en objetos que almacenan estados y poseen comportamiento – Diseño basado en jerarquía de clases para modelamiento • Lenguajes orientados a objetos para la encapsulación: – privacidad y la visibilidad de miembros de la clase – Interfaces revelan único método de firmas – clases base abstractas • Definición de las categorías, campos, métodos y constructores. • Las subclases, herencia y método de alteración temporal. • Subtipificación: – Polimorfismo artículo Subtipo; upcasts implícitos en lenguajes con tipos. – Noción de reemplazo de comportamiento: los subtipos de actuar como supertipos. – Relación entre subtipos y la herencia. • Uso de colección de clases, iteradores, y otros componentes de la librería estándar. • Asignación dinámica: definición de método de llamada.	• Diseñar e implementar una clase [Usar] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar] • Comparar y contrastar (1) el enfoque procedur/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 [Evaluar] • Explicar la relación entre la herencia orientada a objetos (código compartido y <i>overriding</i>) y subtipificación (la idea de un subtipo es ser utilizable en un contexto en el que espera al supertipo) [Familiarizarse] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma más natural por cada lenguaje [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 11: Templates and STL (6 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Class templates. • Basic concepts of the Standard Template Library (STL) including: vector, list, stack, queue.	• Understand the concepts of class templates. [Familiarizarse] • Implement and create new generic data types. [Usar] • Understand the basic structures of the STL. [Familiarizarse] • Use basic data structures like stack, queue, list, and vector from the STL. [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 12: Operator Overloading (4 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Operator overloading definition.	• Understand the concepts of operator overloading. [Familiarizarse] • Implement the overloading of allowed operators in the programming language. [Usar]
Readings : [Str13], [Jos19], [Dei17]	

Unit 13: File Handling (4 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• File input and output (I/O).	• Understand the concepts of file manipulation. [Familiarizarse] • Create programs to read and write files. [Usar]
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 2026-I

1. COURSE

CS1D1-2-CS2023. Discrete Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS1D1-2-CS2023. Discrete Structures
2.2 Semester	: 2 nd 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

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. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Usage)

AG-C10) Investigation: Studies complex computing problems using information science methods. (Familiarity)

7. TOPICS

Unit 1: Funciones, relaciones y conjuntos (22 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Conjuntos: – Diagramas de Venn – Unión, intersección, complemento – Producto Cartesiano – Potencia de conjuntos – Cardinalidad de Conjuntos finitos • Relations: – Reflexivity, symmetry, transitivity – Equivalence relations – Partial order relations and sets – Extremal elements of a partially ordered sets • Funciones: – Suryecciones, inyecciones, biyecciones – Inversas – Composición	• Explicar con ejemplos la terminología básica de funciones, relaciones y conjuntos [Evaluar] • Realizar las operaciones asociadas con conjuntos, funciones y relaciones [Evaluar] • Relacionar ejemplos prácticos para conjuntos funciones o modelos de relación apropiados e interpretar la asociación de operaciones y terminología en contexto [Evaluar]
Readings : [Gri03], [Ros07], [Vel06]	

Unit 2: Lógica básica (14 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Lógica proposicional. • Conectores lógicos. • Tablas de verdad. • Forma normal (conjuntiva y disyuntiva) • Validación de fórmula bien formada. • Reglas de inferencia proposicional (conceptos de modus ponens y modus tollens) • Logica de predicados: – Cuantificación universal y existencial • Limitaciones de la lógica proposicional y de predicados (ej. problemas de expresividad)	• Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Usar] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar] • Usar reglas de inferencia para construir demostraciones en lógica proposicional y de predicados [Usar] • 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] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar] • Describir las fortalezas y limitaciones de la lógica proposicional y de predicados [Usar]
Readings : [Ros07], [Gri03], [Vel06]	

Unit 3: Técnicas de demostración (14 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Nociones de implicancia, equivalencia, conversión, inversa, contrapositivo, negación, y contradicción • Estructura de pruebas matemáticas. • Demostración directa. • Refutar por contraejemplo. • Demostración por contradicción. • Inducción sobre números naturales. • Inducción estructural. • Inducción leve y fuerte (Ej. Primer y Segundo principio de la inducción) • Definiciones matemáticas recursivas. • Conjuntos bien ordenados.	• Identificar la técnica de demostración utilizada en una demostración dada [Evaluar] • Describir la estructura básica de cada técnica de demostración (demostración directa, demostración por contradicción e inducción) descritas en esta unidad [Usar] • Aplicar las técnicas de demostración (demostración directa, demostración por contradicción e inducción) correctamente en la construcción de un argumento solido [Usar] • Determine que tipo de demostración es la mejor para un problema dado [Evaluar] • Explicar el paralelismo entre ideas matemáticas y/o inducción estructural para la recursión y definir estructuras recursivamente [Familiarizarse] • Explicar la relación entre inducción fuerte y débil y dar ejemplos del apropiado uso de cada uno [Evaluar] • Enunciar el principio del buen-orden y su relación con la inducción matemática [Familiarizarse]
Readings : [Ros07], [Vel06], [Sch12], [Vel06]	

Unit 4: Data Representation (10 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Numerical representation: sign-magnitude, floating point. • Representation of other objects: sets, relations, functions.	• Explain numerical representations such as sign-magnitude and floating point. [Evaluar]. • Carry out arithmetic operations using different kinds of representations. [Evaluar]. • Explain the floating point standard IEEE-754 [Familiarizarse].
Readings : [Ros07], [Gri03], [Vel06]	

Unit 5: Digital Logic and Data Representation (10 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Reticles: Types and properties. • Boolean algebras. • Boolean Functions and Expressions. • Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form. • Logical gates. • Circuit Minimization.	• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Evaluar]. • Explain the algebraic structures of reticulum and its types [Evaluar]. • Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Evaluar]. • Explain the properties that satisfies a Boolean algebra [Evaluar]. • Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Evaluar]. • Find the canonical forms of a Boolean function [Evaluar]. • Represent a Boolean function as a Boolean circuit using logic gates [Evaluar]. • Minimize a Boolean function. [Evaluar].
Readings : [Ros07], [Gri03]	

Unit 6: Fundamentos de conteo (40 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Técnicas de Conteo: – Conteo y cardinalidad de un conjunto – Regla de la suma y producto – Principio de inclusión-exclusión – Progresión geométrica y aritmética • Principio de las casillas. • Permutaciones y combinaciones: – Definiciones básicas – Identidad de Pascal – Teorema del binomio • Resolviendo relaciones de recurrencia: – Un ejemplo de una relación de recurrencia simple, como los números de Fibonacci – Otras ejemplos, mostrando una variedad de soluciones • Aritmetica modular basica	• Aplicar argumentos de conteo, incluyendo las reglas del producto y de la suma, principio de inclusión-exclusión y progresiones aritméticas/geométricas [Familiarizarse] • Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse] • Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse] • 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] • Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse] • Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse] • Realizar cálculos que involucran aritmética modular [Familiarizarse]
Readings : [Gri97]	

Unit 7: Árboles y Grafos (40 hours)	
Competences Expected: 1,6,AG-C08,AG-C10,AG-C12	
Topics	Learning Outcomes
• Árboles. – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Árboles de expansión/bosques. • Isomorfismo en grafos.	• 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] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse] • 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] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse] • Explicar como construir un árbol de expansión de un grafo [Familiarizarse] • Determinar si dos grafos son isomorfos [Familiarizarse]
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.
- [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 2026-I

1. COURSE

MA103FCCS. Integral Calculus (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA103FCCS. Integral Calculus
2.2 Semester	: 2 nd Semester
2.3 Credits	: 5
2.4 Horas	: 4 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	: MA102FCCS. Differential Calculus. (1 st 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. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Familiarity)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Familiarity)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: The Indefinite Integral (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Antiderivatives and the indefinite integral. • Basic integration rules. • Integration by substitution.	• Calculate antiderivatives of basic functions. [Familiarizarse] • Apply the basic integration rules. [Usar] • Solve indefinite integrals using the substitution technique. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 2: The Definite Integral (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Riemann sums and the definite integral. • The Fundamental Theorem of Calculus. • Calculating areas.	• Approximate definite integrals using Riemann sums. [Familiarizarse] • Apply the Fundamental Theorem of Calculus to evaluate definite integrals. [Usar] • Calculate areas under curves using definite integrals. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 3: Techniques of Integration (12 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Integration by parts. • Integration of trigonometric functions. • Integration by partial fractions. • Improper integrals.	• Apply the technique of integration by parts. [Familiarizarse] • Integrate trigonometric functions using identities and substitution techniques. [Usar] • Solve integrals using the technique of partial fractions. [Evaluar] • Evaluate improper integrals. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 4: Applications of the Definite Integral (12 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
• Calculating areas between curves. • Calculating volumes of solids of revolution. • Arc length. • Work, average value, and centroids.	• Calculate the area between two curves using definite integrals. [Familiarizarse] • Calculate the volume of solids of revolution using different methods. [Usar] • Calculate the arc length of a curve. [Evaluar] • Apply integrals to calculate work, average value, and centroids. [Evaluar]
Readings : [Ste15], [LE14]	

Unit 5: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
- Algorithm analysis (e.g., calculating time complexity). - Signal and image processing (e.g., integral transforms). - Probability and statistics (e.g., probability density functions).	- Use integrals to analyze the time complexity of algorithms. [Familiarizarse] - Apply integral transforms in signal and image processing. [Usar] - Use integrals in the context of probability density functions. [Evaluar]
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 2026-I

1. COURSE

FI101FCCS. Physics I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: FI101FCCS. Physics I
2.2 Semester	: 2 nd Semester
2.3 Credits	: 5
2.4 Horas	: 4 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

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)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Usage)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)

7. TOPICS

Unit 1: Kinematics (8 hours)	
Competences Expected: 1,AG-C07,AG-C08	
Topics	Learning Outcomes
• Displacement, velocity, and acceleration. • Uniform and uniformly accelerated linear motion. • Projectile motion. • Uniform circular motion.	• Define and calculate displacement, velocity, and acceleration. [Familiarizarse] • Solve problems involving linear motion and projectile motion. [Usar] • Analyze uniform circular motion. [Evaluar]
Readings : [YF18], [SJ18]	

Unit 2: Dynamics (10 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Newton's laws of motion. • Forces of friction. • Work and energy. • Work-kinetic energy theorem. • Power.	• State and apply Newton's laws of motion. [Familiarizarse] • Calculate the work done by a force. [Usar] • Apply the work-kinetic energy theorem to solve dynamics problems. [Evaluar]
Readings : [YF18], [SJ18]	

Unit 3: Conservation of Energy (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Potential energy. • Conservation of mechanical energy. • Conservative and non-conservative forces.	• Define and calculate potential energy. [Familiarizarse] • Apply the principle of conservation of mechanical energy. [Usar] • Distinguish between conservative and non-conservative forces. [Evaluar]
Readings : [YF18], [SJ18]	

Unit 4: Systems of Particles and Conservation of Linear Momentum (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Center of mass. • Linear momentum. • Conservation of linear momentum. • Collisions.	• Calculate the center of mass of a system of particles. [Familiarizarse] • Apply the principle of conservation of linear momentum. [Usar] • Analyze elastic and inelastic collisions. [Evaluar]
Readings : [YF18], [SJ18]	

Unit 5: Rotation (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Rotational kinematics. • Rotational dynamics. • Moment of inertia. • Torque and rotational kinetic energy.	• Describe rotational motion using angular variables. [Familiarizarse] • Calculate the moment of inertia of simple objects. [Usar] • Apply the laws of rotational dynamics. [Evaluar]
Readings : [YF18], [SJ18]	

Unit 6: Applications in Computing (8 hours)	
Competences Expected: 1,AG-C07,AG-C12	
Topics	Learning Outcomes
• Physical simulations. • Computer graphics. • Robotics.	• Explain how the principles of physics are used in physical simulations. [Familiarizarse] • Describe the application of physics in computer graphics. [Usar] • Analyze the use of physics in robotics. [Evaluar]
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 2026-I

1. COURSE

ID102. Technical and professional English II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: ID102. Technical and professional English II
2.2 Semester	: 2 nd Semester
2.3 Credits	: 2
2.4 Horas	: 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	: ID101. Technical and professional English I. (1 st 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

3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04) Communication: Communicates effectively in complex computing activities. (Usage)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Usage)

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)

7. TOPICS

Unit 1: How long ago? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Simple past • Negative sentences with ago. • Conjunctions • Expressions of Time in past • Phonetic relations and symbols • Expressions to give the date	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Food you like! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Accounting and Non-Accounting Nouns • Expressions with Would like and I'd like • Quantifiers • Meals around the world • Formal requests • Formal letters	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: The world of work (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Adjectives • Sentences with Comparative Adjectives • Sentences with Superlative Adjectives • Cities and countryside • Directional indications	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Looking good! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present continuous • Affirmative sentences, Negatives and Questions • Use of Whose • Possessive pronouns • Clothing and colors • Expressions to wear in clothing stores • Phonetic symbols.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Life is an adventure! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Use of going to • Future time sentences • Expressions of Quantity. • Action verbs • Vocabulary of the climate • Expressions of Suggestion • Write a postcard	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: You're pretty smart! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Forms of Questions • Adverbs and Adjectives • Vocabulary description of feelings • Expressions for train travel • Writing Short Stories • Readings	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Have you ever? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Perfect present • Keywords with never, ever, and yet • Vocabulary verbs in Past participle • Expressions for airplane travel • Writing thank-you letters • Readings	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

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

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

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

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

Chapter 3

Third Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS113-CS2023. Objects oriented programming II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS113-CS2023. Objects oriented programming II
2.2 Semester	: 3 rd 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	: CS112-CS2023. Objects oriented programming I. (2 nd 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 explores C++-specific OOP mastery, focusing on high-performance systems development. Key topics include:

Core Advanced Concepts:

- Template metaprogramming (TMP) and Substitution Failure Is Not An Error (SFINAE)
- Move semantics, perfect forwarding, and Resource Acquisition Is Initialization (RAII) optimization
- Multiple inheritance pitfalls and virtual inheritance patterns

Concurrency & Systems Programming:

- `std::thread`, `async/await`, and thread synchronization
- Lock-free programming with atomic types
- Thread-safe OOP design patterns (e.g., singleton variants)

Modern C++ Paradigms:

- CRTP (Curiously Recurring Template Pattern)
- Expression templates for performance-critical code
- Compile-time introspection with `constexpr` and type traits

Real-World Applications:

- Interfacing C++ with other languages (FFI)
- Benchmarking and profiling template-heavy code

Prepares students for game engines, high performance computing, and embedded development where C++ dominates.

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

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)

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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Familiarity)

7. TOPICS

Unit 1: Advanced STL (8 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Associative Containers (<code>std::set</code>, <code>std::map</code>, <code>std::unordered_set</code>, <code>std::unordered_map</code>). • Adapters (<code>std::stack</code>, <code>std::queue</code>, <code>std::priority_queue</code>). • Advanced STL Algorithms. • Functors and Predicates.	• Understand the use of associative containers. [Usar] • Implement programs that use STL adapters. [Usar] • Apply advanced STL algorithms. [Usar] • Use functors and predicates with the STL. [Usar]
Readings : [Str13], [MJo19], [VJG17], [Dei17]	

Unit 2: Advanced Templates (7 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Template Metaprogramming. • SFINAE (Substitution Failure Is Not An Error). • Perfect Forwarding.	• Apply template metaprogramming to solve complex problems. [Usar] • Understand and use SFINAE for template selection. [Usar] • Use Perfect Forwarding for efficient argument passing. [Usar]
Readings : [Str13], [MJo19], [VJG17], [Dei17]	

Unit 3: Variadic Templates (12 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Introduction to Variadic Templates. • Variadic Template Functions. • Variadic Template Methods. • Variadic Template Classes. • Classes inheriting from variable lists of variadic templates. • Example: Implementing a custom tuple class.	• Understand the concept of variadic templates. [Familiarizarse] • Implement variadic functions. [Usar] • Design classes with variadic methods. [Usar] • Create classes with variadic templates. [Usar] • Implement inheritance with variable lists of variadic templates. [Usar] • Apply variadic templates to solve real-world problems. [Usar]
Readings : [Str13], [MJo19], [VJG17], [Dei17]	

Unit 4: Move Semantics and Rvalue References (5 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Lvalues and Rvalues. • Rvalue References. • Move Semantics. • Move Constructors and Move Assignment Operators. • Perfect Forwarding (review).	• Explain move semantics and its purpose in C++. [Familiarizarse] • Define and use rvalue references. [Usar] • Analyze the performance implications of using move semantics. [Evaluar] • Implement move constructors and move assignment operators for custom classes. [Usar] • Apply move semantics to optimize resource management in C++ programs. [Usar]
Readings : [Str13], [MJo19], [VJG17], [Dei17]	

Unit 5: Design Patterns (Creational and Structural) (6 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Singleton, Factory, Builder. • Adapter, Decorator, Facade.	• Understand and apply creational design patterns: Singleton, Factory, Builder. [Usar] • Understand and apply structural design patterns: Adapter, Decorator, Facade. [Usar]
Readings : [Str13], [MJo19], [VJG17], [Dei17]	

Unit 6: Functors (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Definition of Functors. • Functors and Templates. • Passing Functors to Functions using parameters. • Passing Functors to Functions using templates. • Passing Functors to Classes using parameters. • Passing Functors to Classes using templates. • Examples and Applications.	• Introduction to Functors. [Usar] • Using Functors as parameters to functions and classes. [Usar] • Using Functors in functions and classes through templates. [Usar]
Readings : [Str13], [MJo19], [VJG17], [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 edition*. Addison-Wesley, 2013.
- [Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017.
- [VJG17] David Vandervoorde, Nicolai Josuttis, and Douglas Gregor. *C++ Templates: The Complete Guide*. 2nd. Addison-Wesley, 2017.
- [MJo19] Nicolai M. Josuttis. *C++17-The Complete Guide*. 1st. 2019.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS2B1-CS2023. Platform Based Development (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS2B1-CS2023. Platform Based Development
2.2 Semester	: 3 rd 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	: CS112-CS2023. Objects oriented programming I. (2 nd 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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Usage)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Familiarity)

7. TOPICS

Unit 1: Introducción (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial) • Programación a través de APIs específicos. • Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5) • Programación bajo restricciones de plataforma.	• Describir cómo el desarrollo basado en plataforma difiere de la programación de propósito general [Familiarizarse] • Listar las características de lenguajes de plataforma [Familiarizarse] • Escribir y ejecutar un programa simple basado en plataforma [Familiarizarse] • Listar las ventajas y desventajas de la programación con restricciones de plataforma [Familiarizarse]
Readings : [Fie00], [Fie21], [Gro09], [ADC13]	

Unit 2: Plataformas web (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Lenguajes de programación web (e.g., HTML5, Javascript, PHP, CSS) • • Web Platform constraints: Client-Server, Stateless-Stateful, Cache, Uniform Interface, Layered System, Code on Demand, ReST. • Restricción de plataformas web. • Software como servicio. • Estándares web.	• Diseñar e implementar una aplicación web sencilla [Familiarizarse] • Describir las limitaciones que la web pone a los desarrolladores [Familiarizarse] • Comparar y contrastar la programación web con la programación de propósito general [Familiarizarse] • Describir las diferencias entre software como un servicio y productos de software tradicionales [Familiarizarse] • Discutir cómo los estándares de web impactan el desarrollo de software [Familiarizarse] • Revisar una aplicación web existente con un estándar web actual [Familiarizarse]
Readings : [Fie00], [Fie21]	

Unit 3: Desarrollo de servicios y aplicaciones web (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Describe, identify and debug issues related to web application development - Design and development of interactive web applications using HTML5 and Python - Use MySQL for data management and manipulate MySQL with Python - Design and development of asynchronous web applications using Ajax techniques - Using dynamic client side Javascript scripting language and server side python scripting language with Ajax - Apply XML / JSON technologies for data management with Ajax - Use framework, services and Ajax web APIs and apply design patterns to web application development	- Server-side python scripting language: variables, data types, operations, strings, functions, control statements, arrays, files and directory access, maintain state. [Usar] - Web programming approach using embedded python. [Usar] - Accessing and Manipulating MySQL. [Usar] - The Ajax web application development approach. [Usar] - DOM and CSS used in JavaScript. [Usar] - Asynchronous Content Update Technologies. [Usar] - XMLHttpRequest objects use to communicate between clients and servers. [Usar] - XML and JSON. [Usar] - XSLT and XPath as mechanisms for transforming XML documents. [Usar] - Web services and APIs (especially Google Maps). [Usar] - Macros Ajax for the development of contemporary web applications. [Usar] - Design patterns used in web applications. [Usar]
Readings : [FR11]	

Unit 4: Plataformas móviles (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Lenguajes de Programación para Móviles. - Design Principles: Segregation of Interfaces, Single Responsibility, Separation of concerns, Dependency Inversion. - Desafíos con movilidad y comunicación inalámbrica. - Aplicaciones Location-aware. - Rendimiento / Compensación de Potencia. - Restricciones de las Plataformas Móviles. - Tecnologías Emergentes.	- Diseñar e implementar una aplicación móvil para una plataforma móvil dada [Familiarizarse] - Discutir las limitaciones que las plataformas móviles ponen a los desarrolladores [Familiarizarse] - Discutir el rendimiento vs pérdida de potencia [Familiarizarse] - Compare y contraste la programación móvil con la programación de proposito general [Familiarizarse]
Readings : [Mar17], [ADC13]	

Unit 5: Mobile Applications for Android Handheld Systems (25 hours)	
Competences Expected:	
Topics	Learning Outcomes
• The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps	• Students identify necessary software and install it on their personal computers. • Students perform various tasks to familiarize themselves with the Android platform and Environment for development. [Usar] • 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] • Students build applications that require starting multiple activities through both standard and custom methods. [Usar] • Students build applications that require standard and custom permissions. [Usar] • Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usar] • 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] • Students build an application that uses location information to collect latitude, length of places they visit. [Usar]
Readings : [ADC13]	

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

- [Fie00] Roy Thomas Fielding. "Fielding dissertation: Chapter 5: Representational state transfer (rest)". In: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (2000).
- [Gro09] R. Grove. *Web Based Application Development*. Jones & Bartlett Learning, 2009.
- [FR11] Eric Freeman and Elisabeth Robson. *Head first HTML5 programming: building web apps with JavaScript*. O'Reilly Media, Inc., 2011.

-
- [ADC13] J. Annuzzi, L. Darcey, and S. Conder. *Introduction to Android Application Development: Android Essentials*. Developer's Library. Pearson Education, 2013.
- [Mar17] Robert C Martin. *Clean architecture: a craftsman's guide to software structure and design*. Prentice Hall Press, 2017.
- [Fie21] Roy T. Fielding. "REST APIs Must Be Hypertext-Driven: Revisiting the Architectural Constraints". In: *IEEE Software* 38.3 (2021). Artículo del creador de REST actualizando sus principios para aplicaciones modernas., pp. 27–32.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

AI161. Applied AI (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: AI161. Applied AI
2.2 Semester	: 3 rd 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	: CS111. Introduction to Programming. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course provides a practical introduction to Artificial Intelligence (AI) for students from all scientific and engineering disciplines. Focused on developing AI literacy and practical skills, it covers fundamental concepts, modern AI tools (including Western and Chinese platforms), and responsible usage. Students will learn to effectively interact with diverse AI systems, write quality prompts, and apply AI solutions to problems across various domains while understanding ethical implications and cultural contexts of AI deployment.

5. GOALS

- Develop comprehensive AI literacy by understanding fundamental concepts, capabilities, and limitations of modern AI systems across different platforms and cultures.
- Master effective prompt engineering techniques and interaction patterns with various AI tools (Western: ChatGPT, Claude, Gemini; Chinese: DeepSeek, Kimi, ERNIE).
- Apply diverse AI tools to solve practical problems in scientific, engineering, and academic contexts while maintaining critical evaluation of outputs across platforms.
- Understand ethical considerations, biases, cultural contexts, and responsible usage of AI technologies in global professional and academic settings.
- Develop workflows that integrate multiple AI tools to enhance productivity and problem-solving capabilities while understanding regional strengths and specializations.

6. COMPETENCES

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

7. TOPICS

Unit 1: AI Fundamentals and Global Landscape (16 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
- What is AI? Definitions, history, and current global landscape. - Types of AI systems: chatbots, image generators, research assistants across regions. - Western AI ecosystems: OpenAI, Anthropic, Google, Microsoft. - Chinese AI ecosystems: DeepSeek, Kimi, ERNIE, Zhipu AI, Baidu. - AI capabilities and limitations: comparative analysis across platforms. - Digital literacy in the AI era: critical thinking about diverse AI outputs.	- Apply fundamental AI knowledge to identify system types and their capabilities [Usar]. - Analyze global intelligent systems and their distinctive characteristics [Evaluar]. - Critically evaluate AI outputs across different cultural contexts [Evaluar].
Readings : [Ng24], [MM24], [UNE23], [Ins24b]	

Unit 2: Effective AI Interaction and Cross-Platform Prompt Engineering (20 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
- Principles of effective prompt writing: clarity, context, constraints across platforms. - Prompt patterns: persona, template, chain-of-thought, few-shot for different AI systems. - Western tools: specific techniques for ChatGPT, Claude, Gemini, Copilot. - Chinese tools: specific features and best practices for DeepSeek, Kimi, ERNIE. - Cross-platform strategies: leveraging different AI strengths for complex tasks. - Iterative refinement: how to improve prompts based on outputs from diverse systems. - Practical sessions: comparative prompt writing workshops across platforms.	- Design effective prompt solutions for diverse AI systems [Evaluar]. - Develop cross-platform workflows that integrate different AI tools [Usar]. - Implement iterative refinement strategies to optimize AI outputs [Usar].
Readings : [Whi+23], [Ins24a], [Mic24], [Dee24]	

Unit 3: AI Applications Across Disciplines and Platforms (16 hours)	
Competences Expected: 2,6,AG-C09,AG-C12	
Topics	Learning Outcomes
- AI for research: literature review, data analysis using multiple AI assistants. - AI for writing: academic papers, reports with cross-platform verification. - AI for problem-solving: scientific calculations, engineering design with specialized tools. - AI for creativity: brainstorming, concept development across cultural contexts. - Platform-specific strengths: when to use Western vs. Chinese AI tools. - Discipline-specific workshops: tailored applications using diverse AI ecosystems. - Case studies: real-world applications in scientific research across regions.	- Apply AI fundamentals to solve discipline-specific problems [Usar]. - Design solutions that integrate multiple intelligent systems for complex tasks [Evaluar]. - Develop practical applications using Western and Chinese AI tools [Usar]. - Implement workflows that leverage specific strengths of each platform [Evaluar].
Readings : [Mol23], [Goo24], [Clo24], [Ope24]	

Unit 4: Global AI Ethics and Responsible Usage (12 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
- Understanding AI biases: Western and Eastern cultural perspectives. - Ethical frameworks: comparing EU AI Act, Chinese regulations, and global standards. - Academic integrity: proper citation and AI usage in coursework across platforms. - Privacy and data security: regional differences in AI data handling. - Cultural sensitivity: navigating AI outputs in global contexts. - Environmental impact: sustainability considerations of different AI systems. - Developing personal guidelines for ethical AI usage in international settings. - Case studies: ethical dilemmas in AI deployment across regions.	- Demonstrate professional responsibility in using AI technologies [Evaluar]. - Apply professional ethical principles in global AI contexts [Usar]. - Develop ethical guidelines to mitigate biases in AI systems [Evaluar]. - Evaluate ethical implications of AI deployment across different cultures [Usar].
Readings : [UNE23], [Uni24], [BG+21], [Cyb24], [Lab23]	

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

- [BG+21] Emily M. Bender, Timnit Gebru, et al. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” In: *FACCT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (2021). Foundational paper on AI ethics with ongoing relevance.
- [Lab23] Tencent AI Lab. *Responsible AI Practices in Chinese Tech Industry*. Industry perspective on AI ethics from leading Chinese tech company. 2023. URL: <https://ai.tencent.com/>.
- [Mol23] Ethan Mollick. “ChatGPT and How AI Disrupts Industries”. In: *Harvard Business Review* (2023). Analysis of AI’s practical impact across global sectors.
- [UNE23] UNESCO. *AI and Education: Guidance for Policy-Makers*. Latest guidance on AI literacy and ethical implementation in education. 2023. URL: <https://unesdoc.unesco.org/>.
- [Whi+23] Jules White et al. *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. Peer-reviewed research on prompt engineering patterns. 2023. URL: <https://arxiv.org/abs/2302.11382>.
- [Clo24] Alibaba Cloud. *AI Ethics and Best Practices in Chinese Context*. Chinese perspective on AI ethics and implementation guidelines. 2024. URL: <https://www.alibabacloud.com/>.
- [Cyb24] Chinese Academy of Cyberspace Studies. *AI Governance and Ethics in China*. Official Chinese perspective on AI governance and ethical standards. 2024. URL: <http://www.cac.gov.cn/>.
- [Dee24] DeepSeek. *DeepSeek Model Documentation and Best Practices*. Official documentation for DeepSeek AI models and usage guidelines. 2024. URL: <https://platform.deepseek.com/>.
- [Goo24] Google. *Introduction to Responsible AI*. Google’s updated framework for responsible AI development and use. 2024. URL: <https://cloud.google.com/learn/responsible-ai>.
- [Ins24a] Prompt Engineering Institute. *Prompt Engineering Guide*. Continuously updated comprehensive prompt engineering resource. 2024. URL: <https://www.promptingguide.ai/>.
- [Ins24b] Tencent Research Institute. *AI Development in China: Current Status and Future Trends*. Comprehensive analysis of Chinese AI ecosystem development. 2024. URL: <https://tri.tencent.com/>.
- [Mic24] Microsoft. *Prompt Crafting for AI Systems*. Official Microsoft prompt engineering guidance updated for 2024. 2024. URL: <https://learn.microsoft.com/en-us/training/paths/prompt-engineering/>.
- [MM24] Ethan Mollick and Lilach Mollick. *Co-Intelligence: Living and Working with AI*. Practical guide to human-AI collaboration. Penguin Random House, 2024.
- [Ng24] Andrew Ng. *AI for Everyone*. Updated online course covering AI fundamentals for non-technical audiences. 2024. URL: <https://www.deeplearning.ai/courses/ai-for-everyone/>.
- [Ope24] OpenAI. *Best Practices for Prompt Engineering*. Updated official prompt engineering guidelines from OpenAI. 2024. URL: <https://platform.openai.com/docs/guides/prompt-engineering>.
- [Uni24] European Union. *AI Act: Regulatory Framework*. Comprehensive AI regulation framework effective 2024. 2024. URL: <https://digital-strategy.ec.europa.eu/en/policies/ai-act>.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

MA104FCCS. Advanced Differential and Integral Calculus (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA104FCCS. Advanced Differential and Integral Calculus
2.2 Semester	: 3 rd Semester
2.3 Credits	: 5
2.4 Horas	: 4 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	: MA102FCCS. Differential Calculus. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Advanced differential and integral calculus is fundamental for understanding multidimensional phenomena in engineering and computer science. This course delves into the concepts of vector calculus, including vector functions, multivariable calculus, multiple integrals, and fundamental theorems. These tools are essential for modeling complex systems, optimizing multivariable functions, and solving problems in computer graphics, computer vision, and physical simulations.

5. GOALS

- Understand and manipulate vector functions of one and several variables.
- Apply differential and integral calculus to problems in multiple dimensions.
- Use Green's, Gauss's, and Stokes' theorems in practical applications.
- Model and solve engineering problems using line and surface integrals.

6. COMPETENCES

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

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Vector Functions of a Single Real Variable (14 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Definition of vector-valued functions of a real variable. • Limits, continuity, and derivatives of vector functions. • Tangent, normal, and binormal vectors. • Curvature, torsion, and Frenet formulas. • Arc length and parametrization. • Osculating plane and moving frame.	• Calculate derivatives and integrals of vector functions. [Familiarizarse] • Determine tangent, normal, and binormal vectors for a curve. [Usar] • Compute curvature and torsion using Frenet formulas. [Evaluar] • Parametrize curves using arc length. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

Unit 2: Real-Valued Functions of a Vector Variable (18 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Functions of several variables: graphs and level curves. • Limits and continuity in multiple variables. • Partial and directional derivatives. • Gradient vector and differential. • Tangent plane and normal line. • Implicit function theorem. • Multivariable maxima and minima. • Taylor's theorem for several variables.	• Calculate partial and directional derivatives. [Familiarizarse] • Determine tangent planes and linear approximations. [Usar] • Optimize functions of several variables. [Evaluar] • Apply the implicit function theorem. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

Unit 3: Vector-Valued Functions of Vector Variables (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Transformations from $\mathbb{R}^n$ to $\mathbb{R}^m$. • Derivative and differential of transformations. • Jacobian matrix and Jacobian. • Inverse function theorem. • Implicit function theorem. • Transformations in polar, cylindrical, and spherical coordinates.	• Compute the Jacobian matrix of a transformation. [Familiarizarse] • Apply the inverse function theorem. [Usar] • Perform variable changes using coordinate transformations. [Evaluar] • Use the implicit function theorem in multivariable contexts. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

Unit 4: Multiple Integrals (18 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Double integrals over rectangles and general regions. • Iterated integrals and change of variables. • Double integrals in polar coordinates. • Triple integrals and applications. • Triple integrals in cylindrical and spherical coordinates. • Calculation of areas, volumes, and centers of mass.	• Evaluate double and triple integrals. [Familiarizarse] • Apply change of variables in multiple integrals. [Usar] • Calculate volumes and areas using multiple integrals. [Evaluar] • Use cylindrical and spherical coordinates in triple integrals. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

Unit 5: Line Integrals (10 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Line integrals of the first and second kind. • Vector fields and work. • Green's theorem. • Path independence. • Conservative fields and potential functions. • Fundamental theorems of calculus for line integrals.	• Calculate line integrals over curves. [Familiarizarse] • Apply Green's theorem in the plane. [Usar] • Determine if a vector field is conservative. [Evaluar] • Construct potential functions for conservative fields. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

Unit 6: Surface Integrals (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C08,AG-C11	
Topics	Learning Outcomes
• Parametric surfaces and surface area. • Surface integrals of the first kind (scalar fields). • Surface integrals of the second kind (vector fields). • Stokes' theorem. • Gauss's theorem (Divergence theorem). • Curl and divergence of vector fields.	• Calculate surface integrals. [Familiarizarse] • Apply Stokes' theorem to oriented surfaces. [Usar] • Use the divergence theorem in 3D problems. [Evaluar] • Compute curl and divergence of vector fields. [Evaluar]
Readings : [Apostol1997], [MarsdenTromba]	

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 2026-I

1. COURSE

ST251FCCS. Probability Calculation (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: ST251FCCS. Probability Calculation
2.2 Semester	: 3 rd 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	: MA102FCCS. Differential Calculus. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Probability and statistics are fundamental in computer science for algorithm analysis, system modeling, decision-making under uncertainty, and data analysis. This course integrates probability theory with statistical methods, focusing on computational applications such as algorithm analysis, machine learning, system modeling, and data science.

5. GOALS

- Understand the fundamentals of probability and statistical inference.
- Apply probability distributions and statistical methods to solve computing problems.
- Develop skills for modeling stochastic systems and performing data inference.

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

Unit 1: Probability Fundamentals (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Probability spaces and axioms • Event operations: union, intersection, complement • Counting techniques: permutations and combinations • Conditional probability and independence • Bayes' theorem and applications	• Define sample spaces and apply probability axioms. [Familiarizarse] • Use counting techniques to calculate probabilities. [Usar] • Apply Bayes' theorem in classification problems. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 2: Discrete Random Variables (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Discrete random variables • Probability mass function (PMF) and cumulative distribution function (CDF) • Expectation, variance, and moments • Distributions: Bernoulli, Binomial, Geometric, Poisson • Applications in computational system modeling	• Define and characterize discrete random variables. [Familiarizarse] • Calculate expectation and variance for different distributions. [Usar] • Apply discrete distributions in network and system modeling. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 3: Continuous Random Variables (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Continuous random variables • Probability density function (PDF) and CDF • Transformations of random variables • Distributions: Uniform, Exponential, Normal • Applications in system simulation and queuing theory	• Differentiate between discrete and continuous variables. [Familiarizarse] • Calculate probabilities using density functions. [Usar] • Model service times and arrival times using continuous distributions. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 4: Multivariate Distributions and Dependence (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Joint and marginal distributions • Conditional distributions • Covariance and correlation • Independence of random variables • Applications in multidimensional data analysis	• Calculate joint and marginal distributions. [Familiarizarse] • Measure dependence using covariance and correlation. [Usar] • Analyze relationships between variables in datasets. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 5: Limit Theorems and Approximations (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Law of Large Numbers • Central Limit Theorem • Moments and generating functions • Probabilistic inequalities (Chernoff, Markov, Chebyshev) • Applications in big data and algorithm analysis	• State and interpret limit theorems. [Familiarizarse] • Apply CLT in distribution approximations. [Usar] • Use inequalities in algorithm bound analysis. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 6: Bayesian Inference (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Bayesian updating with discrete/continuous priors • Maximum A Posteriori (MAP) estimation • Bayesian credible intervals • Conjugate priors • Applications in Naive Bayes classifiers and ML	• Differentiate between frequentist and Bayesian approaches. [Familiarizarse] • Perform Bayesian belief updating. [Usar] • Implement MAP estimation in machine learning problems. [Evaluar]
Readings : [Ros14], [Dev16]	

Unit 7: Statistical Inference (6 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Parameter estimation: MLE (Maximum Likelihood Estimation) • Hypothesis testing: z-test, t-test, chi-square test • Confidence intervals • Bootstrapping and resampling methods • Applications in model validation and data science	• Estimate parameters using maximum likelihood methods. [Familiarizarse] • Perform hypothesis testing to validate assumptions. [Usar] • Construct confidence intervals and apply bootstrapping. [Evaluar]
Readings : [Dev16]	

Unit 8: Regression and Advanced Applications (6 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
• Simple linear regression • Least squares fitting • Poisson processes • Monte Carlo simulation • Introduction to Markov chains • Applications in prediction and system modeling	• Implement linear regression models. [Familiarizarse] • Use Monte Carlo simulation to solve complex problems. [Usar] • Apply stochastic processes in real system modeling. [Evaluar]
Readings : [Dev16]	

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

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

ID103. Technical and professional English III (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: ID103. Technical and professional English III
2.2 Semester	: 3 rd Semester
2.3 Credits	: 2
2.4 Horas	: 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. (2 nd 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 relation of ideas .

6. COMPETENCES

3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04) Communication: Communicates effectively in complex computing activities. (Usage)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Usage)

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)

7. TOPICS

Unit 1: Getting to know you! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present, Past, and Future Times. • Interrogative sentences with Wh-. • Words with more than one meaning. • Parts of the sentence • Expressions for free time	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: The way we live! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Simple present tense. • Present Continuous Time. • Collocations. • Vocabulary of the countries of the world. • Expressions of anger. • Connectors.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: It all went wrong! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Past simple tense. • Continuous past tense. • Irregular Verbs. • Time expressions. • Connectors of time.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Let's go shopping! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Expressions of Indefinite Quantity. • Affirmative sentences, Negatives and Questions. • Use of Articles. • Product prices. • Filling of formats and surveys • Expressions for shopping	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: What do you want to do? (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Verbal Patterns I. • Future Intentions. • Verbs of Perception. • Vocabulary of feelings. • Expressions of Plans and Ambitions.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: The best in the world! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• What's it like?. • Adjectives • Comparative and superlative. • Synonyms and antonyms. • Indications of direction . • Readings.	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Fame! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present Perfect and Simple Past • Expressions for, ever, since • Adverbs • Expressions that come in pairs • Short answers • Celebrities	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

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

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

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

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

Chapter 4

Fourth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS210-CS2023. Algorithms and Data Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Course	:	CS210-CS2023. Algorithms and Data Structures
2.2 Semester	:	4 th 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	:	CS113-CS2023. Objects oriented programming II. (3 rd 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. (Assessment)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Graphs (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Graph Concept • Directed Graphs and Non-directed Graphs. • Using Graphs. • Measurement of efficiency ,in time and space. • Adjacency matrices. • Tag adjacent matrices. • Adjacency Lists. • Implementation of graphs using adjacency matrices. • Graph Implementation using adjacency lists • Insertion, search and deletion of nodes and edges. • Graph search algorithms.	• Acquire Dexterity to Perform Correct Implementation. [Usar] • Develop knowledge to decide when it is better to use one implementation technique than another. [Usar]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 2: Scatter Matrices (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Initial concepts. • Dense Matrices • Measurement of Efficiency in Time and Space • Static scatter vs. dynamic matrix creation. • Insert, search, and delete methods.	• Understand the use and implementation of scatter matrices.[Evaluar]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 3: Balanced Trees (16 hours)	
Competences Expected:	
Topics	Learning Outcomes
• AVL Trees. • Measurement of Efficiency. • Simple and Composite Rotations • Insertion, deletion and search. • Trees B , B+ B* y Patricia.	• Understand the basic functions of these complex structures in order to acquire the capacity for their implementation. [Evaluar]
Readings : [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 2026-I

1. COURSE

CS211-CS2023. Theory of Computation (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS211-CS2023. Theory of Computation
2.2 Semester	: 4 th 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	: CS1D1-2-CS2023. Discrete Structures. (2 nd 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

- 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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C10) Investigation: Studies complex computing problems using information science methods. (Familiarity)

7. TOPICS

Unit 1: Computabilidad y complejidad básica de autómatas (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Máquinas de estado finito. • Expresiones regulares. • Problema de la parada. • Gramáticas libres de contexto. • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. • Máquinas de Turing, o un modelo formal equivalente de computación universal. • Máquinas de Turing no determinísticas. • Jerarquía de Chomsky. • La tesis de Church-Turing. • Computabilidad. • Teorema de Rice. • Ejemplos de funciones no computables. • Implicaciones de la no-computabilidad.	• Discute el concepto de máquina de estado finito [Evaluar] • Diseña una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar] • Genere una expresión regular para representar un lenguaje específico [Evaluar] • Explique porque el problema de la parada no tiene solución algorítmica [Evaluar] • Diseña una gramática libre de contexto para representar un lenguaje especificado [Evaluar] • Define las clases P y NP [Evaluar] • Explique el significado de NP-Complejidad [Evaluar] • Explica la tesis de Church-Turing y su importancia [Familiarizarse] • Explica el teorema de Rice y su importancia [Familiarizarse] • Da ejemplos de funciones no computables [Familiarizarse] • Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él [Familiarizarse]
Readings : [Mar10], [Lin11], [Sip12]	

Unit 2: Complejidad Computacional Avanzada (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Revisión de las clases P y NP; introducir espacio P y EXP. • Jerarquía polinomial. • NP completitud (Teorema de Cook). • Problemas NP completos clásicos. • Técnicas de reducción.	• Define la clase P-Space y su relación con la clase EXP [Evaluar] • Define la clase P-Space y su relación con la clase EXP [Evaluar] • Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar] • Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar] • Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar]
Readings : [Mar10], [Lin11], [Sip12], [HU13]	

Unit 3: Teoría y Computabilidad Avanzada de Autómatas (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Conjuntos y Lenguajes: – Lenguajes Regulares. – Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs) – Autómata finito no determinístico (Nondeterministic Finite Automata NFAs) – Equivalencia de DFAs y NFAs. – Revisión de expresiones regulares; su equivalencia con autómatas finitos. – Propiedades de cierre. – Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos. • Lenguajes libres de contexto: – Autómatas de pila (Push-down automata (PDAs) – Relación entre PDA y gramáticas libres de contexto. – Propiedades de los lenguajes libres de contexto.	• Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar] • Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar]
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 2026-I

1. COURSE

CS221-CS2023. Computer Systems Architecture (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS221-CS2023. Computer Systems Architecture
2.2 Semester	: 4 th 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	: CS1D1. Discrete Structures. (2 nd 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

- 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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Lógica digital y sistemas digitales (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Revisión e historia de la Arquitectura de Computadores. • Lógica combinacional vs. secuencial/Arreglos de puertas de campo programables como bloque fundamental de construcción lógico combinacional-secuencial. • Múltiples representaciones / Capas de interpretación (El hardware es solo otra capa) • Herramientas de diseño asistidas por computadora que procesan hardware y representaciones arquitecturales. • Registrar transferencia notación / Hardware language descriptivo (Verilog/VHDL) • Restricción física (Retrasos de Entrada, fan-in, fan-out, energía/poder)	
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 2: Organización de la Máquina a Nivel Ensamblador (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • Paquetes de instrucciones y tipos (manipulación de información, control, I/O) • Assembler / Programación en Lenguaje de Máquina. • Formato de instrucciones. • Modos de direccionamiento. • Llamada a subrutinas y mecanismos de retorno. • I/O e Interrupciones. • Montículo (Heap) vs. Estático vs. Pila vs. Segmentos de código.	• Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse] • Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse] • Describir el paralelismo a nivel de instrucciones y sus peligros, y cómo es esto tratado en pipelines de proceso típicos [Familiarizarse] • Resumir cómo se representan las instrucciones, tanto a nivel de máquina bajo el contexto de un ensamblador simbólico [Familiarizarse] • Demostrar cómo se mapean los patrones de lenguajes de alto nivel en notaciones en lenguaje ensamblador o en código máquina [Usar] • Explicar los diferentes formatos de instrucciones, así como el direccionamiento por instrucción, y comparar formatos de tamaño fijo y variable [Usar] • Explicar como las llamadas a subrutinas son manejadas a nivel de ensamblador [Usar] • Explicar los conceptos básicos de interrupciones y operaciones de entrada y salida (I/O) [Familiarizarse] • Escribir segmentos de programa simples en lenguaje ensamblador [Usar] • Ilustrar cómo los bloques constructores fundamentales en lenguajes de alto nivel son implementados a nivel de lenguaje máquina [Usar]
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 3: Organización funcional (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Implementación de rutas de datos simples, incluyendo la canalización de instrucciones, detección de riesgos y la resolución. • Control de unidades: Realización Cableada vs Realización Microprogramada. • Instrucción (Pipelining) • Introducción al paralelismo al nivel de instrucción (PNI)	• Comparar implementaciones alternativas de ruta de datos [Evaluar] • Discutir el concepto de puntos de control y la generación de señales de control usando implementaciones a nivel de circuito o microprogramadas [Familiarizarse] • Explicar el paralelismo a nivel de instrucciones básicas usando pipelining y los mayores riesgos que pueden ocurrir [Usar] • Diseñar e implementar un procesador completo, incluyendo ruta de datos y control [Usar] • Calcular la cantidad promedio de ciclos por instrucción de una implementación con procesador y sistema de memoria determinados [Evaluar]
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 4: Organización y Arquitectura del Sistema de Memoria (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Sistemas de Almacenamiento y su Tecnología. • Jerarquía de Memoria: importancia de la localización temporal y espacial. • Organización y Operaciones de la Memoria Principal. • Latencia, ciclos de tiempo, ancho de banda e intercalación. • Memorias caché (Mapeo de direcciones, Tamaño de bloques, Reemplazo y Políticas de almacenamiento) • Multiprocesador coherencia cache / Usando el sistema de memoria para las operaciones de sincronización de memoria / atómica inter-core. • Memoria virtual (tabla de página, TLB) • Manejo de Errores y confiabilidad. • Error de codificación, compresión de datos y la integridad de datos.	• Identifique las principales tecnologías de memoria (Por ejemplo: SRAM, DRAM, Flash, Disco Magnético) y su relación costo beneficio [Familiarizarse] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Usar] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Usar] • 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]
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 5: Interfaz y comunicación (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Fundamentos de I/O: Handshaking, Bbuffering, I/O programadas, interrupciones dirigidas de I/O. - Interrumpir estructuras: interrumpir reconocimiento, vectorizado y priorizado. - Almacenamiento externo, organización física y discos. - Buses: Protocolos de bus, arbitraje, acceso directo a memoria (DMA). - Introducción a Redes: comunicación de redes como otra capa de acceso remoto. - Soporte Multimedia. - Arquitecturas RAID.	- Explicar como las interrupciones son aplicadas para implementar control de entrada-salida y transferencia de datos [Familiarizarse] - Identificar diversos tipos de buses en un sistema computacional [Familiarizarse] - Describir el acceso a datos desde una unidad de disco magnético [Usar] - Comparar organizaciones de red conocidas como organizaciones en bus/Ethernet, en anillo y organizaciones conmutadas versus ruteadas [Evaluar] - 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] - Describir las ventajas y limitaciones de las arquitecturas RAID [Familiarizarse]
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 6: Multiprocesamiento y arquitecturas alternativas (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Ley potencial. - Ejemplos de juego de instrucciones y arquitecturas SIMD y MIMD. - Redes de interconexión (Hypercube, Shuffle-exchange, Mesh, Crossbar) - Sistemas de memoria de multiprocesador compartido y consistencia de memoria. - Coherencia de cache multiprocesador.	- Discutir el concepto de procesamiento paralelo mas allá del clásico modelo de von Neumann [Evaluar] - Describir diferentes arquitecturas paralelas como SIMD y MIMD [Familiarizarse] - Explicar el concepto de redes de interconexión y mostrar diferentes enfoques [Usar] - Discutir los principales cuidados en los sistemas de multiprocesamiento presentes con respecto a la gestión de memoria y describir como son tratados [Familiarizarse] - 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]
Readings : [Patterson2014], [Stallings2014], [Par05]	

Unit 7: Mejoras de rendimiento (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Arquitectura superescalar. • Predicción de ramificación, Ejecución especulativa, Ejecución fuera de orden. • Prefetching. • Procesadores vectoriales y GPU's • Soporte de hardware para multiprocesamiento. • Escalabilidad. • Arquitecturas alternativas, como VLIW / EPIC y aceleradores y otros tipos de procesadores de propósito especial.	• Describir las arquitecturas superescalares y sus ventajas [Familiarizarse] • Explicar el concepto de predicción de bifurcaciones y su utilidad [Usar] • Caracterizar los costos y beneficios de la precarga prefetching [Evaluar] • Explicar la ejecución especulativa e identifique las condiciones que la justifican [Evaluar] • 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] • Describir la importancia de la escalabilidad en el rendimiento [Evaluar]
Readings : [Patterson2014], [Stallings2014], [Par05]	

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

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers*. New York: Oxford Univ. Press, 2005.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS271-CS2023. Data Management (Mandatory)

2. GENERAL INFORMATION

- 2.1 Course** : CS271-CS2023. Data Management
- 2.2 Semester** : 4th 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-CS2023. Objects oriented programming I. (2nd Sem)
- 2.8 Prerequisites** :
 - CS1D1-2-CS2023. Discrete Structures. (2nd 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

- 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)

- 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)

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

Unit 1: Sistemas de Bases de Datos (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Enfoque y Evolución de Sistemas de Bases de Datos. • Componentes del Sistema de Bases de Datos. • Diseño de las funciones principales de un DBMS. • Arquitectura de base de datos e independencia de datos. • Uso de un lenguaje de consulta declarativa. • Sistemas de apoyo a contenido estructurado y / o corriente. • Enfoques para la gestión de grandes volúmenes de datos (por ejemplo, sistemas de bases de datos NoSQL, uso de MapReduce).	• Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Usar] • Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Usar] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Usar] • Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Usar] • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Usar] • Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar] • Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Usar] • Describe los enfoques principales para almacenar y procesar largos volúmenes de datos [Usar]
Readings : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unit 2: Modelado de datos (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Modelado de datos • Modelos conceptuales (e.g., entidad-relación, diagramas UML) • Modelos de hoja de cálculo • Modelos Relacionales. • Modelos orientados a objetos. • Modelos de datos semi-estructurados (expresados usando DTD o XML Schema, por ejemplo)	• Compare y contrasta modelos apropiados de datos, incluyendo estructuras sus estructuras internas, para diversos tipos de datos [Usar] • Describe los conceptos en notación de modelos (ejm. Diagramas Entidad-Relación o UML) y cómo deben de ser usados [Usar] • Define la terminología fundamental a ser usada en un modelo relacional de datos [Usar] • Describe los principios básicos del modelo relacional de datos [Usar] • Aplica los conceptos de modelado y la notación de un modelo relacional de datos [Usar] • Describe los principios básicos del modelo relacional de datos [Usar] • Describe los principios básicos del modelo relacional de datos [Usar] • Da una semi estructura equivalente (ejm. en DTD o Esquema XML) para un esquema relacional dado [Usar]
Readings : [SW04], [EN04], [KS02]	

Unit 3: Indexación (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• El impacto de índices en el rendimiento de consultas. • La estructura basica de un indice. • Mantener un buffer de datos en memoria. • Creando índices con SQL. • Indexando texto. • Indexando la web (e.g., web crawling)	• Generar un archivo índice para una colección de recursos [Usar] • Explicar la función de un índice invertido en la localización de un documento en una colección [Usar] • Explicar cómo rechazar y detener palabras que afectan a la indexación [Usar] • Identificar los índices adecuados para determinado el esquema relacional y el conjunto de consultas [Usar] • Estimar el tiempo para recuperar información, cuando son usados los índices comparado con cuando no son usados [Usar] • Describir los desafíos claves en el rastreo web, por ejemplo, la detección de documentos duplicados, la determinación de la frontera de rastreo [Usar]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 4: Bases de Datos Relacionales (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Mapeo de esquemas conceptuales a esquemas relacionales. • Entidad y integridad referencial. • Algebra relacional y calculo relacional. • Diseño de bases de datos relacionales. • Dependencia funcional. • Descomposición de un esquema. • Llaves candidatas, SuperLlaves y cierre de un conjunto de atributos. • Formas Normales (BCNF) • Dependencias multi-valoradas (4NF) • Uniendo dependencias (PJNF, 5NF) • Teoría de la representación.	• Prepara un esquema relacional de un modelo conceptual desarrollado usando el modelo entidad-relación [Usar] • Explica y demuestra los conceptos de restricciones de integridad de la entidad e integridad referencial (incluyendo la definición del concepto de clave foránea) [Usar] • 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] • Escribe consultas en cálculo relacional de tuplas [Usar] • Escribe consultas en cálculo relacional de tuplas [Usar] • Determina la dependencia funcional entre dos o más atributos que son subconjunto de una relación [Usar] • Conecta restricciones expresadas como clave primaria y foránea, con dependencias funcionales [Usar] • Calcula la cerradura de un conjunto de atributos dado dependencias funcionales [Usar] • Determina si un conjunto de atributos forma una superclave y/o una clave candidata de una relación dada dependencias funcionales [Usar] • Evalua una descomposición propuesta, a fin de determinar si tiene una unión sin pérdidas o preservación de dependencias [Usar] • Describe las propiedades de la FNBC, FNUP (forma normal unión de proyecto), 5FN [Usar] • Explica el impacto de la normalización en la eficacia de las operaciones de una base de datos especialmente en la optimización de consultas [Usar] • Describe que es una dependencia de multi valor y cual es el tipo de restricciones que especifica [Usar]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 5: Lenguajes de Consulta (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Visión general de lenguajes de base de datos. • SQL (definición de datos, formulación de consultas, sublenguaje update, restricciones, integridad) • Selecciones • Proyecciones • Select-project-join • Agregaciones y agrupaciones. • Subconsultas. • Entornos QBE de cuarta generación. • Diferentes maneras de invocar las consultas no procedimentales en lenguajes convencionales. • Introducción a otros lenguajes importantes de consulta (por ejemplo, XPATH, SPARQL) • Procedimientos almacenados.	• Crear un esquema relacional de bases de datos en SQL que incorpora restricciones clave y restricciones de integridad de entidad e integridad referencial [Usar] • Usar SQL para crear tablas y devuelve (SELECT) la información de una base de datos [Usar] • Evaluar un conjunto de estrategias de procesamiento de consultas y selecciona la estrategia óptima [Usar] • Crear una consulta no-procedimental al llenar plantillas de relaciones para construir un ejemplo del resultado de una consulta requerida [Usar] • Adicionar consultas orientadas a objetos en un lenguaje stand-alone como C++ o Java (ejm. SELECT ColMethod() FROM Objeto) [Usar] • Escribir un procedimiento almacenado que trata con parámetros y con algo de flujo de control de tal forma que tenga funcionalidad [Usar]
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 2026-I

1. COURSE

CS401. Research Methodology (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS401. Research Methodology
2.2 Semester	: 4 th 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	: CS100-CS2023. Introduction to Computer Science. (2 nd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to teach students how to conduct scientific research in the field of computer science. The course instructors will determine a specific study area for each student and provide relevant bibliography for analysis. Based on this and additional bibliographic sources (researched by the student), the student should be able to construct a survey-type article on the assigned topic.

5. GOALS

- To teach students how to initiate scientific research in the field of computer science.
- To familiarize students with the main sources for obtaining relevant bibliography for research work in computer science: Researchindex, IEEE-CS¹, ACM².
- To enable students to analyze existing proposals on a specific topic and relate them coherently in a literature review.
- To enable students to write technical documents in computer science using L^AT_EX.
- To enable students to reproduce existing results in a specific topic through experimentation.
- The deliverables for this course are:

Partial Progress: Mastery of the article topic and preliminary bibliography in L^AT_EXarticle format.

Final: Understanding of the survey-type article, completed document containing, optionally, experimental results of the studied technique(s).

6. COMPETENCES

- 1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. (Usage)
- AG-C08) Problem Analysis:** Identifies, formulates, and analyzes complex computing problems. (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)

¹<http://www.computer.org>

²<http://www.acm.org>

AG-C10) Investigation: Studies complex computing problems using information science methods. (Assessment)

3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04) Communication: Communicates effectively in complex computing activities. (Usage)

4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

7. TOPICS

Unit 1: Scientific Initiation in Computer Science (48 hours)	
Competences Expected: 1,3,4,7,AG-C02,AG-C04,AG-C10	
Topics	Learning Outcomes
• Bibliographic search in computer science. • Writing technical articles in computer science.	• Learn how to conduct proper research in the field of computer science[Usar] • Know the appropriate bibliographic sources for this field[Usar] • Know how to write a document according to the characteristics required by conferences in this field[Usar]
Readings : [IEE08], [ACM08], [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

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

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

MA106FCCS. Numerical Methods (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA106FCCS. Numerical Methods
2.2 Semester	: 4 th 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. Integral Calculus. (2 nd 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

Unit 1: Introduction to Numerical Methods (4 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Computer representation of numbers. • Round-off and truncation errors. • Error propagation. • Analysis of stability and convergence.	• Explain how numbers are represented in a computer and the limitations of this representation. [Familiarizarse] • Differentiate between round-off and truncation errors. [Usar] • Analyze how errors propagate in numerical calculations. [Evaluar]
Readings : [CC15], [BF10]	

Unit 2: Solving Nonlinear Equations (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Bisection method. • Newton-Raphson method. • Secant method.	• Apply the bisection method to find roots of equations. [Familiarizarse] • Use the Newton-Raphson method to approximate solutions. [Usar] • Implement the secant method to solve nonlinear equations. [Evaluar]
Readings : [CC15], [BF10]	

Unit 3: Interpolation and Polynomial Approximation (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Lagrange polynomial interpolation. • Newton interpolation. • Splines.	• Construct Lagrange interpolating polynomials. [Familiarizarse] • Apply Newton interpolation. [Usar] • Use splines to approximate functions. [Evaluar]
Readings : [CC15], [BF10]	

Unit 4: Numerical Integration (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Trapezoidal rule. • Simpson's rule. • Gaussian quadrature.	• Apply the trapezoidal rule to approximate integrals. [Familiarizarse] • Use Simpson's rule to calculate integrals numerically. [Usar] • Apply Gaussian quadrature for numerical integration. [Evaluar]
Readings : [CC15], [BF10]	

Unit 5: Numerical Solution of Ordinary Differential Equations (8 hours)	
Competences Expected: 1,6,AG-C07	
Topics	Learning Outcomes
• Euler's method. • Runge-Kutta methods.	• Apply Euler's method to approximate solutions of ODEs. [Familiarizarse] • Implement Runge-Kutta methods to solve ODEs numerically. [Usar]
Readings : [CC15], [BF10]	

Unit 6: Applications in Computing (12 hours)	
Competences Expected: 1,6,AG-C07,AG-C12	
Topics	Learning Outcomes
• Simulation of physical systems. • Scientific modeling. • Machine learning (e.g., model optimization).	• Use numerical methods to simulate physical systems. [Familiarizarse] • Apply numerical methods in scientific modeling. [Usar] • Implement numerical methods in machine learning algorithms. [Evaluar]
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 2026-I

1. COURSE

ID104. Technical and professional English IV (Elective)

2. GENERAL INFORMATION

2.1 Course	: ID104. Technical and professional English IV
2.2 Semester	: 4 th Semester
2.3 Credits	: 2
2.4 Horas	: 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	: ID103. Technical and professional English III. (3 rd 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

3) Communicate effectively in a variety of professional contexts.. (Assessment)

AG-C04) Communication: Communicates effectively in complex computing activities. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Usage)

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)

7. TOPICS

Unit 1: Do and don't! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Mode Auxiliaries should, must and have got to. • Affirmative, negative and interrogative sentences with modals. • Terms for formal letters. • Parts of short answers. • Expressions for occupations.	• 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
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Going places! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present and Future Present Time with Will • First conditional • Collocations • Vocabulary of prepositions of place and time • Expressions of connection of ideas	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: Scared to death! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Infinitive and gerund verb patterns • What + Infinitive • Something + infinitive • Expressions of feelings • Exclamations of surprise	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Things that changed the world! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Passive Voice Â Â Â Â Â Â • Affirmative Prayers, Negatives and Questions Â Â Â Â Â Â • Use of participles, verbs and nouns that go together Â Â Â Â Â Â • Signals. Signs and notes Â Â Â Â Â Â • Summaries Â Â Â Â Â Â • Expressions to indicate prohibition	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Dreams and reality! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Second Conditional Â Â Â Â Â Â • Auxiliar of mode "might" Â Â Â Â Â Â • Phrase Verbs Â Â Â Â Â Â • Social expressions vocabulary Â Â Â Â Â Â • Adverbs Â Â Â Â Â Â • Expressions to give advice	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Making a living! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Present Perfect Continuous Â Â Â Â Â Â • Present Continuous Â Â Â Â Â Â • Occupations Â Â Â Â Â Â • Word formation Â Â Â Â Â Â • Adverbs Â Â Â Â Â Â • Expressions of use on the phone	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: All you need is love! (0 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Past Perfect and Past Simple • Report Expressions • Expressions of words in different contexts • Short and formal farewells • Love Stories	• 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.
Readings : [SJ02], [Cam06], [Mac99]	

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

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

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

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

Chapter 5

Fifth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS212-CS2023. Analysis and Design of Algorithms (Mandatory)

2. GENERAL INFORMATION

- | | | |
|-----------------------------------|---|--|
| 2.1 Course | : | CS212-CS2023. Analysis and Design of Algorithms |
| 2.2 Semester | : | 5 th 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 | : | <ul style="list-style-type: none">• CS210-CS2023. Algorithms and Data Structures. (4th Sem)• CS211-CS2023. Theory of Computation. (4th 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)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (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. (Familiarity)

7. TOPICS

Unit 1: Análisis Básico (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Diferencias entre el mejor, el esperado y el peor caso de un algoritmo. - Análisis asintótico de complejidad de cotas superior y esperada. - Clases de complejidad como constante, logarítmica, lineal, cuadrática y exponencial. - Asymptotic Notation - Análisis de algoritmos iterativos y recursivos. - Inductive proofs and correctness of algorithms - Algunas versiones del Teorema Maestro.	- Explique a que se refiere con “mejor”, “esperado” y “peor” caso de comportamiento de un algoritmo [Evaluar] - Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Evaluar] - Lista y contraste de clases estándares de complejidad [Evaluar] - Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Evaluar] - Analyze worst-case running times of algorithms using asymptotic analysis [Evaluar] - Usar relaciones recurrentes para determinar el tiempo de complejidad de algoritmos recursivamente definidos [Evaluar] - Resuelve relaciones de recurrencia básicas, por ejemplo. usando alguna forma del Teorema Maestro [Evaluar] - Argue the correctness of algorithms using inductive proofs [Evaluar]
Readings : [KT05], [DPV06], [Cor+09], [SF13], [Knu97]	

Unit 2: Estrategias Algorítmicas (30 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Algoritmos de fuerza bruta. - Algoritmos voraces. - Divide y vencerás. - Programación Dinámica.	- 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] - Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar] - Utiliza un enfoque voraz para resolver un problema específico y determina si la regla escogida lo guía a una solución óptima [Evaluar] - Usa programación dinámica para resolver un problema determinado [Evaluar] - Determina el enfoque algorítmico adecuado para un problema [Evaluar]
Readings : [KT05], [DPV06], [Cor+09], [Als99]	

Unit 3: Algoritmos y Estructuras de Datos fundamentales (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Grafos y algoritmos en grafos: – Algoritmos de la ruta más corta (algoritmos de Dijkstra y Floyd) – Árbol de expansión mínima (algoritmos de Prim y Kruskal) • Cache oblivious algorithms • Number theory and cryptography	• 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] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar] • 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] • Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Evaluar]
Readings : [KT05], [DPV06], [Cor+09], [SW11], [GT09]	

Unit 4: Computabilidad y complejidad básica de autómatas (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Completos. • Reductions	• Define las clases P y NP [Familiarizarse] • Explique el significado de NP-Complejidad [Familiarizarse]
Readings : [KT05], [DPV06], [Cor+09]	

Unit 5: Estructuras de Datos Avanzadas y Análisis de Algoritmos (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Grafos (ej. Ordenamiento Topológico, encontrando componentes fuertemente conectados) • Algoritmos aleatorios. • Análisis amortizado. • Análisis Probabilístico. • Approximation Algorithms • Linear Programming	• Entender el mapeamiento de problemas del mundo real a soluciones algorítmicas (ejemplo, problemas de grafos, programas lineales, etc) [Familiarizarse] • Seleccionar y aplicar técnicas avanzadas de análisis (ejemplo, amortizado, probabilístico, etc) para algoritmos [Usar]
Readings : [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 2026-I

1. COURSE

CS261-CS2023. Artificial Intelligence (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS261-CS2023. Artificial Intelligence
2.2 Semester	: 5 th 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	: ST251FCCS. Probability Calculation. (3 rd 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

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)

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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Cuestiones fundamentales (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? – El Test de Turing – Razonamiento Racional versus No Racional • Características del Problema: – Observable completamente versus observable parcialmente – Individual versus multi-agente – Determinístico versus estocástico – Estático versus dinámico – Discreto versus continuo • Naturaleza de agentes: – Autónomo versus semi-autónomo – Reflexivo, basado en objetivos, y basado en utilidad – La importancia en percepción e interacciones con el entorno • Cuestiones filosóficas y éticas.	• Describir el test de Turing y el experimento pensado "cuarto chino" (<i>Chinese Room</i>) [Usar] • Determinando las características de un problema dado que la Inteligencia Artificial deberían resolver [Usar]
Readings : [De 06], [Pon+14]	

Unit 2: Agentes (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Definición de Agentes • Arquitectura de agentes (Ej. reactivo, en capa, cognitivo) • Teoría de agentes • Racionalidad, teoría de juegos: – Agentes de decisión teórica – Procesos de decisión de Markov (MDP) • Agentes de Software, asistentes personales, y acceso a información: – Agentes colaborativos – Agentes de recolección de información – Agentes creíbles (carácter sintético, modelamiento de emociones en agentes) • Agentes de aprendizaje • Sistemas Multi-agente – Agentes Colaborativos – Equipos de Agentes – Agentes Competitivos (ej., subastas, votaciones) – Sistemas de enjambre y modelos biológicamente inspirados	• Lista las características que definen un agente inteligente [Usar] • Describe y contrasta las arquitecturas de agente estándares [Usar] • Describe las aplicaciones de teoría de agentes para dominios como agentes de software, asistentes personales, y agentes creíbles [Usar] • Describe los paradigmas primarios usados por agentes de aprendizaje [Usar] • Demuestra mediante ejemplos adecuados como los sistemas multi-agente soportan interacción entre agentes [Usar]
Readings : [Nil01], [RN03], [Pon+14]	

Unit 3: Estrategias de búsquedas básicas (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Espacios de Problemas (estados, metas y operadores), solución de problemas mediante búsqueda. • Factored representation (factoring state hacia variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heurísticas y búsqueda informada (hill-climbing, generic best-first, A*) • El espacio y el tiempo de la eficiencia de búsqueda. • Dos jugadores juegos (introducción a la búsqueda minimax). • Satisfacción de restricciones (backtracking y métodos de búsqueda local).	• 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] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Usar] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Usar] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Usar]
Readings : [Nil01], [Pon+14]	

Unit 4: Búsqueda Avanzada (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Búsqueda estocástica: – Simulated annealing – Algoritmos genéticos – Búsqueda de árbol Monte-Carlo • Construcción de árboles de búsqueda, espacio de búsqueda dinámico, explosión combinatoria del espacio de búsqueda. • Implementación de búsqueda A*, búsqueda en haz. • Búsqueda Minimax, poda alfa-beta. • Búsqueda Expectimax (MDP-Solving) y los nodos de azar.	• Diseñar e implementar una solución a un problema con algoritmo genético [Usar] • Diseñar e implementar un esquema de recocido simulado (<i>simulated annealing</i>) para evitar mínimos locales en un problema [Usar] • Diseñar e implementar una búsqueda A* y búsqueda en haz (<i>beam search</i>) para solucionar un problema [Usar] • Aplicar búsqueda minimax con poda alfa-beta para simplificar el espacio de búsqueda en un juego con dos jugadores [Usar] • Comparar y contrastar los algoritmos genéticos con técnicas clásicas de búsqueda [Usar] • Comparar y contrastar la aplicabilidad de varias heurísticas de búsqueda, para un determinado problema [Usar]
Readings : [Gol89], [Nil01], [RN03], [Pon+14]	

Unit 5: Razonamiento Bajo Incertidumbre (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Revisión de Probabilidad Básica • Variables aleatorias y distribuciones de probabilidad: – Axiomas de probabilidad – Inferencia probabilística – Regla de Bayes • Independencia Condicional • Representaciones del conocimiento: – Redes bayesianas * Inferencia exacta y su complejidad * Métodos de Muestreo aleatorio (Monte Carlo) (p.e. Muestreo de Gibbs) – Redes Markov – Modelos de probabilidad relacional – Modelos ocultos de Markov	• Aplicar la regla de Bayes para determinar el cumplimiento de una hipótesis [Usar] • Explicar cómo al tener independencia condicional permite una gran eficiencia en sistemas probabilísticos [Usar] • Identificar ejemplos de representación de conocimiento para razonamiento bajo incertidumbre [Usar] • Indicar la complejidad de la inferencia exacta. Identificar métodos para inferencia aproximada [Usar]
Readings : [KF09], [RN03]	

Unit 6: Aprendizaje Automático Básico (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Definición y ejemplos de la extensa variedad de tareas de aprendizaje de máquina, incluida la clasificación. • Aprendizaje inductivo • Aprendizaje simple basado en estadísticas, como el clasificador ingenuo de Bayes, árboles de decisión. • El problema exceso de ajuste. • Medición clasificada con exactitud.	• Listar las diferencias entre los tres principales tipos de aprendizaje: supervisado, no supervisado y por refuerzo [Usar] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Usar] • Explicar la diferencia entre aprendizaje inductivo y deductivo [Usar] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Usar] • Aplicar un algoritmo de aprendizaje estadístico simple como el Clasificador Naive Bayesiano e un problema de clasificación y medirla precisión del clasificador [Usar]
Readings : [Mit98], [RN03], [Pon+14]	

Unit 7: Aprendizaje de máquina avanzado (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina - Aprendizaje general basado en estadística, estimación de parámetros (máxima probabilidad) - Programación lógica inductiva (<i>Inductive logic programming ILP</i>) - Aprendizaje supervisado - Aprendizaje basado en árboles de decisión - Aprendizaje basado en redes neuronales - Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>) - Aprendizaje y <i>clustering</i> no supervisado - EM - K-means - Mapas auto-organizados - Aprendizaje semi-supervisado. - Aprendizaje de modelos gráficos - Evaluación del desempeño (tal como cross-validation, area bajo la curva ROC) - Aplicación de algoritmos Machine Learning para Minería de datos.	- Explica las diferencias entre los tres estilos de aprendizaje: supervisado, por refuerzo y no supervisado [Usar] - Implementa algoritmos simples para el aprendizaje supervisado, aprendizaje por refuerzo, y aprendizaje no supervisado [Usar] - Determina cuál de los tres estilos de aprendizaje es el apropiado para el dominio de un problema en particular [Usar] - 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] - Evalúa el rendimiento de un sistema de aprendizaje simple en un conjunto de datos reales [Usar] - Describe el estado del arte en la teoría del aprendizaje, incluyendo sus logros y limitantes [Usar] - Explica el problema del sobreajuste, conjuntamente con técnicas para determinar y manejar el problema [Usar]
Readings : [RN03], [KF09], [Mur12]	

Unit 8: Procesamiento del Lenguaje Natural (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Gramaticas determinísticas y estocásticas • Algoritmos de parseo – Gramáticas libres de contexto (CFGs) y cuadros de parseo (e.g. Cocke-Younger-Kasami CYK) – CFGs probabilísticos y ponderados CYK • Representación del significado / Semántica – Representación de conocimiento basado en lógica – Roles semánticos – Representaciones temporales – Creencias, deseos e intenciones • Metodos basados en el corpus • N-gramas y Modelos ocultos de Markov (HMMs) • Suavizado y back-off • Ejemplos de uso: POS etiquetado y morfología • Recuperación de la información: – Modelo de espacio vectorial * TF & IDF – Precision y cobertura • Extracción de información • Traducción de lenguaje • Clasificación y categorización de texto: – Modelo de bolsa de palabras	• Define y contrasta gramáticas de tipo estocásticas y determinísticas, dando ejemplos y demostrando como adecuar cada una de ellas [Usar] • Simula, aplica, o implementa algoritmos clásicos y estocásticos para el parseo de un lenguaje natural [Usar] • Identifica los retos de la representación del significado [Usar] • Lista las ventajas de usar corpus estándares. Identifica ejemplos de corpus actuales para una variedad de tareas de PLN [Usar] • Identifica técnicas para la recuperación de la información, traducción de lenguajes, y clasificación de textos [Usar]
Readings : [Nil01], [RN03], [Pon+14]	

Unit 9: Visión y percepción por computador (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Visión Computacional – Adquisición de imágenes, representación, procesamiento y propiedades – Representación de formas, reconocimiento y segmentación de objetos – Análisis de movimiento • Modularidad en reconocimiento. • Enfoques de reconocimiento de patrones – Algoritmos de clasificación y medidas de calidad de la clasificación. – Técnicas estadísticas.	• Resumir la importancia del reconocimiento de imágenes y objetos en Inteligencia Artificial (AI) e indicar varias aplicaciones significativas de esta tecnología [Usar] • 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] • Implementar reconocimiento de objetos en 2d basados en la representación del contorno y/o regiones basadas en formas [Usar] • 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] • 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] • Implementar un algoritmo de clasificación que segmenta percepciones de entrada en categorías de salida y evalúa cuantitativamente la clasificación resultante [Usar] • 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]
Readings : [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 2026-I

1. COURSE

CS272-CS2023. Databases II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS272-CS2023. Databases II
2.2 Semester	: 5 th 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	: CS271-CS2023. Data Management. (4 th 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

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Familiarity)

7. TOPICS

Unit 1: Diseño Físico de Bases de Datos (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Almacenamiento y estructura de archivos. • Archivos indexados. • Archivos Hash. • Archivos de Firma. • Árboles B. • Archivos con índice denso. • Archivos con registros de tamaño variable. • Eficiencia y Afinación de Bases de Datos.	• Explica los conceptos de registro, tipos de registro, y archivos, así como las diversas técnicas para colocar registros de archivos en un disco [Usar] • Da ejemplos de la aplicación de índices primario, secundario y de agrupamiento [Usar] • Distingue entre un índice no denso y uno denso [Usar] • Implementa índices de multinivel dinámicos usando árboles-B [Usar] • Explica la teoría y la aplicación de técnicas de hash internas y externas [Usar] • Usa técnicas de hasp para facilitar la expansión de archivos dinámicos [Usar] • Describe las relaciones entre hashing, compresión, y búsquedas eficientes en bases de datos [Usar] • Evalúa el costo y beneficio de diversos esquemas de hashing [Usar] • Explica como el diseño físico de una base de datos afecta la eficiencia de las transacciones en ésta [Usar]
Readings : [Bur04], [Cel05]	

Unit 2: Procesamiento de Transacciones (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Transacciones. • Fallo y recuperación. • Control concurente. • Interacción de gestión de transacciones con el almacenamiento, especialmente en almacenamiento.	• Crear una transacción mediante la incorporación de SQL en un programa de aplicación [Usar] • Explicar el concepto de confirmaciones implícitas [Usar] • Describir los problemas específicos para la ejecución de una transacción eficiente [Usar] • Explicar cuando y porqué se necesita un <i>rollback</i>, y cómo registrar todo asegura un <i>rollback</i> adecuado [Usar] • Explicar el efecto de diferentes niveles de aislamiento sobre los mecanismos de control de concurrencia [Usar] • Elejir el nivel de aislamiento adecuado para la aplicación de un protocolo de transacción especificado [Usar] • Identificar los límites apropiados de la transacción en programas de aplicación [Usar]
Readings : [Phi97], [Ram04]	

Unit 3: Almacenamiento y Recuperación de Información (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Documentos, publicación electrónica, markup, y lenguajes markup. • Tries, archivos invertidos, Árboles PAT, archivos de firma, indexación. • Análisis Morfológico, stemming, frases, stop lists. • Distribuciones de frecuencia de términos, incertidumbre, fuzificación (fuzzyness), ponderación. • Espacio vectorial, probabilidad, lógica, y modelos avanzados. • Necesidad de Información , Relevancia, evaluación, efectividad. • Thesauri, ontologías, clasificación y categorización, metadata. • Información bibliográfica, bibliometría, citaciones. • Enrutamiento y filtrado. • Búsqueda multimedia. • Información de resumen y visualización. • Búsqueda por facetas (por ejemplo, el uso de citas, palabras clave, esquemas de clasificación). • Librerías digitales. • Digitalización, almacenamiento, intercambio, objetos digitales, composición y paquetes. • Metadata y catalogación. • Nombramiento, repositorios, archivos • Archivamiento y preservación, integridad • Espacios (Conceptual, geográfico, 2/3D, Realidad virtual) • Arquitecturas (agentes, autobuses, envolturas / mediadores), de interoperabilidad. • Servicios (búsqueda, de unión, de navegación, y así sucesivamente). • Gestión de derechos de propiedad intelectual, la privacidad y la protección (marcas de agua).	• Explica los conceptos básicos de almacenamiento y recuperación de la información [Usar] • Describe que temas son específicos para una recuperación de la información eficiente [Usar] • Da aplicaciones de estrategias alternativas de búsqueda y explica porqué una estrategia en particular es apropiada para una aplicación [Usar] • Diseña e implementa un sistema de almacenamiento y recuperación de la información o librería digital de tamaño pequeño a mediano [Usar] • Describe algunas de las soluciones técnicas a los problemas relacionados al archivamiento y preservación de la información en una librería digital [Usar]
Readings : [Pet98], [Ram04]	

Unit 4: Bases de Datos Distribuidas (36 hours)	
Competences Expected:	
Topics	Learning Outcomes
• DBMS Distribuidas – Almacenamiento de datos distribuido – Procesamiento de consultas distribuido – Modelo de transacciones distribuidas – Soluciones homogéneas y heterogéneas – Bases de datos distribuidas cliente-servidor • Parallel DBMS – Arquitecturas paralelas DBMS: memoria compartida, disco compartido, nada compartido; – Aceleración y ampliación, por ejemplo, el uso del modelo de procesamiento MapReduce – Replicación de información y modelos de consistencia débil	• Explicar las técnicas usadas para la fragmentación de datos, replicación, y la asignación durante el proceso de diseño de base de datos distribuida [Usar] • Evaluar estrategias simples para la ejecución de una consulta distribuida para seleccionar una estrategia que minimise la cantidad de transferencia de datos [Usar] • 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] • Describir el control concurrente distribuido basados en técnicas de copia distinguidos y el método de votación. [Usar] • Describir los tres niveles del software en el modelo cliente servidor [Usar]
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 Ozsu. *Principles of Distributed Database Systems, Second Edition*. Prentice Hall, 1999.
- [Bur04] Donald K. Burleson. *Physical Database Design Using Oracle*. CRC Press, 2004.
- [Ram04] Shamkant B. Navathe Ramez Elmasri. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.
- [Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS291-CS2023. Software Engineering I (Mandatory)

2. GENERAL INFORMATION

- | | | |
|-----------------------------------|---|---|
| 2.1 Course | : | CS291-CS2023. Software Engineering I |
| 2.2 Semester | : | 5 th 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 | : | <ul style="list-style-type: none">• CS113. Objects oriented programming II. (3rd Sem)• CS271. Data Management. (4th Sem) |

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Modern software engineering requires professionals capable of designing robust and maintainable systems in cloud-native and agile environments. This course develops competencies in clean architecture, design patterns, professional testing, and DevSecOps practices, preparing students for the challenges of today's industry.

5. GOALS

- Develop competencies in designing scalable and maintainable software architectures by applying SOLID principles and established design patterns.
- Master professional development tools including typed Python, advanced pytest, collaborative Git, and automation with Makefiles.
- Implement agile methodologies with quality metrics, project management, and continuous improvement in development teams.
- Apply security principles in software design and testing practices that ensure the reliability of critical systems.
- Produce professional technical documentation using international standards such as C4 Model and UML to communicate design decisions.

6. COMPETENCES

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Familiarity)

7. TOPICS

Unit 1: Version Control and Team Development (Git) (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Workflows: <i>trunk-based</i> (recommended) vs. Git-flow. • Interactive rebase, cherry-pick, bisect, reflog. • <i>Conventional Commits</i> messages and professional Pull Requests. • Git hooks + basic GitHub Actions (linters + tests). • Resolution of complex conflicts in code and configuration files. • Meta-goal: at least 50 PRs reviewed and received during the course.	• Evaluate and select between modern workflows (trunk-based vs. Git-flow) according to project and team characteristics. • Apply advanced Git commands (rebase, cherry-pick, bisect) to maintain a clean history, selectively incorporate changes, and efficiently debug error introduction. • Create commit messages following the Conventional Commits standard and draft professional Pull Requests that facilitate code review. • Configure and use Git hooks and basic GitHub Actions pipelines to automate quality verification (linters) and test execution. • Resolve complex conflicts in merges or rebases, both in code and configuration files, analyzing differences and making integrative decisions. • Demonstrate competence in a collaborative development environment through active code review and incorporation of feedback in their own work (meta-goal of 50 PRs).
Readings : [CS23], [TH24], [Fow19], [Mic23]	

Unit 2: Linux and Bash for Developers (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Processes, signals, permissions, pipes and redirections. • grep, awk, sed, find, rg, fd on real code and logs. • Robust Bash scripting: set -euo pipefail, functions, getopts, trap. • SSH + keys, configuration in ssh-config, <i>agent forwarding</i>. • Introduction to Makefile as a local orchestration tool.	• Manage system processes, file permissions, and data flows efficiently using pipes, redirections, and signals. • Build text processing pipelines to analyze code and logs in real time, using tools like grep, awk, sed and their modern alternatives (rg, fd). • Design robust and maintainable Bash scripts that include strict error handling, named parameters, and resource cleanup through trap. • Configure and use secure SSH connections with key authentication, multiplexing, and agent forwarding for efficient remote development. • Use Makefile to orchestrate complex development tasks, automating build, testing, and local deployment workflows.
Readings : [Sho19], [RN24], [Ker21], [Fou23]	

Unit 3: Procesos de Software (4 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno. • Introducción a modelos del proceso de software (e.g., cascada, incremental, agil): – Actividades con ciclos de vida de software. • Programación a gran escala versus programación individual. • Evaluación de modelos de proceso de software. • Conceptos de calidad de software. • Mejoramiento de procesos. • Modelos de madurez de procesos de software. • Mediciones del proceso de software.	
Readings : [PM19], [Som20], [Mar17]	

Unit 4: Ingeniería de Requisitos (6 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Al describir los requisitos funcionales utilizando, por ejemplo, los casos de uso o historias de los usuarios. • Propiedades de requisitos, incluyendo la consistencia, validez, integridad y viabilidad. • Requisitos de software elicitation. • Descripción de datos del sistema utilizando, por ejemplo, los diagramas de clases o diagramas entidad-relación. • Requisitos no funcionales y su relación con la calidad del software. • Evaluación y uso de especificaciones de requisitos. • Requisitos de las técnicas de modelado de análisis. • La aceptabilidad de las consideraciones de certeza/incertidumbre sobre el comportamiento del software/sistema. • Prototipos. • Conceptos básicos de la especificación formal de requisitos. • Especificación de requisitos. • Validación de requisitos. • Rastreo de requisitos.	
Readings : [WB13], [Coh04], [Robertson2012]	

Unit 5: Diseño de Software (16 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
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 standar widget set) • Diseños de refactorización utilizando patrones de diseño • 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. • Aaplicaciones 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 seccurre design). – Principio de privilegios mínimos – Principio de falla segura por defecto – Principio de aceptabilidad psicológica	
4 Readings : [Mar17], [Gam+94], [New21], [Fow18]	

Unit 6: Principios de Diseño Seguro (4 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Menor privilegio y aislamiento. • Valores predeterminados a prueba de fallos. • Diseño abierto. • La seguridad de extremo a extremo. • La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas) • Diseño de seguridad. • Las tensiones entre la seguridad y otros objetivos de diseño. • Mediación completa. • El uso de componentes de seguridad vetados. • Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque) • Seguridad utilizable. • Componibilidad de seguridad. • Prevención, detección y disuasión.	
Readings : [HL06], [Mar17], [Sho14]	

Unit 7: Verificación y Validación de Software (10 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Verificación y validación de conceptos. • Inspecciones, revisiones, auditorias. • Tipos de pruebas, incluyendo la interfaz humano computador, usabilidad, confiabilidad, seguridad, desempeño para la especificación. • Fundamentos de testeo: – Pruebas de Unit, integración, validación y de Sistema – Creación de plan de pruebas y generación de casos de test – Técnicas de test de caja negra y caja blanca – Test de regresión y automatización de pruebas • Seguimiento de defectos. • Limitaciones de testeo en dominios particulares, tales como sistemas paralelos o críticos en cuanto a seguridad. • Enfoques estáticos y enfoques dinámicos para la verificación. • Desarrollo basado en pruebas. • Plan de Validación, documentación para validación. • Pruebas Orientadas a Objetos, Sistema de Pruebas. • Verificación y validación de artefactos no codificados (documentación, archivos de ayuda, materiales de entrenamiento) • Logeo fallido, error crítico y apoyo técnico para dichas actividades. • Estimación fallida y terminación de las pruebas que incluye la envíos por defecto.	
Readings : [Bec02], [Mes07], [FP09]	

Unit 8: Métodos de Desarrollo (4 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Comprensión de programas. • Exactitud del programa. – Tipos de error (syntaxis, logica, tiempo de ejecución) – El concepto de la especificacion – Programación defensiva (ejem., codificación segura, manejo de excepciones) – Revision de codigo – Fundamentos de <i>testing</i> y generación de casos de prueba – El rol y el uso de contratos, incluyendo pre- y post- condiciones – Pruebas unitarias • Refactorización simple. • Entornos modernos de programación: – Búsqueda de código. – Programación usando libreria de componentes y sus APIs. • Estrategias de depuración. • Documentación y estilo de programas.	
Readings : [Fow18], [Mar17], [Eva03]	

Unit 9: Algoritmos y Diseño (6 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Conceptos y propiedades de los algoritmos – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación	
Readings : [McC04], [Mar17], [SW11]	

Unit 10: Fundamentos y Conceptos en Seguridad (6 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• CIA (Confidencialidad, Integridad, Disponibilidad) • Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque . • Autenticación y autorización, control de acceso (vs. obligatoria discrecional) • Concepto de la confianza y la honradez . • Ética (revelación responsable)	
Readings : [Smith2019], [Sho14], [HL06]	

Unit 11: Gestión de Proyectos de Software (5 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• La participación del equipo: – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) • Estimación de esfuerzo (a nivel personal) • Riesgo. – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • Gestión de equipos: – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación • Gestión de proyectos: – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio • Software de medición y técnicas de estimación. • Aseguramiento de la calidad del software y el rol de las mediciones. • Riesgo. – Identificación de riesgos y gestión. – Análisis riesgo y evaluación. – La tolerancia al riesgo (por ejemplo, riesgo adverso, riesgo neutral, la búsqueda de riesgo) – Planificación de Riesgo • En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.	
Readings : [Coh09], [Schwaber2020], [Reinertsen2009]	

Unit 12: Evolución de Software (3 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Desarrollo de Software en el contexto de código grande pre existente – Cambios de software – Preocupaciones y ubicación de preocupaciones – <i>Refactoring</i> • Evolución de Software. • Características de Software mantenible. • Sistemas de Reingeniería. • Reuso de Software. – Segmentos de código – Bibliotecas y <i>frameworks</i> – Componentes – Líneas de Producto	
Readings : [Fea04], [Fow18], [Mar17]	

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS2S1-CS2023. Operating systems (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS2S1-CS2023. Operating systems
2.2 Semester	: 5 th 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-CS2023. Computer Systems Architecture. (4 th 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

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)

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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Visión general de Sistemas Operativos (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Los mecanismos de apoyo modelos cliente-servidor, dispositivos de mano. • Cuestiones de diseño (eficiencia, robustez, flexibilidad, portabilidad, seguridad, compatibilidad) • Influencias de seguridad, creación de redes, multimedia, sistemas de ventanas.	• Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Evaluar] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 2: Principios de Sistemas Operativos (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Métodos de estructuración (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de usuario de estado / sistema y la protección, la transición al modo kernel.	• Explicar el concepto de una capa lógica [Familiarizarse] • Explicar los beneficios de construir capas abstractas en forma jerárquica [Familiarizarse] • Describir el valor de la API y <i>middleware</i> [Familiarizarse] • Describir como los recursos computacionales son usados por aplicaciones de software y administradas por el software del sistema [Familiarizarse] • Contrastar el modo <i>kernel</i> y modo usuario en un sistema operativo [Evaluar] • Discutir las ventajas y desventajas del uso de procesamiento interrumpido [Familiarizarse] • Explicar el concepto de una capa lógica [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 3: Concurrency (9 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Diagramas de estado. • Estructuras (lista preparada, bloques de control de procesos, y así sucesivamente) • Despacho y cambio de contexto. • El papel de las interrupciones. • Gestionar el acceso a los objetos del sistema operativo atómica. • La implementación de primitivas de sincronización. • Cuestiones multiprocesador (spin-locks, reentrada)	• Describir la necesidad de concurrencia en el marco de un sistema operativo [Familiarizarse] • Demostrar los potenciales problemas de tiempo de ejecución derivados de la operación simultánea de muchas tareas diferentes [Usar] • 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] • Explicar los diferentes estados por los que una tarea debe pasar y las estructuras de datos necesarias para el manejo de varias tareas [Familiarizarse] • Resumir las técnicas para lograr sincronización en un sistema operativo (por ejemplo, describir como implementar semáforos usando primitivas del sistema operativo.) [Familiarizarse] • Describir las razones para usar interruptores, despacho, y cambio de contexto para soportar concurrencia en un sistema operativo [Familiarizarse] • Crear diagramas de estado y transición para los dominios de problemas simples [Usar]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 4: Planificación y despacho (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Planificación preventiva y no preferente. • Planificadores y políticas. • Procesos y subprocesos. • Plazos y cuestiones en tiempo real.	• 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] • Describir las relaciones entre los algoritmos de planificación y dominios de aplicación [Familiarizarse] • Discutir los tipos de planeamiento de procesos <i>scheduling</i> de corto, a mediano, a largo plazo y I/O [Familiarizarse] • Describir las diferencias entre procesos y hebras [Familiarizarse] • Comparar y contrastar enfoques estáticos y dinámicos para <i>scheduling</i> en tiempo real [Evaluar] • Hablar sobre la necesidad de tiempos límites de <i>scheduling</i> [Familiarizarse] • 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]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 5: Manejo de memoria (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Revisión de la memoria física y hardware de gestión de memoria. • Conjuntos de trabajo y thrashing. • El almacenamiento en caché	• Explicar la jerarquía de la memoria y costo-rendimiento de intercambio [Familiarizarse] • Resumir los principios de memoria virtual tal como se aplica para el almacenamiento en cache y paginación [Familiarizarse] • Evaluar las ventajas y desventajas en términos del tamaño de memoria (memoria principal, memoria caché, memoria auxiliar) y la velocidad del procesador [Evaluar] • Defiende las diferentes formas de asignar memoria a las tareas, citando las ventajas relativas de cada uno [Familiarizarse] • 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] • 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]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 6: Seguridad y protección (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Visión general de la seguridad del sistema . • Política / mecanismo de separación. • Métodos de seguridad y dispositivos. • Protección, control de acceso y autenticación. • Las copias de seguridad.	• Articular la necesidad para la protección y seguridad en un sistema operativo [Familiarizarse] • Resumir las características y limitaciones de un sistema operativo usado para proporcionar protección y seguridad [Familiarizarse] • Explicar el mecanismo disponible en un OS para controlar los accesos a los recursos [Familiarizarse] • Realizar tareas de administración de sistemas sencillas de acuerdo a una política de seguridad, por ejemplo la creación de cuentas, el establecimiento de permisos, aplicación de parches y organización de backups regulares [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 7: Máquinas virtuales (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Tipos de virtualización (incluyendo Hardware / Software, OS, Servidor, Servicio, Red) • Paginación y la memoria virtual. • Sistemas de archivos virtuales. • Los Hypervisor. • Virtualización portátil; emulación vs aislamiento. • Costo de la virtualización.	• Explicar el concepto de memoria virtual y la forma cómo se realiza en hardware y software [Familiarizarse] • Diferenciar emulación y el aislamiento [Familiarizarse] • Evaluar virtualización de compensaciones [Evaluar] • Discutir sobre hipervisores y la necesidad para ellos en conjunto con diferentes tipos de hipervisores [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 8: Manejo de dispositivos (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Características de los dispositivos serie y paralelo. • Haciendo de abstracción de dispositivos. • Estrategias de buffering. • Acceso directo a memoria. • La recuperación de fallos.	• Explique la diferencia clave entre dispositivos seriales y paralelos e identificar las condiciones en las cuales cada uno es apropiado [Familiarizarse] • Identificar los requerimientos para recuperación de errores [Familiarizarse] • Explique <i>buffering</i> y describir las estrategias para su aplicación [Familiarizarse] • 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] • Describir las ventajas y desventajas de acceso directo a memoria y discutir las circunstancias en cuales se justifica su uso [Familiarizarse] • Identificar la relación entre el hardware físico y los dispositivos virtuales mantenidos por el sistema operativo [Familiarizarse] • Implementar un controlador de dispositivo simple para una gama de posibles equipos [Usar]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 9: Sistema de archivos (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Archivos: los datos, metadatos, operaciones, organización, amortiguadores, secuenciales, no secuencial. • Directorios: contenido y estructura. • Los sistemas de archivos: partición, montar sistemas de archivos / desmontar, virtuales. • Técnicas estándar de implementación . • Archivos asignados en memoria. • Sistemas de archivos de propósito especial. • Naming, búsqueda, acceso, copias de seguridad. • La bitacora y los sistemas de archivos estructurados (log)	• Describir las decisiones que deben tomarse en el diseño de sistemas de archivos [Familiarizarse] • Comparar y contrastar los diferentes enfoques para la organización de archivos, el reconocimiento de las fortalezas y debilidades de cada uno. [Evaluar] • 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] • Resumir el uso de diarios y como los sistemas de archivos de registro estructurado mejora la tolerancia a fallos [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 10: Sistemas empujados y de tiempo real (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Proceso y programación de tareas. • Los requisitos de gestión de memoria / disco en un entorno en tiempo real. • Los fracasos, los riesgos y la recuperación. • Preocupaciones especiales en sistemas de tiempo real.	• Describir que hace a un sistema un sistema en tiempo real [Familiarizarse] • Explicar la presencia y describir las características de latencia en sistemas de tiempo real [Familiarizarse] • Resumir los problemas especiales que los sistemas en tiempo real presentan, incluyendo el riesgo, y cómo se tratan estos problemas [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 11: Tolerancia a fallas (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Conceptos fundamentales: sistemas fiables y disponibles. • Redundancia espacial y temporal. • Los métodos utilizados para implementar la tolerancia a fallos. • 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.	• Explicar la importancia de los términos tolerancia a fallos, fiabilidad y disponibilidad [Familiarizarse] • Explicar en términos generales la gama de métodos para implementar la tolerancia a fallos en un sistema operativo [Familiarizarse] • Explicar cómo un sistema operativo puede continuar funcionando después de que ocurra una falla [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [AD14]	

Unit 12: Evaluación del desempeño de sistemas (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
• ¿Por qué el rendimiento del sistema debe ser evaluado? • ¿Qué se va a evaluar? • Sistemas de políticas de rendimiento, por ejemplo, el almacenamiento en caché, de paginación, la programación, la gestión de memoria, y la seguridad. • Modelos de evaluación: analítica, simulación, o de implementación específico determinista. • Cómo recoger los datos de evaluación (perfiles y mecanismos de localización)	• Describir las medidas de rendimiento utilizados para determinar cómo el sistema funciona [Familiarizarse] • Explicar los principales modelos de evaluación utilizados para evaluar un sistema [Familiarizarse]
Readings : [Avi12], [Sta05], [Tan06], [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

- [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 2026-I

1. COURSE

ID105. Technical and professional English V (Elective)

2. GENERAL INFORMATION

2.1 Course	: ID105. Technical and professional English V
2.2 Semester	: 5 th Semester
2.3 Credits	: 2
2.4 Horas	: 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	: ID104. Technical and professional English IV. (4 th 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

3) Communicate effectively in a variety of professional contexts.. (Assessment)

AG-C04 Communication: Communicates effectively in complex computing activities. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C11 Tool Usage: Applies modern computing tools in problem-solving. (Assessment)

5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. (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: It's a wonderful world (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Auxiliary verbs • Verb tenses • Negative Questions and Prayers • Short answers • Word formation • Colloquial expressions • Error correction	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 2: Happiness! (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Simple present • Present continuous • Passive Voice in Present • Verbs for sports and free time • Types of numbers and • Inventions / Modern World • Corrección de errores	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 3: Telling tales! (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Simple Past Time • Past Continuous • Passive Voice in Past • Vocabulary of Art and Literature • Expressions to give and ask opinions • Stories and stories	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 4: Doing the right thing! (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Mode Auxiliary Verbs I. • Affirmative, Negative and Modals Questions • Use of nationalities and other adjectives • Expressions of orders and offers • Guide to Good Manners • Form Fill • Phonetic symbols	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 5: On the move! (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Future with Will • Future Time Prayers with going to • Use of might for future • Climate Expressions • Vocabulary of the climate • Expressions for hotels and transportation • E-mails	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 6: I just love it! (0 hours)	
Competences Expected: 5	
Topics	Learning Outcomes
• Questions with Shapes Like • Patrones Verbales II • Vocabulario de Comida, Lugares y ocupaciones • Palabras que van unidas en contexto • Expresiones para vistas y sonidos • Composición de Impresiones personales	• 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.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

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

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

Chapter 6

Sixth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS231-CS2023. Networking and Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS231-CS2023. Networking and Communication
2.2 Semester	: 6 th 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	: CS2S1-CS2023. Operating systems . (5 th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Introducción a redes (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Técnicas de Switching (por ejemplo, de circuitos, de paquetes) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Principios de capas (encapsulación, multiplexación) • Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)	• Articular la organización de la Internet [Familiarizarse] • Listar y definir la terminología de red apropiada [Familiarizarse] • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse] • Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse]
Readings : [KR13]	

Unit 2: Aplicaciones en red (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Esquemas de denominación y dirección (DNS, direcciones IP, identificadores de recursos uniformes, etc) • Las aplicaciones distribuidas (cliente / servidor, peer-to-peer, nube, etc) • HTTP como protocolo de capa de aplicación . • Multiplexación con TCP y UDP • API de Socket	• Listar las diferencias y las relaciones entre los nombres y direcciones en una red [Familiarizarse] • Definir los principios detrás de esquemas de denominación y ubicación del recurso [Familiarizarse] • Implementar una aplicación simple cliente-servidor basada en <i>sockets</i> [Usar]
Readings : [KR13]	

Unit 3: Entrega confiable de datos (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Control de errores (técnicas de retransmisión, temporizadores) • El control de flujo (agradecimientos, ventana deslizante) • Problemas de rendimiento (pipelining) • TCP	• Describir el funcionamiento de los protocolos de entrega fiables [Familiarizarse] • Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse] • Diseñar e implementar un protocolo confiable simple [Usar]
Readings : [KR13]	

Unit 4: Ruteo y reenvío (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Enrutamiento vs reenvío . • Enrutamiento estático . • Protocolo de Internet (IP) • Problemas de escalabilidad (direccionamiento jerárquico)	• Describir la organización de la capa de red [Familiarizarse] • Describir cómo los paquetes se envían en una red IP [Familiarizarse] • Listar las ventajas de escalabilidad de direccionamiento jerárquico [Familiarizarse]
Readings : [KR13]	

Unit 5: Redes de área local (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Problemas de Acceso Múltiple. • Enfoques comunes a Acceso múltiple (exponencial backoff, multiplexación por división de tiempo, etc) • Redes de área local . • Ethernet . • Switching .	• Describir como los paquetes son enviados en una red Ethernet [Familiarizarse] • Describir las diferencias entre IP y Ethernet [Familiarizarse] • Describir las etapas usadas en un enfoque común para el problema de múltiples accesos [Familiarizarse]
Readings : [KR13]	

Unit 6: Asignación de recursos (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Necesidad de asignación de recursos . • Asignación fija (TDM, FDM, WDM) versus la asignación dinámica . • De extremo a extremo frente a las red de enfoque asistida . • Justicia. • Principios del control de congestión. • Enfoques para la congestión (por ejemplo, redes de distribución de contenidos)	• Describir como los recursos pueden ser almacenados en la red [Familiarizarse] • Describir los problemas de congestión en una red grande [Familiarizarse] • Comparar y contrastar las técnicas de almacenamiento estático y dinámico [Familiarizarse] • Comparar y contrastar los enfoques actuales de la congestión [Familiarizarse]
Readings : [KR13]	

Unit 7: Celulares (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Principios de redes celulares. • Redes 802.11 • Problemas en el apoyo a los nodos móviles (agente local)	• Describir la organización de una red inalámbrica [Familiarizarse] • Describir como las redes inalámbricas soportan usuarios móviles [Familiarizarse]
Readings : [KR13], [Cha16]	

Unit 8: Redes sociales (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Panorama de las redes sociales. • Ejemplo plataformas de redes sociales. • Estructura de los grafos de redes sociales. • Análisis de redes sociales.	• Discutir los principios fundamentales (como pertenencia, confianza) de una red social [Familiarizarse] • Describir como redes sociales existentes operan [Familiarizarse] • Construir un grafo de una red social a partir de datos de la red [Usar] • Analizar una red social para determinar quienes son las personas importantes [Usar] • Evaluar una determinada interpretación de una pregunta de red social con los datos asociados [Familiarizarse]
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 2026-I

1. COURSE

CS311-CS2023. Competitive Programming (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS311-CS2023. Competitive Programming
2.2 Semester	: 6 th 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	: CS212-CS2023. Analysis and Design of Algorithms. (5 th 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. (Assessment)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Introduction (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Introduction to Competitive Programming • Computational model • Runtime and space complexity • Recurrence and recursion • Divide and conquer	• Identify and learn how to use the resources in the Random Access Machine (RAM) computational model. [Usar] • Compute the runtime and space complexity for written algorithms. [Usar] • Compute the recurrence relations for recursive algorithms. [Usar] • Solve problems related to searching and sorting. [Usar] • Learning to select the right algorithms for divide-and-conquer problems. [Usar] • Design new algorithms for real-world problem solving. [Usar]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 2: Data structure (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Arrays and strings problems • Linked lists problems • Stacks and queues problems • Trees problems • Hash tables problems • Heaps problems	• Recognize different data structures, their complexities, uses and restrictions.[Usar] • Identify the type of data structure appropriate to the resolution of the problem. [Usar] • Recognize types of problems associated with operations on data structures such as searching, inserting, deleting and updating.[Usar]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 3: Algorithmic Design Paradigms (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Brute force • Divide and conquer • Backtracking • Greedy • Dynamic Programming	• Learning the different algorithmic design paradigms.[Usar] • Learning to select the right algorithms for different problems applying different algorithmic design paradigms.[Usar]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 4: Graphs (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Graphs transversal • Graphs applications • Shortest path • Networks and flows	• Identify problems classified as graph problems. [Usar] • Learn how to select the right algorithms for network problems (transversal, MST, shortest-path, network and flows). [Usar]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 5: Advanced topics (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Number theory • Probabilities and combinations • String algorithms (tries, string hashing, z-algorithm) • Geometric algorithms	• Learning to select the right algorithms for problems in number theory and mathematics as they are important in competitive programming. [Usar] • Learning to select the right algorithms for problems about probabilities and combinations, strings and computational geometry. [Usar]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 6: Domain specific problems (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Latency and throughput • Parallelism • Networks • Storage • High availability • Caching • Proxies • Load balancers • Key-value stores • Replicating and sharing • Leader election • Rate limiting • Logging and monitoring	• Learning to design systems for different domain-specific problems by applying knowledge about networks, distributed computing, high availability, storage and system architecture.[Usar]
Readings : [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*. Stringer, 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 2026-I

1. COURSE

CS312-CS2023. Advanced Data Structures (Mandatory)

2. GENERAL INFORMATION

- | | | |
|-----------------------------------|---|--|
| 2.1 Course | : | CS312-CS2023. Advanced Data Structures |
| 2.2 Semester | : | 6 th 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 | : | <ul style="list-style-type: none">• CS212-CS2023. Analysis and Design of Algorithms. (5th Sem)• ID104. Technical and professional English IV. (4th 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

- 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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Multidimensional Data (16 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Introducción al curso. • Introducción a datos multidimensionales. • Maldición de la dimensionalidad.	• Introducir la transcendencia de la representación multidimensional de datos. [Usar] • Entender la complejidad de lidiar con datos multidimensional y de alta dimensión.[Usar] • Entender la maldición de la dimensionalidad, y su impacto en el indizado de grandes volúmenes de datos.[Usar] • Presentar y discutir aplicaciones reales de datos multidimensionales en motores de búsqueda.[Usar]
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 2: Multidimensional Acces Data Structures (16 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Introducción a estructuras de datos espaciales. • Estructuras espaciales, Quadtree,Octree y visualización. • Kd-Tree. • Introducción a R-Tress. • R tree (Guttman). • R+ tree. • R* tree. • Variación R*-tree y relación con paginación y tamaño de bloques. • X-tree.	• Introducir los fundamentos teóricos de estructuras de datos espaciales. • Entender los beneficios y limitaciones deestructuras de datos espaciales basadas en árbol. • Implementar diferentes estructuras de datos para el indizado de grandes volumenenes de datos. • Entender los fundamentos e implementar estrategias de búsqueda como vecinos mas próximos y búsquedas por rango.
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 3: Approximate Access Methods (20 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Métodos de Acceso Métrico para distancias discretas • Métodos de Acceso Métrico para distancias continuas	• Que el alumno entienda conozca e implemente algunos métodos de acceso métrico[Usar] • Que el alumno entienda la importancia de estos Métodos de Acceso para la Recuperación de Información por Similitud[Usar]
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 4: Métodos de Acceso Aproximados (20 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Space Filling Curves: Hilbert curve y Z-order • Proyecciones y complejidad. • Locally sensitive hashing (LSH)	• Entender, conocer e implementar algunos métodos de acceso aproximados. • 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.
Readings : [Sam06], [PI06], [Zez+07]	

Unit 5: Clustering (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Introducción a Clustering. • Kmeans y DBScan. • Clustering Applications. • Clustering Ensemble.	• Introducir los fundamentos teóricos para el clustering de datos multidimensionales. • Implementar diferentes estrategias para el clustering de datos multidimensionales, como basados en partición, en jerarquía o en densidad. • Entender los fundamentos, aplicaciones e implementar ensambles de métodos de clustering. • Implementar ensambles de métodos de clustering con datos reales.
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 6: Temporal Data Structures (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Introducción a Estructuras de datos temporales. • Versionando la estructura de Datos. • Persistencia • Retroactividad	• Introducir los fundamentos teóricos de estructuras de datos temporales. • Entender, discutir e implementar Persistencia y sus tipos. • Entender, discutir e implementar Retroactividad y sus tipos. • Entender y discutir los beneficios y limitaciones entre persistencia y retroactividad.
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94]	

Unit 7: Final Talks (8 hours)	
Competences Expected: 1,2	
Topics	Learning Outcomes
• Seminarios de trabajo de investigación.	• Investigar sobre nuevos métodos para el indizado de grandes volúmenes de datos complejos. • Presentar y dirigir la discusión sobre métodos para indizados de Big Data investigado.
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 2026-I

1. COURSE

CS342-CS2023. Compilers (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS342-CS2023. Compilers
2.2 Semester	: 6 th 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-CS2023. Theory of Computation. (4 th 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

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)

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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Representación de programas (5 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Programas que tienen otros programas como entrada tales como interpretes, compiladores, revisores de tipos y generadores de documentación. • Árboles de sintaxis abstracta, para contrastar la sintaxis correcta. • Estructuras de datos que representan código para ejecución, traducción o transmisión. • Compilación en tiempo just-in time y re-compilación dinámica. • Otras características comunes de las máquinas virtuales, tales como carga de clases, hilos y seguridad.	• Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse] • Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Familiarizarse] • Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse] • 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] • 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] • Discutir las ventajas, desventajas y dificultades del término (<i>just-in-time</i>) y recompilación automática [Familiarizarse] • Identificar los servicios proporcionados por los sistemas de tiempo de ejecución en lenguajes modernos [Familiarizarse]
Readings : [Lou04b]	

Unit 2: Traducción y ejecución de lenguajes (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Interpretación vs. compilación a código nativo vs. compilación de representación portable intermedia. • Pipeline de traducción de lenguajes: análisis, revisión opcional de tipos, traducción, enlazamiento, ejecución: – Ejecución como código nativo o con una máquina virtual – Alternativas como carga dinámica y codificación dinámica de código (o “just-in-time”) • Representación en tiempo de ejecución de construcción del lenguaje núcleo tales como objetos (tablas de métodos) y funciones de primera clase (cerradas) • Ejecución en tiempo real de asignación de memoria: pila de llamadas, montículo, datos estáticos: – Implementación de bucles, recursividad y llamadas de cola • Gestión de memoria: – Gestión manual de memoria: asignación, limpieza y reuso de la pila de memoria – Gestión automática de memoria: recolección de datos no utilizados (<i>garbage collection</i>) como una técnica automática usando la noción de accesibilidad	• 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] • Distinguir sintaxis y parseo de la semántica y la evaluación [Evaluar] • Bosqueje una representación de bajo nivel de tiempo de ejecución de construcciones del lenguaje base, tales como objetos o cierres (<i>closures</i>) [Evaluar] • 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] • Identificar y corregir las pérdidas de memoria y punteros desreferenciados [Evaluar] • Discutir los beneficios y limitaciones de la recolección de basura (<i>garbage collection</i>), incluyendo la noción de accesibilidad [Evaluar]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 3: Análisis de sintaxis (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Exploración (análisis léxico) usando expresiones regulares. • 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. • Generación de exploradores (scanners) y analizadores a partir de especificaciones declarativas.	• Usar gramáticas formales para especificar la sintaxis de los lenguajes [Evaluar] • Usar herramientas declarativas para generar parseadores y escáneres [Evaluar] • Identificar las características clave en las definiciones de sintaxis: ambigüedad, asociatividad, precedencia [Evaluar]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 4: Análisis semántico de compiladores (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Representaciones de programas de alto nivel tales como árboles de sintaxis abstractas. - Alcance y resolución de vínculos. - Revisión de tipos. - Especificaciones declarativas tales como gramáticas atribuidas.	- Implementar analizadores sensibles al contexto y estáticos a nivel de fuente, tales como, verificadores de tipos o resolutores de identificadores para identificar las ocurrencias de vínculo [Evaluar] - Describir analizadores semánticos usando una gramática con atributos [Evaluar]
Readings : [Aho+11], [Lou04a], [App02], [TS98]	

Unit 5: Generación de código (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Llamadas a procedimientos y métodos en envío. - Compilación separada; vinculación. - Selección de instrucciones. - Calendarización de instrucciones. - Asignación de registros. - Optimización por rendija (peephole)	- Identificar todos los pasos esenciales para convertir automáticamente código fuente en código ensamblador o otros lenguajes de bajo nivel [Evaluar] - Generar código de bajo nivel para llamadas a funciones en lenguajes modernos [Evaluar] - Discutir por qué la compilación separada requiere convenciones de llamadas uniformes [Evaluar] - Discutir por qué la compilación separada limita la optimización debido a efectos de llamadas desconocidas [Evaluar] - 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]
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 2026-I

1. COURSE

AI263. Introduction to Machine Learning (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: AI263. Introduction to Machine Learning
2.2 Semester	: 6 th 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	: CS261-CS2023. Artificial Intelligence. (5 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamentals of machine learning, covering classical and modern algorithms for classification, regression, and clustering problems. It focuses on practical implementation using scikit-learn and TensorFlow, with applications in computer vision and natural language processing.

5. GOALS

- Understand the mathematical principles behind ML algorithms.
- Implement complete ML pipelines with Python.
- Evaluate and optimize models using standard metrics.

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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

7. TOPICS

Unit 1: ML Fundamentals (15 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• Supervised vs unsupervised learning • Overfitting and regularization • Cross-validation and learning curves	• Explain the bias-variance tradeoff [Familiarizarse] • Implement k-fold cross-validation [Usar]
Readings : [Bis06], [GBC16]	

Unit 2: Linear Models (15 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• Linear and logistic regression • Support Vector Machines (SVM) • Linear discriminant analysis	• Program linear models with scikit-learn [Usar] • Interpret regression coefficients [Evaluar]
Readings : [HTF09], [Mur12]	

Unit 3: Ensembles and Trees (15 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• Decision trees and random forests • Gradient Boosting (XGBoost, LightGBM) • Stacking and voting	• Optimize hyperparameters with GridSearch [Usar] • Visualize decision trees [Evaluar]
Readings : [HTF09], [Gér22]	

Unit 4: Basic Neural Networks (15 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• Multilayer perceptrons (MLP) • Backpropagation and optimizers • Introduction to Keras/TensorFlow	• Build simple neural networks [Usar] • Monitor training with TensorBoard [Evaluar]
Readings : [GBC16], [Cho21]	

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

FI201FCCS. Computational Physics (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: FI201FCCS. Computational Physics
2.2 Semester	: 6 th 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	: FI101FCCS. Physics I. (2 nd 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)

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Usage)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Usage)

AG-C11) Tool Usage: Applies modern computing tools in problem-solving. (Usage)

7. TOPICS

Unit 1: Optics and Light Propagation (10 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
• Nature of light. • Reflection and refraction. • Lenses and mirrors. • Interference and diffraction. • Illumination models (e.g., Phong, Blinn-Phong).	• Describe the properties of light and its propagation. [Familiarizarse] • Apply the laws of reflection and refraction. [Usar] • Implement illumination models in computer graphics. [Evaluar]
Readings : [YF18], [Hec17]	

Unit 2: Collisions and Energy Transfer (8 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
• Impulse and linear momentum. • Elastic and inelastic collisions. • Conservation of energy in collisions. • Deformation of elastic meshes (e.g., mass-spring model).	• Apply the principles of conservation of linear momentum and energy in collisions. [Familiarizarse] • Model the deformation of elastic meshes due to impact. [Usar] • Implement collision simulations in a computational environment. [Evaluar]
Readings : [YF18], [Tay05]	

Unit 3: Rigid Body Mechanics (8 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
• Rotation of rigid bodies. • Moment of inertia. • Torque and rotational kinetic energy.	• Describe the rotation of rigid bodies. [Familiarizarse] • Calculate the moment of inertia. [Usar] • Apply the equations of rotational dynamics. [Evaluar]
Readings : [YF18], [Tay05]	

Unit 4: Fluid Dynamics (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Basic principles of fluid dynamics. • Viscosity. • Laminar and turbulent flow.	• Describe the properties of fluids. [Familiarizarse] • Explain the concepts of viscosity and laminar/turbulent flow. [Usar] • Solve simple fluid dynamics problems. [Evaluar]
Readings : [YF18]	

Unit 5: Thermodynamics (6 hours)	
Competences Expected: 1,6	
Topics	Learning Outcomes
• Laws of thermodynamics. • Heat transfer.	• State the laws of thermodynamics. [Familiarizarse] • Describe the mechanisms of heat transfer. [Usar]
Readings : [YF18]	

Unit 6: Physical Simulation (10 hours)	
Competences Expected: 1,6,AG-C12	
Topics	Learning Outcomes
• Numerical methods for physical simulation. • Verlet integration. • Collision detection. • Particle systems. • Constraints and solvers.	• Implement basic numerical methods for physical simulation. [Familiarizarse] • Use Verlet integration to simulate motion. [Usar] • Implement collision detection algorithms. [Evaluar]
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.

Chapter 7

Seventh Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS251-CS2023. Computer graphics (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS251-CS2023. Computer graphics
2.2 Semester	: 7 th 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	: MA106FCCS. Numerical Methods. (4 th 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

- 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)

- 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)

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

Unit 1: Conceptos Fundamentales (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual. • Soluciones de compensación entre el almacenamiento de datos y los datos re-computing es personalizado por vectores y raster en representaciones de imágenes. • Modelos de color sustractivo Aditivo y (CMYK y RGB) y por qué estos proporcionan una gama de colores. • Animación como una secuencia de imágenes fijas.	• 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] • Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse] • 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] • 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]
Readings : [HB90]	

Unit 2: Rendering Básico (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Renderizado en la naturaleza, por ejemplo, la emisión y dispersión de la luz y su relación con la integración numérica. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización) • Radiometría básica, triángulos similares y modelos de proyecciones • Afinamiento y Transformaciones de Sistemas de coordenadas • <i>Ray tracing</i> • Visibilidad y oclusión, incluyendo soluciones a este problema, como el almacenamiento en búfer de profundidad, algoritmo del pintor, y el trazado de rayos. • Rasterización triangular simple. • Renderización con una API basada en shader. • Aplicación de la representación de estructuras de datos espaciales. • Muestreo y anti-aliasing. • Renderizado Forward and Backward (i.e., <i>ray-casting</i> y rasterización)	• 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] • Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse] • Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar] • Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afin [Usar] • 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] • Contrastar la renderización hacia adelante <i>forward</i> y hacia atrás <i>backward</i> [Evaluar] • Explicar el concepto y las aplicaciones de mapeo de texturas, muestreo y el <i>anti-aliasing</i> [Familiarizarse] • Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse] • Implementar un sencillo renderizador en tiempo real utilizando una API de rasterización (por ejemplo, OpenGL) utilizando buffers de vértices y <i>shaders</i> [Usar] • Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar] • Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar]
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 3: Programación de Sistemas Interactivos (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Manejo de eventos e interacción de usuario. • Enfoques para el diseño, implementación y evaluación de la interacción sin mouse – Interfaces táctiles y multitáctiles. – Interfaces compartidas, incorporadas y grandes – Nuevas modalidades de entrada (tales como datos de sensores y localización) – Nuevas ventanas, por ejemplo, iPhone, Android – Reconocimiento de voz y procesamiento del lenguaje natural – Interfaces utilizables y tangibles – Interacción persuasiva y emoción – Tecnologías de interacción ubicuas y contextuales (UbiComp) – Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada) – Visualización e interacción de ambiente / periféricos	• Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Evaluar]
Readings : [HB90]	

Unit 4: Modelado Geométrico (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Operaciones geométricas básicas como cálculo de intersección y pruebas de proximidad. • Volúmenes, voxels y representaciones basadas en puntos. • Curvas polinomiales y Superficies paramétricas. • Representación implícita de curvas y superficies. • 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. • 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. • Técnicas de subdivisión espacial. • Modelos procedimentales como fractales, modelamiento generativo y sistemas L. • Modelos deformables de forma libre y elásticamente deformables. • Subdivisión de superficies. • Modelado multiresolución. • Reconstrucción. • Representación de Geometría Sólida Constructiva (GSC)	• Representar curvas y superficies utilizando formas tanto implícitas y paramétricas [Usar] • Crear modelos poliédrico simples por teselación de superficies [Usar] • Generar un modelo fractal o terreno usando un método de procedimiento [Usar] • Generar una malla de un conjunto de puntos adquiridos por un scanner laser [Usar] • Construct modelos de geometría sólida constructiva a partir de simples primitivas, tales como cubos y superficies cuádricas [Usar] • Contrastar métodos de modelización con respecto a espacio y tiempo de complejidad y calidad de imagen [Evaluar]
Readings : [HB90], [Shr+13]	

Unit 5: Renderizado Avanzado (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Tiempo (desenfoco de movimiento), la posición del objetivo (enfoco), y la frecuencia continua (color) y su impacto en la representación. • Mapeo de Sombras. • Selectiva de oclusión. • Dispersión de la Superficie. • Renderizado no fotorealístico. • Arquitectura del GPU. • Sistemas visuales humanos incluida la adaptación a la luz, la sensibilidad al ruido, y la fusión de parpadeo.	• Demostrar como un algoritmo calcula una solución a la ecuación de renderización [Evaluar] • Demostrar las propiedades de un algoritmo de renderización, por ejemplo, completo, consistente, e imparcial [Evaluar] • 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] • Discutir como una técnica artística particular puede ser implementada en un renderizador [Familiarizarse] • Explicar como reconocer las técnicas gráficas usadas para crear una imagen en particular [Familiarizarse]
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 6: Animación por computadora (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Cinématica directa e inversa. • Detección de colisiones y respuesta. • Animación procedimental empleando ruido, reglas (boids/crowds) y sistemas de partículas. • Algoritmos Skinning. • 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. • Animación de Cuadros Principales • Splines • Estructuras de datos para rotaciones, como cuaterniones. • Animación de Cámara. • Captura de Movimiento.	• Calcular la orientación de partes articuladas de un modelo de una localización y orientación usando un enfoque de cinemática inversa [Usar] • Implementar el método de interpolación <i>spline</i> para producir las posiciones y orientaciones en medio [Usar] • Implementar algoritmos para el modelamiento físico de partículas dinámicas usando simplemente la mecánica de Newton, por ejemplo Witkin & Kass, serpientes y gusanos, Euler simpléctica, Stormer/Verlet, o métodos de punto medio de Euler [Usar] • 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] • Usar el software de animación común para construir formas orgánicas simples usando <i>metaball</i> y el esqueleto [Usar]
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 2026-I

1. COURSE

CS292-CS2023. Software Engineering II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS292-CS2023. Software Engineering II
2.2 Semester	: 7 th 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 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Modern software development requires mastery of DevSecOps, infrastructure as code, and SRE practices. This course prepares students to design, build, and operate reliable and secure systems in cloud-native environments, integrating automation, observability, and security throughout the software lifecycle.

5. GOALS

- Master complete DevSecOps pipelines integrating build, testing, security, and automated deployments with modern tools.
- Implement secure infrastructure as code using Terraform with policy as code and software supply chain security practices.
- Design and deploy containerized applications in Docker and Kubernetes with observability and SRE practices for high reliability.
- Apply agile methodologies with professional Git, integrated CI/CD, and team metrics for effective collaborative development.
- Integrate NIST SSDF and SLSA standards in the software lifecycle ensuring traceability and compliance from development to production.

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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 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)

- 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. (Familiarity)

7. TOPICS

Unit 1: Métodos de Desarrollo (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Comprensión de programas. • Exactitud del programa. – Tipos de error (sintaxis, logica, tiempo de ejecución) – El concepto de la especificación – Programación defensiva (ejem., codificación segura, manejo de excepciones) – Revision de código – Fundamentos de <i>testing</i> y generación de casos de prueba – El rol y el uso de contratos, incluyendo pre- y post- condiciones – Pruebas unitarias • Refactorización simple. • Entornos modernos de programación: – Búsqueda de código. – Programación usando librería de componentes y sus APIs. • Estrategias de depuración. • Documentación y estilo de programas.	
Readings : [Kim+16], [Mor20], [FHK21]	

Unit 2: Conceptos Fundamentales de Programación (10 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Sintaxis y semántica básica de un lenguaje de alto nivel. • Variables y tipos de datos primitivos (ej., numeros, caracteres, booleanos) • Expresiones y asignaciones. • Operaciones básicas I/O incluyendo archivos I/O. • Estructuras de control condicional e iterativas. • Paso de funciones y parámetros. • Concepto de recursividad.	
Readings : [Bur18], [MK18], [LF20]	

Unit 3: Métodos de Desarrollo (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Comprensión de programas. • Exactitud del programa. – Tipos de error (syntaxis, logica, tiempo de ejecución) – El concepto de la especificacion – Programación defensiva (ejem., codificación segura, manejo de excepciones) – Revision de codigo – Fundamentos de <i>testing</i> y generación de casos de prueba – El rol y el uso de contratos, incluyendo pre- y post- condiciones – Pruebas unitarias • Refactorización simple. • Entornos modernos de programación: – Búsqueda de código. – Programación usando libreria de componentes y sus APIs. • Estrategias de depuración. • Documentación y estilo de programas.	
Readings : [Bey+16], [Bey+18], [AR12]	

Unit 4: Introducción a redes (12 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Técnicas de Switching (por ejemplo, de circuitos, de paquetes) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Principios de capas (encapsulación, multiplexación) • Roles de las diferentes capas (aplicación, transporte, red, enlace de datos, física)	
Readings : [Bur18], [BF16], [HBB17]	

Unit 5: Principios de Sistemas Operativos (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Métodos de estructuración (monolítico, capas, modular, los modelos micro-kernel) • Abstracciones, procesos y recursos. • Los conceptos de interfaces de programa de aplicación (API) • La evolución de las técnicas de hardware / software y las necesidades de aplicación • Organización de dispositivos. • Interrupciones: métodos e implementaciones. • Concepto de usuario de estado / sistema y la protección, la transición al modo kernel.	
Readings : [Lov10], [Ker10], [TB15]	

Unit 6: Introducción (10 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Visión general de plataformas (ejemplo, Web, Mobil, Juegos, Industrial) • Programación a través de APIs específicos. • Visión general de lenguajes de plataforma (ejemplo, Objective C, HTML5) • Programación bajo restricciones de plataforma.	
Readings : [Kle17], [SF13], [Fow12]	

Unit 7: Principios de Diseño Seguro (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Menor privilegio y aislamiento. • Valores predeterminados a prueba de fallos. • Diseño abierto. • La seguridad de extremo a extremo. • La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas) • Diseño de seguridad. • Las tensiones entre la seguridad y otros objetivos de diseño. • Mediación completa. • El uso de componentes de seguridad vetados. • Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque) • Seguridad utilizable. • Componibilidad de seguridad. • Prevención, detección y disuasión.	
Readings : [Smi21], [Sho14], [HL06]	

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

- [HL06] Michael Howard and David LeBlanc. *Writing Secure Code*. 2nd. Microsoft Press, 2006.
- [Ker10] Michael Kerrisk. *The Linux Programming Interface*. No Starch Press, 2010.
- [Lov10] Robert Love. *Linux System Programming: Talking Directly to the Kernel and C Library*. 2nd. O'Reilly Media, 2010.
- [AR12] John Allspaw and Jesse Robbins. *Web Operations: Keeping the Data On Time*. O'Reilly Media, 2012.
- [Fow12] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2012.
- [SF13] Pramod Sadalage and Martin Fowler. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 2013.
- [Sho14] Adam Shostack. *Threat Modeling: Designing for Security*. Wiley, 2014.

-
- [TB15] Andrew Tanenbaum and Herbert Bos. *Modern Operating Systems*. 4th. Pearson, 2015.
- [Bey+16] Betsy Beyer et al. *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016.
- [BF16] Matt Butcher and Matt Farina. *Docker in Practice*. 2nd. Manning Publications, 2016.
- [Kim+16] Gene Kim et al. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.
- [HBB17] Kelsey Hightower, Brendan Burns, and Joe Beda. *Kubernetes: Up and Running*. O'Reilly Media, 2017.
- [Kle17] Martin Kleppmann. *Designing Data-Intensive Applications*. O'Reilly Media, 2017.
- [Bey+18] Betsy Beyer et al. *The Site Reliability Workbook: Practical Ways to Implement SRE*. O'Reilly Media, 2018.
- [Bur18] Brendan Burns. *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services*. O'Reilly Media, 2018.
- [MK18] Karl Matthias and Sean Kane. *Docker: Up and Running*. 2nd. O'Reilly Media, 2018.
- [LF20] James Lewis and Martin Fowler. *Microservices: A Definition of This New Architectural Term*. Available online. ThoughtWorks, 2020.
- [Mor20] Kief Morris. *Infrastructure as Code: Dynamic Systems for the Cloud Age*. 2nd. O'Reilly Media, 2020.
- [FHK21] Nicole Forsgren, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press, 2021.
- [Smi21] Richard Smith. *DevSecOps: A Leader's Guide to Producing Secure Software without Compromising Speed, Agility, and Innovation*. Apress, 2021.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS2H1-CS2023. User Experience (UX) (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS2H1-CS2023. User Experience (UX)
2.2 Semester	: 7 th 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-CS2023. Software Engineering I. (5 th 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

- 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)

AG-C09 Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04 Communication: Communicates effectively in complex computing activities. (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)

7. TOPICS

Unit 1: Fundamentos (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Contextos para IHC (cualquiera relacionado con una interfaz de usuario, p.e., página web, aplicaciones de negocios, aplicaciones móviles y juegos) - Heurística de usabilidad y los principios de pruebas de usabilidad. - Procesos para desarrollo centrado en usuarios, p.e., enfoque inicial en usuarios, pruebas empíricas, diseño iterativo. - Principios del buen diseño y buenos diseñadores; ventajas y desventajas de ingeniería. - Diferentes medidas para evaluación, p.e., utilidad, eficiencia, facilidad de aprendizaje, satisfacción de usuario.	- Discutir por qué el desarrollo de software centrado en el hombre es importante [Familiarizarse] - 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] - Resumir los preceptos básicos de la interacción psicológica y social [Familiarizarse] - Desarrollar y usar un vocabulario conceptual para analizar la interacción humana con el software: disponibilidad, modelo conceptual, retroalimentación, y demás [Familiarizarse]
Readings : [Dix+04], [Sto+05], [RS11]	

Unit 2: Factores Humanos (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Modelos cognoscitivos que informan diseño de interacciones, p.e., atención, percepción y reconocimiento, movimiento, memoria, golfos de expectativa y ejecución. - Capacidades físicas que informan diseño de interacción, p.e. percepción del color, ergonomía. - Accesibilidad, p.e., interfaces para poblaciones con diferentes habilidades (p.e., invidentes, discapacitados) - Interfaces para grupos de población de diferentes edades (p.e., niños, mayores de 80)	Crear y dirigir una simple prueba de usabilidad para una aplicación existente de software [Familiarizarse]
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unit 3: Diseño y Testing centrados en el usuario (16 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Enfoque y características del proceso de diseño. • Requerimientos de funcionalidad y usabilidad. • Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual. • Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas. • Análisis de tareas, incluidos los aspectos cualitativos de la generación de modelos de análisis de tareas. • Consideración de IHC como una disciplina de diseño: – Sketching – Diseño participativo – Sketching – Diseño participativo • Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página. • Prototipos de baja fidelidad (papel) • Técnicas de evaluación cuantitativa ej. evaluación Keystroke-level. • Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar. • Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación. • Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización. • Reportar los resultados de las evaluaciones. • Internacionalización, diseño para usuarios de otras culturas, intercultural.	• Llevar a cabo una evaluación cuantitativa y discutir / informar sobre los resultados [Familiarizarse] • Para un grupo de usuarios determinado, realizar y documentar un análisis de sus necesidades [Familiarizarse] • Discutir al menos un standard nacional o internacional de diseño de interfaz de usuario [Familiarizarse] • Explicar cómo el diseño centrado en el usuario complementa a otros modelos de proceso software [Familiarizarse] • Utilizar <i>lo-fi</i> (baja fidelidad) técnicas de prototipado para recopilar y reportar, las respuestas del usuario [Usar] • Elegir los métodos adecuados para apoyar el desarrollo de una específica interfaz de usuario [Evaluar] • Utilizar una variedad de técnicas para evaluar una interfaz de usuario dada [Evaluar] • Comparar las limitaciones y beneficios de los diferentes métodos de evaluación [Evaluar]
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unit 4: Diseño de Interacción (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Principios de interfaces gráficas de usuario (GUIs) • Elementos de diseño visual (disposición, color, fuentes, etiquetado) • Manejo de fallas humanas/sistema. • Estándares de interfaz de usuario. • Presentación de información: navegación, representación, manipulación. • Técnicas de animación de interfaz (ej. grafo de escena) • Clases Widget y bibliotecas. • Internacionalización, diseño para usuarios de otras culturas, intercultural. • Elección de estilos de interacción y técnicas de interacción.	• Crear una aplicación simple, junto con la ayuda y la documentación, que soporta una interfaz gráfica de usuario [Usar]
Readings : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unit 5: Nuevas Tecnologías Interactivas (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Elección de estilos de interacción y técnicas de interacción. • Enfoques para el diseño, implementación y evaluación de la interacción sin mouse – Interfaces táctiles y multitáctiles. – Interfaces compartidas, incorporadas y grandes – Nuevas modalidades de entrada (tales como datos de sensores y localización) – Nuevas ventanas, por ejemplo, iPhone, Android – Reconocimiento de voz y procesamiento del lenguaje natural – Interfaces utilizables y tangibles – Interacción persuasiva y emoción – Tecnologías de interacción ubicuas y contextuales (UbiComp) – Inferencia bayesiana (por ejemplo, texto predictivo, orientación guiada) – Visualización e interacción de ambiente / periféricos • Salida: – Sonido – Visualización estereoscópica – Forzar la simulación de retroalimentación, dispositivos hápticos • Arquitectura de Sistemas: – Motores de Juego – Realidad Aumentada móvil – Simuladores de vuelo – CAVEs – Imágenes médicas	• Describe cuando son adecuadas las interfaces sin uso de ratón [Familiarizarse] • Comprende las posibilidades de interacción que van más allá de las interfaces de ratón y puntero [Familiarizarse] • Discute las ventajas (y desventajas) de las interfaces no basadas en ratón [Usar] • Describir el modelo óptico realizado por un sistema de gráficos por computadora para sintetizar una visión estereoscópica [Familiarizarse] • Describir los principios de las diferentes tecnologías de seguimiento de espectador [Familiarizarse] • Determinar los requerimientos básicos en interfaz, software, hardware, y configuraciones de software de un sistema VR para una aplicación específica [Evaluar]
Readings : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unit 6: Colaboración y Comunicación (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• La comunicación asíncrona en grupo, por ejemplo, el correo electrónico, foros, redes sociales. • Medios de comunicación social, informática social, y el análisis de redes sociales. • Colaboración en línea, espacios "inteligentes" y aspectos de coordinación social de tecnologías de flujo de trabajo. • Comunidades en línea. • Personajes de Software y agentes inteligentes, mundos virtuales y avatares. • Psicología Social	• Describir la diferencia entre la comunicación sincrónica y asíncrona [Familiarizarse] • Comparar los problemas de IHC en la interacción individual con la interacción del grupo [Familiarizarse] • Discuta varias problemas de interés social planteados por el software colaborativo [Usar] • Discutir los problemas de IHC en software que personifica la intención humana [Evaluar]
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 2026-I

1. COURSE

AI264. Deep Learning (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: AI264. Deep Learning
2.2 Semester	: 7 th 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	: AI263. Introduction to Machine Learning. (6 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course covers modern deep learning fundamentals, including convolutional networks, transformer architectures, and training techniques for advanced AI systems.

5. GOALS

- Implement deep neural networks using modern frameworks (PyTorch/TensorFlow)
- Design architectures for specific domains (vision, language, etc.)
- Analyze ethical limitations of deep models

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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

AG-C10) Investigation: Studies complex computing problems using information science methods. (Usage)

7. TOPICS

Unit 1: DL Foundations (15 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• Modern backpropagation • Regularization (Dropout, BatchNorm) • Advanced optimizers	• Implement DNNs from scratch [Evaluar] • Diagnose training problems [Usar]
Readings : [GBC16]	

Unit 2: Advanced Architectures (25 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
• ResNets and deep CNNs • Transformers and attention • Generative models (VAEs, GANs)	• Adapt architectures to specific domains [Evaluar] • Train transformers on real datasets [Usar]
Readings : [Vas+17], [He+16]	

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

- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [He+16] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CVPR* (2016), pp. 770–778.
- [Vas+17] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems* 30 (2017).



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

BI101FCCS. Biology (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: BI101FCCS. Biology
2.2 Semester	: 7 th 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

Biology, the science of life, provides a foundational framework for interdisciplinary fields such as bioinformatics, computational biology, and biologically inspired artificial intelligence. This course offers an introduction to the essential concepts of biology, ranging from the cell to evolution, with particular emphasis on computational applications and perspectives.

5. GOALS

- Develop an understanding of the fundamental principles of cellular and molecular biology.
- Examine the core processes of life, including DNA replication, transcription, and translation.
- Recognize the intersections between biology and computing in fields such as bioinformatics.
- Acquire the ability to analyze biological data using both qualitative and quantitative approaches.
- Cultivate interdisciplinary thinking that bridges biology and computer science.

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

Unit 1: Introduction to Biology and Life (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Characteristics of living organisms and definitions of life. • Levels of biological organization (from molecules to ecosystems). • The scientific method in biological research. • Computational models of living systems.	• Articulate the defining characteristics of living organisms. [Familiarizarse] • Identify and distinguish the levels of biological organization. [Usar] • Apply the scientific method within a biological context. [Evaluar] • Explain computational models of ecosystems. [Usar]
Readings : [Urr+17], [Rav+17]	

Unit 2: The Chemistry of Life (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Atoms, molecules, and chemical bonding. • Major classes of biomolecules: proteins, carbohydrates, lipids, nucleic acids. • Parallels with data structures in computer science. • DNA sequences as computational representations.	• Identify the major classes of biomolecules and describe their functions. [Familiarizarse] • Explain relationships between biological structures and computational data structures. [Usar] • Analyze biological sequences from a computational perspective. [Evaluar]
Readings : [Urr+17], [Rav+17]	

Unit 3: Cell Structure and Function (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Prokaryotic and eukaryotic cell types. • Cellular organelles and their functions. • Membrane structure and transport mechanisms. • Biological networks and computational modeling.	• Compare and contrast prokaryotic and eukaryotic cells. [Familiarizarse] • Describe the structure and function of key organelles. [Usar] • Explain mechanisms of membrane transport. [Evaluar] • Introduce biological networks and approaches to their computational modeling. [Usar]
Readings : [Urr+17], [Rav+17]	

Unit 4: Metabolism and Energy (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Photosynthesis and cellular respiration. • Enzymes and biological catalysis. • Analogies with optimization algorithms. • Energy efficiency in biological and computational systems.	• Explain the processes of photosynthesis and cellular respiration. [Familiarizarse] • Describe the role of enzymes in metabolism. [Usar] • Draw parallels between biological and computational approaches to energy efficiency. [Evaluar]
Readings : [Urr+17], [Rav+17]	

Unit 5: Fundamentals of Genetics (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• DNA: structure and replication. • RNA: transcription and translation. • Principles of Mendelian inheritance. • Applications in genetic algorithms.	• Describe the structure and replication of DNA. [Familiarizarse] • Explain the processes of transcription and translation. [Usar] • Demonstrate understanding of Mendelian inheritance. [Evaluar] • Apply genetic principles to computational algorithms. [Usar]
Readings : [Urr+17], [Rav+17]	

Unit 6: Evolution and Biological Diversity (6 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Theory of evolution by natural selection. • Evidence supporting evolutionary theory. • Biodiversity and systems of biological classification. • Computational simulations of evolutionary processes.	• Explain the theory of evolution by natural selection. [Familiarizarse] • Describe key lines of evidence supporting evolution. [Usar] • Analyze mechanisms of evolutionary change. [Evaluar] • Apply evolutionary concepts through computational simulations. [Usar]
Readings : [Urr+17], [Rav+17]	

Unit 7: Ecology: Principles and Applications (8 hours)	
Competences Expected: 1,AG-C07	
Topics	Learning Outcomes
• Interactions among organisms. • Biogeochemical cycles. • Population dynamics. • Computational modeling of ecological systems.	• Describe fundamental ecological interactions. [Familiarizarse] • Explain the role of biogeochemical cycles. [Usar] • Analyze population dynamics. [Evaluar] • Employ computational modeling in addressing ecological problems. [Usar]
Readings : [Urr+17], [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

- [Rav+17] Peter H. Raven et al. *Biology*. McGraw-Hill Education, 2017.
- [Urr+17] Lisa A. Urry et al. *Campbell Biology*. Pearson, 2017.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

FG211-ACM. Professional Ethics (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: FG211-ACM. Professional Ethics
2.2 Semester	: 7 th 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	: None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course adapts professional ethical principles to Artificial Intelligence, integrating ACM/IEEE codes with AI-specific challenges. Combines regulatory frameworks (GDPR) with current societal impact cases.

5. GOALS

- Apply ACM/IEEE ethics codes to AI problems.
- Analyze compliance with regulations (GDPR) in autonomous systems.
- Design ethical protocols for AI projects.

6. COMPETENCES

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

7. TOPICS

Unit 1: Ethical Foundations in AI (10 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• Comparison: ACM vs IEEE ethics codes for AI. • Algorithmic transparency and fairness principles.	• Explain differences between ethical frameworks [Familiarizarse]. • Evaluate compliance in real cases [Evaluar].
Readings : [ACM18], [IEE20]	

Unit 2: Privacy and Regulations (14 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• GDPR applied to ML: right to explanation. • Ethical auditing of datasets (e.g., ImageNet).	• Implement GDPR checklists for models [Usar]. • Identify violations in case studies [Evaluar].
Readings : [VB18], [Mül21]	

Unit 3: Bias and Fairness (18 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• Fairness metrics (demographic parity, equality of opportunity). • Tools: IBM Fairness 360, Google What-If.	• Measure biases in models using Python [Usar]. • Propose mitigation strategies [Evaluar].
Readings : [For20], [Meh+21]	

Unit 4: Accountability in Autonomous Systems (18 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• Legal attribution in AI failures (e.g., self-driving cars). • ISO standards for trustworthy AI.	• Draft liability clauses [Evaluar]. • Analyze legal conflicts [Familiarizarse].
Readings : [Bos14], [Com19]	

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

- [Bos14] Nick Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.
- [ACM18] ACM. *ACM Code of Ethics and Professional Conduct*. Tech. rep. 2018. URL: <https://www.acm.org/code-of-ethics>.
- [VB18] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 2018.
- [Com19] IEEE Ethics Committee. *Ethical Dilemmas in Engineering*. Tech. rep. 2019. URL: <https://www.ieee.org/ethics>.
- [For20] ACM Ethics Task Force. *Case Studies in Computing and Society*. Tech. rep. 2020. URL: <https://ethics.acm.org/case-studies/>.
- [IEE20] IEEE. *IEEE Code of Ethics*. Tech. rep. 2020. URL: <https://www.ieee.org/about/corporate/governance/p7-8.html>.

-
- [Meh+21] Ninareh Mehrabi et al. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Computing Surveys* 54.6 (2021). doi: 10.1145/3457607.
- [Mül21] Vincent C. Müller. *Ethics of Artificial Intelligence and Robotics*. Cambridge University Press, 2021.

Chapter 8

Eighth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS281-CS2023. Computing in Society (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS281-CS2023. Computing in Society
2.2 Semester	: 8 th Semester
2.3 Credits	: 2
2.4 Horas	: 2 HT;
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

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

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

- 3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04) Communication: Communicates effectively in complex computing activities. (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. (Familiarity)

7. TOPICS

Unit 1: Historia (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Pre-historia – El mundo antes de 1946. • Historia del hardware, software, redes. • Pioneros de la Computación. • Historia de Internet.	• Identificar importantes tendencias en la historia del campo de la computación [Familiarizarse] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Familiarizarse]
Readings : [LL04], [McL00]	

Unit 2: Contexto Social (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Implicancias sociales de la computación en un mundo conectado en red. • Impacto de los medios sociales en el individualismo, colectivismo y en la cultura. • Crecimiento y control de la Internet • 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. • Los problemas de accesibilidad, incluyendo los requisitos legales. • Computación consciente del contexto.	• 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] • 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] • Interpretar el contexto social de un determinado diseño y su aplicación [Evaluar] • Evaluar la eficacia de un diseño y aplicación dada a partir de datos empíricos [Familiarizarse] • Resumir las implicaciones de los medios sociales en el individualismo frente al colectivismo y la cultura [Familiarizarse] • 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] • 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] • 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] • Explicar las consecuencias de la sensibilidad al contexto en los sistemas de computación ubicua [Familiarizarse]
Readings : [LL04], [McL00]	

Unit 3: Herramientas de Análisis (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Argumentación ética. • Teorías éticas y toma de decisiones. • Suposiciones morales y valores.	• Evaluar las posiciones de las partes interesadas en una situación dada [Familiarizarse] • Analizar errores lógicos básicos en una discusión [Usar] • Analizar un argumento para identificar premisas y la conclusión [Familiarizarse] • Ilustrar el uso de ejemplo y analogía en el argumento ético [Familiarizarse] • Evaluar compensaciones éticos / sociales en las decisiones técnicas [Familiarizarse]
Readings : [LL04], [McL00]	

Unit 4: Ética Profesional (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Community values and the laws by which we live. • La naturaleza del profesionalismo incluido el cuidado, la atención y la disciplina, la responsabilidad fiduciaria y mentoría. • 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. • La certificación profesional, códigos de ética, conducta y práctica, como la ACM / IEEE-CS, SE, AITP, IFIP y las sociedades internacionales. • 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) • El papel del profesional de de computación en las políticas públicas. • Mantenimiento de la conciencia en relación a las consecuencias. • Disidencia ética y la denuncia de irregularidades. • La relación entre la cultura regional y dilemas éticos. • Tratar con el acoso y la discriminación. • Formas de credenciamiento profesional. • Políticas de uso aceptable para la computación en el lugar de trabajo. • Ergonomía y entornos de trabajo computacionales saludables. • Consideraciones a tiempos de entrega de mercado vs estándares de calidad profesional.	• Identificar los problemas éticos que se plantean en el desarrollo de software y determinar cómo abordarlos técnica y éticamente [Usar] • Explicar la responsabilidad ética de velar por la corrección de software, confiabilidad y seguridad [Evaluar] • Describir los mecanismos que normalmente existen para que profesional se mantenga al día [Familiarizarse] • Describir las fortalezas y debilidades de códigos profesionales relevantes como expresiones de profesionalismo y guías para la toma de decisiones [Familiarizarse] • Analizar un problema mundial de computación, observando el papel de los profesionales y funcionarios del gobierno en el manejo de este problema [Familiarizarse] • Evaluar los códigos de ética profesional de la ACM, la Sociedad de Computación de la IEEE, y otras organizaciones [Familiarizarse] • Describir las formas en que los profesionales pueden contribuir a las políticas públicas [Familiarizarse] • Describir las consecuencias de la conducta profesional inadecuada [Usar] • Identificar las etapas progresivas en un incidente de denuncia de irregularidades [Usar] • Identificar ejemplos de cómo interactúa la cultura regional con dilemas éticos [Familiarizarse] • Investigar las formas de acoso, discriminación y formas de ayuda [Usar] • Examine las diversas formas de acreditación de profesionales [Usar] • Explicar la relación entre la ergonomía en los ambientes y la salud de las personas de computación [Usar] • Desarrollar un uso del computador/política de uso aceptable con medidas coercitivas [Familiarizarse] • 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]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 5: Propiedad Intelectual (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Fundamentos filosóficos de propiedad intelectual. • Derechos de propiedad intelectual. • Propiedad intelectual digital intangible (IDIP). • Fundamentos legales para protección de la propiedad intelectual. • Gestión de derechos digitales. • Copyrights, patentes, secretos de comercio, marcas registradas. • Plagiarismo. • Fundamentos del movimiento Open Source. • Piratería de Software.	• Discute la racionalidad de la protección legal de la propiedad intelectual [Evaluar] • Discute las bases filosóficas de la propiedad intelectual [Familiarizarse] • Describe la legislación orientada a los delitos de derechos de autor digitales [Evaluar] • Critica la legislación orientada a los delitos digitales de derechos de autor [Familiarizarse] • Identifica ejemplos contemporáneos de propiedad intelectual digital intangible [Evaluar] • Justifica el uso de material con derechos de autor [Evaluar] [Familiarizarse] • Evalúa los asuntos éticos inherentes a diversos mecanismos de detección de plagio [Familiarizarse] • Interpreta el intento y la implementación de licencias de software [Familiarizarse] • Discute asuntos que involucran la seguridad de patentes en software [Familiarizarse] • Caracteriza y contrasta los conceptos de derechos de autor, patentes y de marcas comerciales [Familiarizarse] • Identifica los objetivos del movimiento de software libre [Evaluar] • Identifica los objetivos del movimiento de software libre [Familiarizarse]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 6: Privacidad y Libertades Civiles (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Fundamentos filosóficos de derechos de privacidad. • Fundamentos legales de protección de privacidad. • 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. • Ramificaciones de privacidad diferencial. • Soluciones basadas en la tecnología para la protección de la privacidad. • Legislación de privacidad en áreas de práctica. • Libertades civiles y diferencias culturales. • Libertad de expresión y sus limitaciones.	• Discute las bases filosóficas para la protección legal de la privacidad personal [Familiarizarse] • Evalúa soluciones para amenazas a la privacidad en bases de datos transaccionales y almacenes de datos [Familiarizarse] • 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] • Describe las ramificaciones de la privacidad diferenciada [Familiarizarse] • Investiga el impacto de soluciones tecnológicas a los problemas de privacidad [Familiarizarse] • Critica la intención, el valor potencial y la implementación de las diversas formas de legislación en privacidad [Familiarizarse] • Identifica estrategias que permitan la apropiada libertad de expresión [Familiarizarse]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 7: Políticas de seguridad, Leyes y crímenes computacionales (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Ejemplos de delitos informáticos y reparación legal para delincuentes informáticos. • Ingeniería social, robo de identidad y recuperación. • Tópicos relacionados al uso de acceso indebido y las infracciones y materia de seguridad. • Motivaciones y ramificaciones del ciberterrorismo y el hacking criminal, cracking. • Efectos de malware, como virus, worms y Trojan horses. • Estrategias de prevención de Crimen. • Políticas de Seguridad.	• Listar ejemplos clásicos de delitos informáticos y incidentes de ingeniería social con impacto social [Familiarizarse] • Indentificar leyes que se aplican a delitos informáticos [Familiarizarse] • Describir la motivación y ramificaciones de cyberterrorismo y hackeo criminal [Familiarizarse] • Examinar los problemas éticos y legales relacionados con el mal uso de accesos y diversas violaciones en la seguridad [Familiarizarse] • Discutir el rol del profesional en seguridad y los problemas que están envueltos [Familiarizarse] • 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] • Escribir una politica de seguridad de una empresa, la cual incluye procedimientos para administrar contraseñas y monitorizar a los empleados [Familiarizarse]
Readings : [LL04], [McL00], [Edi09a], [Edi09b], [Edi10]	

Unit 8: Economía de la Computación (2 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Monopolio y sus implicaciones económicas. • Efecto del suministro de mano de obra calificada y la demanda sobre la calidad de los productos de computación. • Estrategias de precio en el dominio de la computación. • El fenómeno del desarrollo de software outsourcing y off-shoring; impactos en el empleo y la economía. • Consecuencias de la globalización para la profesión de Ciencias de la Computación. • Diferencias en acceso a recursos de computación y el posible efecto de los mismos. • Analisis costo/beneficio de trabajos con consideraciones para manufactura, hardware, software e implicaciones de ingeniería. • Costo estimado versus costo actual in relación al costo total. • Emprendimiento: perspectivas y entrapamientos. • Efectos de red o economías de escala del lado de la demanda. • El uso de la ingeniería económica para hacer frente a las finanzas.	• Resumir los fundamentos para los esfuerzos antimonopolio [Familiarizarse] • Identificar diversas maneras en que la industria de la tecnología de la información está afectada por la escasez de la oferta de trabajo [Familiarizarse] • Identificar la evolución de la estrategia de precios para el cálculo de los bienes y servicios [Familiarizarse] • Discutir los beneficios, los inconvenientes y las implicaciones de <i>off-shoring</i> y <i>outsourcing</i> [Familiarizarse] • Investigar y defender maneras de tratar las limitaciones en el acceso a la computación. [Usar] • Describir los beneficios económicos de efectos de la red [Usar]
Readings : [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 2026-I

1. COURSE

CS3I1-CS2023. Computer Security (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS3I1-CS2023. Computer Security
2.2 Semester	: 8 th 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	: CS231-CS2023. Networking and Communication. (6 th 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 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (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)

7. TOPICS

Unit 1: Fundamentos y Conceptos en Seguridad (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• CIA (Confidencialidad, Integridad, Disponibilidad) • Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque . • Autenticación y autorización, control de acceso (vs. obligatoria discrecional) • Concepto de la confianza y la honradez . • Ética (revelación responsable)	• Analizar las ventajas y desventajas de equilibrar las propiedades clave de seguridad(Confidenciabilidad, Integridad, Disponibilidad) [Familiarizarse] • 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] • Explicar los conceptos de autenticación, autorización, control de acceso [Familiarizarse] • Explicar el concepto de confianza y confiabilidad [Familiarizarse] • 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]
Readings : [Stallings2017], [Kurose2017]	

Unit 2: Principios de Diseño Seguro (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Menor privilegio y aislamiento. • Valores predeterminados a prueba de fallos. • Diseño abierto. • La seguridad de extremo a extremo. • La defensa en profundidad (por ejemplo, la programación defensiva, defensa en capas) • Diseño de seguridad. • Las tensiones entre la seguridad y otros objetivos de diseño. • Mediación completa. • El uso de componentes de seguridad vetados. • Economía del mecanismo (la reducción de la base informática de confianza, minimizar la superficie de ataque) • Seguridad utilizable. • Componibilidad de seguridad. • Prevención, detección y disuasión.	• Describir el principio de privilegios mínimos y el aislamiento que se aplican al diseño del sistema [Familiarizarse] • Resumir el principio de prueba de fallos y negar por defecto [Familiarizarse] • Discutir las implicaciones de depender de diseño abierto o secreto de diseño para la seguridad [Familiarizarse] • Explicar los objetivos de seguridad de datos de extremo a extremo [Familiarizarse] • Discutir los beneficios de tener múltiples capas de defensas [Familiarizarse] • Por cada etapa en el ciclo de vida de un producto, describir que consideraciones de seguridad deberían ser evaluadas [Familiarizarse] • Describir el costo y ventajas y desventajas asociadas con el diseño de seguridad de un producto. [Familiarizarse] • Describir el concepto de mediación y el principio de mediación completa [Familiarizarse] • Conocer los componentes estándar para las operaciones de seguridad, en lugar de reinventar las operaciones fundamentales [Familiarizarse] • 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] • Discutir la importancia de la usabilidad en el diseño de mecanismos de seguridad [Familiarizarse] • Describir problemas de seguridad que surgen en los límites entre varios componentes [Familiarizarse] • Identificar los diferentes roles de mecanismos de prevención y mecanismos de eliminación/disuasión [Familiarizarse]
Readings : [Stallings2017]	

Unit 3: Programación Defensiva (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Validación de datos de entrada y sanitización • Elección del lenguaje de programación y lenguajes con tipos de datos seguro. • Ejemplos de validación de entrada de datos y sanitización de errores. – Desbordamiento de búfer – Errores enteros – Inyección SQL – Vulnerabilidad XSS • Las condiciones de carrera. • Manejo correcto de las excepciones y comportamientos inesperados. • Uso correcto de los componentes de terceros. • Desplegar eficazmente las actualizaciones de seguridad. • Información de control de flujo. • Generando correctamente el azar con fines de seguridad. • Mecanismos para la detección y mitigación de datos de entrada y errores de sanitización. • Fuzzing • El análisis estático y análisis dinámico. • Programa de verificación. • Soporte del sistema operativo (por ejemplo, la asignación al azar del espacio de direcciones, canarios) • El soporte de hardware (por ejemplo, el DEP, TPM)	• Explicar por que la validación de entrada y desinfección de datos es necesario en el frente del control contencioso del canal de entrada [Usar] • 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] • Clasificar los errores de validación de entrada común, y escribir correctamente el código de validación de entrada [Usar] • 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] • Demostrar la identificación y el manejo elegante de las condiciones de error [Familiarizarse] • Explique los riesgos de mal uso de las interfaces con código de terceros y cómo utilizar correctamente el código de terceros [Familiarizarse] • Discutir la necesidad de actualizar el software para corregir las vulnerabilidades de seguridad y la gestión del ciclo de vida de la corrección [Familiarizarse]
Readings : [Stallings2017]	

Unit 4: Ataques y Amenazas (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Atacante metas, capacidades y motivaciones (como economía sumergida, el espionaje digital, la guerra cibernética, las amenazas internas, hacktivismo, las amenazas persistentes avanzadas) • Los ejemplos de malware (por ejemplo, virus, gusanos, spyware, botnets, troyanos o rootkits) • Denegación de Servicio (DoS) y Denegación de Servicio Distribuida (DDoS) • Ingeniería social (por ejemplo, perscando) • Los ataques a la privacidad y el anonimato . • El malware / comunicaciones no deseadas, tales como canales encubiertos y esteganografía.	• Describir tipos de ataques similares en contra de un sistema en particular [Familiarizarse] • Discutir los limitantes de las medidas en contra del malware (ejm. detección basada en firmas, detección de comportamiento) [Familiarizarse] • Identificar las instancias de los ataques de ingeniería social y de los ataques de negación de servicios [Familiarizarse] • Discutir como los ataques de negación de servicios puede ser identificados y reducido [Familiarizarse] • Describir los riesgos de la privacidad y del anonimato en aplicaciones comunmente usadas [Familiarizarse] • Discutir los conceptos de conversión de canales y otros procedimientos de filtrado de datos [Familiarizarse]
Readings : [Abdullah2021]	

Unit 5: Seguridad de Red (4 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• 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) • El uso de cifrado de datos y seguridad de la red . • Arquitecturas para redes seguras (por ejemplo, los canales seguros, los protocolos de enrutamiento seguro, DNS seguro, VPN, protocolos de comunicación anónimos, aislamiento) • Los mecanismos de defensa y contramedidas (por ejemplo, monitoreo de red, detección de intrusos, firewalls, suplantación de identidad y protección DoS, honeypots, seguimientos) • Seguridad para redes inalámbricas, celulares . • Otras redes no cableadas (por ejemplo, ad hoc, sensor, y redes vehiculares) • Resistencia a la censura. • Gestión de la seguridad operativa de la red (por ejemplo, control de acceso a la red configure)	• Describir las diferentes categorías de amenazas y ataques en redes [Familiarizarse] • Describir las arquitecturas de criptografía de clave pública y privada y cómo las ICP brindan apoyo a la seguridad en redes [Familiarizarse] • Describir ventajas y limitaciones de las tecnologías de seguridad en cada capa de una torre de red [Familiarizarse] • Identificar los adecuados mecanismos de defensa y sus limitaciones dada una amenaza de red [Usar]
Readings : [Kurose2017], [Ambedkar2015]	

Unit 6: Criptografía (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• 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. • Tipos de cifrado (por ejemplo, cifrado César, cifrado affine), junto con los métodos de ataque típicas como el análisis de frecuencia. • Apoyo a la infraestructura de clave pública para la firma digital y el cifrado y sus desafíos. • Criptografía de clave simétrica: – El secreto perfecto y el cojín de una sola vez – Modos de funcionamiento para la seguridad semántica y encriptación autenticada (por ejemplo, cifrar-entonces-MAC, OCB, GCM) – Integridad de los mensajes (por ejemplo, CMAC, HMAC) • La criptografía de clave pública: – Permutación de trampa, por ejemplo, RSA – Cifrado de clave pública, por ejemplo, el cifrado RSA, cifrado El Gamal – Las firmas digitales – Infraestructura de clave pública (PKI) y certificados – Supuestos de dureza, por ejemplo, Diffie-Hellman, factoring entero • Protocolos de intercambio de claves autenticadas, por ejemplo, TLS . • Primitivas criptográficas: – generadores pseudo-aleatorios y cifrados de flujo – cifrados de bloque (permutaciones pseudo-aleatorios), por ejemplo, AES – funciones de pseudo-aleatorios – funciones de hash, por ejemplo, SHA2, resistencia colisión – códigos de autenticación de mensaje – funciones derivaciones clave	• Describir el propósito de la Criptografía y listar formas en las cuales es usada en comunicación de datos [Familiarizarse] • 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] • Discutir la importancia de los números primos en criptografía y explicar su uso en algoritmos criptográficos [Familiarizarse] • Ilustrar como medir la entropía y como generar aleatoriedad criptográfica [Usar] • Usa primitivas de clave pública y sus aplicaciones [Usar] • Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse] • Discutir protocolos criptográficos y sus propiedades [Familiarizarse]
Readings : [Paar2010]	

Unit 7: Seguridad en la Web (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Modelo de seguridad Web – Modelo de seguridad del navegador incluida la política de mismo origen – Los límites de confianza de cliente-servidor, por ejemplo, no pueden depender de la ejecución segura en el cliente • Gestión de sesiones, la autenticación: – Single Sign-On – HTTPS y certificados • Vulnerabilidades de las aplicaciones y defensas : – Inyección SQL – XSS – CSRF • Seguridad del lado del cliente : – Política de seguridad Cookies – Extensiones de seguridad HTTP, por ejemplo HSTS – Plugins, extensiones y aplicaciones web – Seguimiento de los usuarios Web • Herramientas de seguridad del lado del servidor, por ejemplo, los cortafuegos de aplicación Web (WAFS) y fuzzers	• Describe el modelo de seguridad de los navegadores incluyendo las políticas del mismo origen y modelos de amenazas en seguridad web [Familiarizarse] • 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] • Investigar los tipos comunes de vulnerabilidades y ataques en las aplicaciones web, y defensas contra ellos [Familiarizarse] • Utilice las funciones de seguridad del lado del cliente [Usar]
Readings : [Ambedkar2015]	

Unit 8: Seguridad de plataformas (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Integridad de código y firma de código. • Arranque seguro, arranque medido, y la raíz de confianza. • Testimonio. • TPM y coprocesadores seguros. • Las amenazas de seguridad de los periféricos, por ejemplo, DMA, IOMMU. • Ataques físicos: troyanos de hardware, sondas de memoria, ataques de arranque en frío. • Seguridad de dispositivos integrados, por ejemplo, dispositivos médicos, automóviles. • Ruta confiable.	• Explica el concepto de integridad de código y firma de códigos, así como el alcance al cual se aplica [Familiarizarse] • Discute los conceptos del origen de la confidencialidad y el de los procesos de arranque y carga segura [Familiarizarse] • Describe los mecanismos de arresto remoto de la integridad de un sistema [Familiarizarse] • Resume las metas y las primitivas claves de los modelos de plataforma confiable (TPM) [Familiarizarse] • Identifica las amenazas de conectar periféricos en un dispositivo [Familiarizarse] • Identifica ataques físicos y sus medidas de control [Familiarizarse] • Identifica ataques en plataformas con hardware que no son del tipo PC [Familiarizarse] • Discute los conceptos y la importancia de ruta confiable [Familiarizarse]
Readings : [Stallings2017]	

Unit 9: Investigación digital (Digital Forensics) (5 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• Principios básicos y metodologías de análisis digital forense. • Diseñar sistemas con necesidades forenses en mente. • Reglas de Evidencia - conceptos generales y las diferencias entre las jurisdicciones y la Cadena de Custodia. • Búsqueda y captura de comprobación: requisitos legales y de procedimiento. • Métodos y normas de evidencia digital. • Las técnicas y los estándares para la conservación de los datos. • Cuestiones legales y reportes incluyendo el trabajo como perito. • Investigación digital de los sistema de archivos. • Los forenses de aplicación. • Investigación digital en la web. • Investigación digital en redes. • Investigación digital en dispositivos móviles. • Ataques al computador/red/sistema. • Detección e investigación de ataque. • Contra investigación digital.	• Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse] • Explica como diseñar software de apoyo a técnicas forenses [Familiarizarse] • Describe los requisitos legales para usar datos recuperados [Familiarizarse] • Describe qué es una investigación digital, las fuentes de evidencia digital, y los límites de técnicas forenses [Familiarizarse] • Describe como se realiza la recolección de datos y el adecuado almacenamiento de los datos originales y de la copia forense [Familiarizarse] • Realiza recolección de datos en un disco duro [Usar] • Describe la responsabilidad y obligación de una persona mientras testifica como un examinador forense [Familiarizarse] • Recupera datos basados en un determinado término de búsqueda en una imagen del sistema [Usar] • Reconstruye el historial de una aplicación a partir de los artefactos de la aplicación [Familiarizarse] • Reconstruye el historial de navegación web de los artefactos web [Familiarizarse] • Captura e interpreta el tráfico de red [Familiarizarse] • Discute los retos asociados con técnicas forenses de dispositivos móviles [Familiarizarse]
Readings : [Stallings2017]	

Unit 10: Seguridad en Ingeniería de Software (4 hours)	
Competences Expected: 2,4,AG-C02,AG-C07,AG-C09	
Topics	Learning Outcomes
• La construcción de la seguridad en el ciclo de vida de desarrollo de software. • Principios y patrones de diseño seguros. • Especificaciones de software seguros y requisitos. • Prácticas de desarrollo de software de seguros. • Asegure probar el proceso de las pruebas de que se cumplan los requisitos de seguridad (incluyendo análisis estático y dinámico)	• Describir los requisitos para la integración de la seguridad en el SDL [Familiarizarse] • 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] • Desarrollar especificaciones para un esfuerzo de desarrollo de software que especifica completamente los requisitos funcionales y se identifican las rutas de ejecución esperadas [Familiarizarse]
Readings : [Stallings2017]	

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 2026-I

1. COURSE

CS3P1-CS2023. Parallel and Distributed Computing (Mandatory)

2. GENERAL INFORMATION

- | | | |
|----------------------------|---|--|
| 2.1 Course | : | CS3P1-CS2023. Parallel and Distributed Computing |
| 2.2 Semester | : | 8 th 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 | : | <ul style="list-style-type: none">• CS212-CS2023. Analysis and Design of Algorithms. (5th Sem)• CS231-CS2023. Networking and Communication. (6th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Fundamentos de paralelismo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Procesamiento Simultáneo Múltiple. • Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos) • Paralelismo, comunicación, y coordinación: – Paralelismo, comunicación, y coordinación – Necesidad de Sincronización • Errores de Programación ausentes en programación secuencial: – Tipos de Datos (lectura/escritura simultánea o escritura/escritura compartida) – Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado) – Falta de vida/progreso (deadlock, starvation)	• Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse] • Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse] • Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]
Readings : [Pac11], [Mat14], [Quinn2003Parallel2003Parallel], [HW10]	

Unit 2: Arquitecturas paralelas (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Procesadores mutlinúcleo. • Memoria compartida vs memoria distribuida. • Multiprocesamiento simétrico. • SIMD, procesamiento de vectores. • GPU, coprocesamiento. • Taxonomía de Flynn. • Soporte a nivel de instrucciones para programación paralela. – Instrucciones atómicas como Compare/Set (Comparar / Establecer) • Problemas de Memoria: – Caches multiprocesador y coherencia de cache – Acceso a Memoria no uniforme (NUMA) • Topologías. – Interconexiones – Clusters – Compartir recursos (p.e., buses e interconexiones)	• Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar] • Describir la arquitectura SMP y observar sus principales características [Evaluar] • Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar] • Describir las ventajas y limitaciones de GPUs vs CPUs [Usar] • Explicar las características de cada clasificación en la taxonomía de Flynn [Usar] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]
Readings : [Pac11], [KH13], [SK10], [HW10]	

Unit 3: Descomposición en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Necesidad de Comunicación y coordinación/sincronización. • Independencia y Particionamiento. • Conocimiento Básico del Concepto de Descomposición Paralela. • Descomposición basada en tareas: – Implementación de estrategias como hebras • Descomposición de Información Paralela – Estrategias como SIMD y MapReduce • Actores y Procesos Reactivos (solicitud de gestores)	• Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar] • Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse] • Escribir un algoritmo paralelo correcto y escalable [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición datos en paralelo [Usar] • Escribir un programa usando actores y/o procesos reactivos [Usar]
Readings : [Pac11], [Mat14], [Qui03], [HW10]	

Unit 4: Comunicación y coordinación (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Memoria Compartida. • La consistencia, y su papel en los lenguaje de programación garantías para los programas de carrera libre. • Pasos de Mensaje: – Mensajes Punto a Punto versus multicast (o basados en eventos) – Estilos para enviar y recibir mensajes Blocking vs non-blocking – Buffering de mensajes • Atomicidad: – Especificar y probar atomicidad y requerimientos de seguridad – Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas – Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas * Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención) – Composición * Componiendo acciones atómicas granulares más grandes usando sincronización * Transacciones, incluyendo enfoques optimistas y conservadores • Consensos: – (Cíclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: – Espera condicional (p.e., empleando variables de condición)	• Usar exclusión mútua para evitar una condición de carrera [Usar] • 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] • Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar] • Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse] • Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar] • 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] • Usar semaforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]
Readings : [Pac11], [Mat14], [Qui03], [HW10]	

Unit 5: Análisis y programación de algoritmos paralelos (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl. • Aceleración y escalabilidad. • Naturalmente (vergonzosamente) algoritmos paralelos. • Patrones Algoritmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) – Algoritmos específicos (p.e., MergeSort paralelo) • Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela) • Cálculos de matriz paralelas. • Productor-consumidor y algoritmos paralelos segmentados. • Ejemplos de algoritmos paralelos no-escalables.	• Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse] • Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar] • Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse] • Identificar tareas independientes en un programa que debe ser paralelizado [Usar] • Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse] • Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su analogo secuencial [Usar] • 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] • Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar] • Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar] • Implementar un algoritmo de matriz paralela [Usar] • Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]
Readings : [Mat14], [Qui03], [HW10]	

Unit 6: Desempeño en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Equilibrio de carga. • La medición del desempeño. • Programación y contención. • Evaluación de la comunicación de arriba. • Gestión de datos: – Costos de comunicación no uniforme debidos a proximidad – Efectos de Cache (p.e., false sharing) – Manteniendo localidad espacial • Consumo de energía y gestión.	• Detectar y corregir un desbalanceo de carga [Usar] • Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar] • Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse] • Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar] • Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse] • Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse] • Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]
Readings : [Pac11], [Mat14], [KH13], [SK10], [HW10]	

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.
- [HW10] Georg Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers (Chapman and 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 2026-I

1. COURSE

CS402. Capstone Project I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS402. Capstone Project I
2.2 Semester	: 8 th 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	: CS401. Research Methodology. (4 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to enable students to conduct a state-of-the-art study on a chosen research topic for investigation.

5. GOALS

- For the student to conduct initial research on a specific topic by performing a state-of-the-art study of the chosen subject.
- For the student to demonstrate mastery in the chosen research line topic.
- For the student to select a faculty member who specializes in the chosen research area as an advisor.
- The deliverables for this course are:

Midterm Progress: Solid bibliography and progress on a Technical Report.

Final: Technical Report with preliminary comparative experiments demonstrating that the student already knows the existing techniques in their project area and selection of a faculty member who specializes in their project area as project advisor.

6. COMPETENCES

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

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

7. TOPICS

Unit 1: State-of-the-Art Survey (60 hours)	
Competences Expected: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12	
Topics	Learning Outcomes
- Conduct a comprehensive state-of-the-art study on a specific topic in Computer Science. - Writing technical articles in computer science.	- Conduct a bibliographic survey of the state-of-the-art on the chosen topic (this likely means 1 or 2 theoretical framework chapters in addition to the introduction which is Chapter I of the thesis) [Usar] - Write a document in LaTeX in article format (<i>paper</i>) with higher quality than in Project I (mastering tables, figures, equations, indexes, bibtex, cross-references, citations, pstricks) [Usar] - Attempt to make presentations using prosper [Usar] - Demonstrate basic experiments [Usar] - Select an advisor who specializes in the research area conducted [Usar]
Readings : [IEE08], [ACM08], [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

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

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

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



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

FG106. Theater (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: FG106. Theater
2.2 Semester	: 8 th 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	: None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Favorece al estudiante a identificarse a la “Comunidad Académica” de la Universidad, en la medida en que le brinda canales naturales de integración a su grupo y a su Centro de Estudios y le permite, desde una visión alternativa, visualizar la valía interior de las personas a su alrededor, a la vez que puede conocer mejor la suya propia. Relaciona al universitario, a través de la experimentación, con un nuevo lenguaje, un medio de comunicación y expresión que va más allá de la expresión verbal conceptualizada. Coadyuva al estudiante en su formación integral, desarrollando en él capacidades corporales. Estimula en él, actitudes anímicas positivas, aptitudes cognitivas y afectivas. Enriquece su sensibilidad y despierta su solidaridad. Desinhibe y socializa, relaja y alegra, abriendo un camino de apertura de conocimiento del propio ser y el ser de los demás.

5. GOALS

- Contribuir a la formación personal y profesional del estudiante, reconociendo, valorando y desarrollando su lenguaje corporal, integrándolo a su grupo, afianzando su seguridad personal, enriqueciendo su intuición, su imaginación y creatividad, motivándolo a abrir caminos de búsqueda de conocimiento de sí mismo y de comunicación con los demás a través de su sensibilidad, de ejercicios de introspección y de nuevas vías de expresión.

6. COMPETENCES

3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04) Communication: Communicates effectively in complex computing activities. (Usage)

7. TOPICS

Unit 1: (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• ¿Qué es el Arte? Una experiencia vivencial y personal. • La llave maestra: la creatividad. • La importancia del teatro en la formación personal y profesional. • Utilidad y enfoque del arte teatral.	• Reconocer la vigencia del Arte y la creatividad en el desarrollo personal y social [Usar]. • Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usar]. • Reconocer nociones básicas del teatro [Usar].
Readings : [Majorana], [PAVIS]	

Unit 2: (6 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Juego, luego existo. • El juego del niño y el juego dramático. • Juegos de integración grupal y juegos de creatividad. • La secuencia teatral.	• Reconocer el juego como herramienta fundamental del teatro [Usar]. • Interiorizar y revalorar el juego como aprendizaje creativo [Usar]. • Acercar al estudiante de manera espontánea y natural, a la vivencia teatral [Usar].
Readings : [Majorana], [PAVIS]	

Unit 3: (9 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Toma de conciencia del cuerpo. • Toma de conciencia del espacio • Toma de conciencia del tiempo • Creación de secuencias individuales y colectivas: Cuerpo, espacio y tiempo. • El uso dramático del elemento: El juego teatral. • Presentaciones teatrales con el uso del elemento.	• Experimentar con nuevas formas de expresión y comunicación [Usar]. • Conocer algunos mecanismos de control y manejo corporal [Usar]. • Brindar caminos para que el alumno pueda desarrollar creativamente su imaginación, su capacidad de relación y captación de estímulos auditivos, rítmicos y visuales [Usar]. • Conocer y desarrollar el manejo de su espacio propio y de sus relaciones espaciales [Usar]. • Experimentar estados emocionales diferentes y climas colectivos nuevos [Usar].
Readings : [Majorana], [PAVIS]	

Unit 4: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Relajación, concentración y respiración. • Desinhibición e interacción con el grupo. • La improvisación. • Equilibrio, peso, tiempo y ritmo. • Análisis del movimiento. Tipos de movimiento. • La presencia teatral. • La danza, la coreografía teatral.	• Ejercitarse en el manejo de destrezas comunicativas no verbales [Usar]. • Practicar juegos y ejercicios de lenguaje corporal, individual y grupalmente [Usar]. • Expresar libre y creativamente sus emociones y sentimientos y su visión de la sociedad a través de representaciones originales con diversos lenguajes [Usar]. • Conocer los tipos de actuación [Usar].
Readings : [Majorana], [PAVIS]	

Unit 5: (3 hours)	
Competences Expected:	
Topics	Learning Outcomes
• El origen del teatro, el teatro griego y el teatro romano. • El teatro medieval , la comedia del arte. • De la pasión a la razón: Romanticismo e Ilustración. • El teatro realista, teatro épico. Brech y Stanislavski. • El teatro del absurdo, teatro contemporáneo y teatro total. • Teatro en el Perú: Yuyashkani, La Tarumba, pataclun, otros.	• Conocer la influencia que la sociedad ha ejercido en el teatro y la respuesta de este arte ante los diferentes momentos de la historia [Usar]. • Apreciar el valor y aporte de las obras de dramaturgos importantes [Usar]. • Analizar el contexto social del arte teatral [Usar]. • Reflexionar sobre el Teatro Peruano y arequipeño [Usar].
Readings : [Majorana], [PAVIS]	

Unit 6: (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Apreciación teatral. Expectación de una o más obras teatrales. • El espacio escénico. • Construcción del personaje • Creación y montaje de una obra teatral . • Presentación en público de pequeñas obras haciendo uso de vestuario, maquillaje, escenografía, utilería y del empleo dramático del objeto.	• Emplear la creación teatral, como manifestación de ideas y sentimientos propios ante la sociedad [Usar]. • Aplicar las técnicas practicadas y los conocimientos aprendidos en una apreciación y/o expresión teatral concreta que vincule el rol de la educación [Usar]. • Intercambiar experiencias y realizar presentaciones breves de ejercicios teatrales en grupo, frente a público [Usar].
Readings : [Majorana], [PAVIS]	

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 2026-I

1. COURSE

EX301FCCS. Extracurricular Activities (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: EX301FCCS. Extracurricular Activities
2.2 Semester	: 8 th 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	: 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

Unit 1: Extracurricular Activities (48 hours)	
Competences Expected: 3,5,AG-C03,AG-C04	
Topics	Learning Outcomes
• Technical or interdisciplinary workshops. • Attendance at conferences, seminars, or talks. • Social outreach (volunteering, mentoring). • Academic skills (hackathons, science fairs).	• Demonstrate participation in at least three activities [Familiarity]. • Prepare reflective reports on experiences [Usage]. • Collaborate in teams for social or technical projects [Evaluar].
Readings : [ACM19], [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

- [ACM19] 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 2026-I

1. COURSE

CS361-CS2023. Computational Vision (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS361-CS2023. Computational Vision
2.2 Semester	: 8 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: AI264. Deep Learning. (7 th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Digital Image Fundamentals (16 hours)	
Competences Expected: 1,6,AG-C08	
Topics	Learning Outcomes
• Spatial filtering (Gaussian, Sobel) • Geometric and morphological transformations • Color spaces (RGB, HSV, LAB)	• Apply basic image processing operations [Usar] • Calibrate filter parameters for real-world cases [Evaluar]
Readings : [Sze10], [GW18]	

Unit 2: Epipolar Geometry and Reconstruction (16 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
• Fundamental and essential matrices • Triangulation and structure-from-motion • Point clouds with Open3D	• Implement 3D reconstruction pipelines [Usar] • Document technical results in reports [Evaluar]
Readings : [HZ04]	

Unit 3: Neural Networks for Vision (16 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
• CNN architectures (ResNet, YOLO) • Transfer learning with TensorFlow • Semantic segmentation (U-Net)	• Train models for image classification [Usar] • Collaborate in teams for integrated projects [Usar]
Readings : [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 2026-I

1. COURSE

CS391-CS2023. Software Engineering III (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS391-CS2023. Software Engineering III
2.2 Semester	: 8 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS292. Software Engineering II. (7 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Software engineering is evolving toward the AI-Native paradigm where autonomous agents co-create software in a secure and observable manner. This course trains engineers capable of designing development platforms augmented with AI, implementing LLMops/AgentOps, and ensuring quality in systems where a significant portion of the lifecycle is automated by artificial intelligence.

5. GOALS

- Design and operate Internal Developer Platforms augmented with autonomous agents that enable creating complete services from natural language with traceability and governance.
- Implement complete LLMops and AgentOps pipelines with prompt versioning, model registry, continuous evaluation, and secure orchestration of agent fleets.
- Develop automated evaluation systems for AI-generated software using proprietary benchmarks, mutation testing, and red teaming of prompts and agents.
- Integrate agents as active participants in modern agile methodologies with automated planning poker, retrospectives with sentiment analysis, and documentation generation.
- Achieve SLSA levels 3-4 and NIST SSDF compliance in repositories where code and infrastructure are generated by AI with provenance and digital signatures.

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. (Assessment)

AG-C09 Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 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)

- 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. (Familiarity)

7. TOPICS

Unit 1: Cuestiones fundamentales (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? – El Test de Turing – Razonamiento Racional versus No Racional • Características del Problema: – Observable completamente versus observable parcialmente – Individual versus multi-agente – Determinístico versus estocástico – Estático versus dinámico – Discreto versus continuo • Naturaleza de agentes: – Autónomo versus semi-autónomo – Reflexivo, basado en objetivos, y basado en utilidad – La importancia en percepción e interacciones con el entorno • Cuestiones filosóficas y éticas.	
Readings : [NC24], [Cha24], [Tea25]	

Unit 2: Cuestiones fundamentales (10 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? – El Test de Turing – Razonamiento Racional versus No Racional • Características del Problema: – Observable completamente versus observable parcialmente – Individual versus multi-agente – Determinístico versus estocástico – Estático versus dinámico – Discreto versus continuo • Naturaleza de agentes: – Autónomo versus semi-autónomo – Reflexivo, basado en objetivos, y basado en utilidad – La importancia en percepción e interacciones con el entorno • Cuestiones filosóficas y éticas.	
Readings : [Bom+21], [Wei+21], [Tea24]	

Unit 3: Programación de Sistemas Interactivos (12 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Patrones de arquitectura de software. Ej Modelo Vista Controlador, Objetos de comando, online, offline. • Patrones de diseño de Interacción: jerarquía visual, distancia navegacional. • Manejo de eventos e interacción de usuario. • Manejo de geometría. • Elección de estilos de interacción y técnicas de interacción. • Presentación de información: navegación, representación, manipulación. • Técnicas de animación de interfaz (ej. grafo de escena) • Clases Widget y bibliotecas. • Bibliotecas modernas de GUI (ej. iOS, Android, JavaFX) constructores de GUI y entornos de programación UI. • Especificación declarativa de Interfaz: Hojas de Estilo y DOMs. • Aplicaciones dirigidas a datos (Páginas web respaldadas por base de datos) • Diseño multiplataforma. • Diseño para dispositivos con restricción de recursos (ej. dispositivos pequeños, móviles)	
Readings : [Bey+18], [KSF23], [Eng24]	

Unit 4: Estructuras de Datos Avanzadas y Análisis de Algoritmos (14 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Árboles balanceados (ej. árboles AVL, Árboles red-black, Árboles biselados (splay trees), Treaps) • Grafos (ej. Ordenamiento Topológico, encontrando componentes fuertemente conectados) • Estructuras de Datos Avanzadas (ej. B-Trees, Fibonacci Heaps) • Estructuras de Datos y Algoritmos basados en cadenas (ej. Arrays de sufijos, Árboles de sufijos, Árboles digitales (Tries) • Redes de Flujo (ej. Flujo Máximo [Algoritmo de Ford-Fulkerson], Flujo Máximo - Mínimo Corte, Máxima Asignación Bipartita) • Programación Lineal (Dualidad, Método Simplex, Algoritmos de punto interior) • Algoritmos Teórico-Numéricos (Aritmética Modular, Prueba del Número Primo, Factorización Entera) • Algoritmos Geométricos (Puntos, Segmentos de Línea, Polígonos [propiedades, intersecciones], encontrando el polígono convexo) • Algoritmos aleatorios. • Algoritmos estocásticos. • Algoritmos de aproximación. • Análisis amortizado. • Análisis Probabilístico. • Algoritmos en línea y análisis competitivo.	
Readings : [Yao+22], [Wu+23], [Mic24]	

Unit 5: Diseño y Testing centrados en el usuario (12 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• Enfoque y características del proceso de diseño. • Requerimientos de funcionalidad y usabilidad. • Técnicas de recolección de requerimientos, ej. entrevistas, encuestas, etnografía e investigación contextual. • Técnicas y herramientas para el análisis y presentación de requerimientos ej. reportes, personas. • Técnicas de creación de prototipos y herramientas, ej. bosquejos, <i>storyboards</i>, prototipos de baja fidelidad, esquemas de página. • Evaluación sin usuarios, usando ambas técnicas cualitativas y cuantitativas. Ej. Revisión estructurada, GOMS, análisis basado en expertos, heurísticas, lineamientos y estándar. • Evaluación con usuarios. Ej. Observación, Método de pensamiento en voz alta, entrevistas, encuestas, experimentación. • Desafíos para la evaluación efectiva, por ejemplo, toma de muestras, la generalización. • Reportar los resultados de las evaluaciones. • Internacionalización, diseño para usuarios de otras culturas, intercultural.	
Readings : [Goo24], [Met24], [Ama24]	

Unit 6: Fundamentos y Conceptos en Seguridad (8 hours)	
Competences Expected: 2,5,6,AG-C03,AG-C07,AG-C09	
Topics	Learning Outcomes
• CIA (Confidencialidad, Integridad, Disponibilidad) • Conceptos de riesgo, amenazas, vulnerabilidades, y los tipos de ataque . • Autenticación y autorización, control de acceso (vs. obligatoria discrecional) • Concepto de la confianza y la honradez . • Ética (revelación responsable)	
Readings : [ST24], [Fou24], [Goo24]	

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

- [Bey+18] Betsy Beyer et al. *The Site Reliability Workbook: Practical Ways to Implement SRE*. O'Reilly Media, 2018.
- [Bom+21] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *arXiv preprint* arXiv:2108.07258 (2021).
- [Wei+21] Laura Weidinger et al. "Ethical and social risks of harm from language models". In: *arXiv preprint* arXiv:2112.04359 (2021).
- [Yao+22] Shunyu Yao et al. "ReAct: Synergizing Reasoning and Acting in Language Models". In: *arXiv preprint* arXiv:2210.03629 (2022).
- [KSF23] Gene Kim, Steven Spear, and Nicole Forsgren. *The Platform Engineering Playbook: Building Internal Developer Platforms*. IT Revolution Press, 2023.
- [Wu+23] Qingyun Wu et al. "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation". In: *arXiv preprint* arXiv:2308.08155 (2023).
- [Ama24] Amazon AWS AI. *Building Reliable AI Systems on AWS*. Amazon Press, 2024.
- [Cha24] Harrison Chase. *LangChain for LLM Application Development: From Prompt Engineering to Agent Orchestration*. O'Reilly Media, 2024.
- [Eng24] Spotify Engineering. *Backstage: Building an Internal Developer Portal*. O'Reilly Media, 2024.
- [Fou24] Linux Foundation. *SLSA Framework v1.0: Supply-chain Levels for Software Artifacts*. Linux Foundation, 2024.
- [Goo24] Google AI Research. *Evaluating AI-Generated Code: Benchmarks and Methodologies*. Google Press, 2024.
- [Met24] Meta AI Research. *Quality Assurance for AI-Native Systems*. Meta Press, 2024.
- [Mic24] Microsoft Research. *AI Software Development: From Copilot to Autonomous Agents*. Microsoft Press, 2024.
- [NC24] Andrew Ng and Harrison Chase. *AI-Native Software Engineering: Building Systems with Autonomous Agents*. DeepLearning.AI Press, 2024.
- [ST24] National Institute of Standards and Technology. *AI Risk Management Framework and Secure AI Supply Chain*. NIST, 2024.
- [Tea24] OpenAI Research Team. *GPT Engineering: Best Practices for Production Systems*. OpenAI, 2024.
- [Tea25] Anthropic Research Team. *Building AI-Native Applications: Principles and Practices*. Anthropic, 2025.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS393-CS2023. Information systems (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS393-CS2023. Information systems
2.2 Semester	: 8 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS291-CS2023. Software Engineering I. (5 th 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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Introduction (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Introduction to information management. • Software for information management. • Technology for information management.	• Correctly apply technology for information management [Evaluar]
Readings : [Som17], [PM15], [LL17]	

Unit 2: Strategy (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Strategy for information management. • Strategy for knowledge management • Strategy for information system.	• Apply and evaluate correctly management strategies [Evaluar]
Readings : [Som17], [PM15]	

Unit 3: Implementation (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Management Information Systems Development. • Change management • Information Architecture	• Implement and correctly evaluate implementation strategies [Evaluar]
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.
- [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.

Chapter 9

Ninth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS370-CS2023. Big Data (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS370-CS2023. Big Data
2.2 Semester	: 9 th 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	: CS3P1-CS2023. Parallel and Distributed Computing . (8 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In today's digital era, processing large data volumes (terabytes to exabytes) is crucial. This course covers modern technologies like Spark, Hadoop, and large-scale graph processing, preparing students to design scalable solutions in distributed and cloud environments.

5. GOALS

- Design distributed applications for massive data processing
- Evaluate Big Data architectures for scalability/latency
- Implement solutions with modern frameworks (Spark, Flink, Hadoop)
- Analyze real-world case studies from tech industries

6. COMPETENCES

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)

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)

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

Unit 1: Big Data Fundamentals (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Lambda and Kappa architectures • Modern distributed systems • Cloud storage (S3, GCS) • Introduction to Spark/Hadoop	• Explain distributed processing challenges [Familiarize] • Compare batch vs streaming architectures [Usar] • Configure cloud environments for Big Data [Usar]
Readings : [Whi21], [Dam+23]	

Unit 2: Spark and Flink (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Spark 3.0+ architecture • Spark SQL and DataFrames • Streaming with Structured Streaming • Flink for real-time • Query optimization	• Develop ETL pipelines with PySpark [Usar] • Implement streaming applications [Evaluar] • Optimize jobs with partitioning [Evaluar]
Readings : [Dam+23], [Per23]	

Unit 3: Storage and Graphs (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Distributed databases (Cassandra, HBase) • Graph processing with GraphX • Delta storage (Delta Lake) • Data Lakes vs Warehouses	• Design NoSQL schemas [Usar] • Implement graph algorithms [Evaluar] • Manage metadata in data lakes [Usar]
Readings : [RWE15], [Zah+16]	

Unit 4: Advanced Applications (13 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Machine Learning with MLlib • CI/CD pipelines for Big Data • Security and governance • Case studies: Netflix, Uber, TikTok	• Deploy ML models at scale [Evaluar] • Implement security policies [Usar] • Analyze industrial architectures [Evaluar]
Readings : [Kle17], [MW15]	

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

- [MW15] Nathan Marz and James Warren. *Big Data: Principles and Best Practices*. Lambda architecture deep dive. Manning, 2015.
- [RWE15] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. Graph processing fundamentals. O'Reilly, 2015.
- [Zah+16] Matei Zaharia et al. "Apache Spark: A unified engine for big data processing". In: *Communications of the ACM* 59.11 (2016), pp. 56–65.
- [Kle17] Martin Kleppmann. *Designing Data-Intensive Applications*. Architecture patterns for Big Data. O'Reilly, 2017.
- [Whi21] Tom White. *Hadoop: The Definitive Guide*. 5th. Covers Hadoop 3 and YARN. O'Reilly, 2021.
- [Dam+23] Jules Damji et al. *Learning Spark: Lightning-Fast Data Analytics*. 2nd. Spark 3.5 with Python/SQL examples. O'Reilly, 2023.
- [Per23] Jean-Georges Perrin. *Spark in Action*. 2nd. Practical Spark applications. Manning, 2023.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS400FCCS. Pre-professional internships (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS400FCCS. Pre-professional internships
2.2 Semester	: 9 th 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	: 110Cr

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course allows students to apply the knowledge acquired during their academic training in a real work environment, under the supervision of a company and the university. Pre-professional internships are essential to develop professional, ethical, and technical competencies, ensuring that the student gains experience in projects related to the degree program.

5. GOALS

- Evaluate student performance in a real work environment, applying Computer Science knowledge.
- Develop professional, ethical, and teamwork skills in a business context.
- Ensure that the activities performed are aligned with the degree program objectives.

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. (Assessment)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Assessment)

- 3) Communicate effectively in a variety of professional contexts.. (Assessment)

AG-C04) Communication: Communicates effectively in complex computing activities. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

AG-C06) Lifelong Learning: Recognizes the importance of continuous learning and adapting to new technologies. (Assessment)

h) A recognition of the need for, and an ability to engage in life-long learning. (Assessment)

AG-C05) Project Management: Applies project management principles in computing to manage projects. (Usage)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (Assessment)

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)

7. TOPICS

Unit 1: Performance Evaluation in Work Environment (48 hours)	
Competences Expected: 2,3,4,5,AG-C01,AG-C02,AG-C03,AG-C04,AG-C05,AG-C06,AG-C09,h,i	
Topics	Learning Outcomes
• Student integration into a professional work team. • Application of technical knowledge in real projects. • Evaluation of ethical and professional competencies in the work environment. • Presentation of reports and feedback from the company and university.	• Demonstrate technical skills in a real work environment [Evaluar]. • Apply ethical and professional principles in their activities [Evaluar]. • Work effectively in teams and communicate professionally [Evaluar].
Readings : [Edu22], [Soc21]	

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

- [Soc21] IEEE Computer Society. *Prácticas Profesionales en Computación: Ética y Gestión*. Manual de buenas prácticas para entornos laborales en TI. IEEE Press, 2021.
- [Edu22] Comité de Educación de la ACM. *Guía para Prácticas Pre-profesionales Exitosas en Ciencia de la Computación*. Tech. rep. Documento oficial con estándares para prácticas en computación. Association for Computing Machinery, 2022. URL: <https://www.acm.org/education>.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS403. Capstone Project II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS403. Capstone Project II
2.2 Semester	: 9 th 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	: CS402. Capstone Project I. (8 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to enable students to develop their research proposal and experiments based on the state-of-the-art previously surveyed.

5. GOALS

- For the student to be capable of formally presenting their thesis project with complete theoretical framework and bibliographic survey.
- For the student to develop their research proposal.
- The deliverables for this course are:

Midterm Progress: Partial progress of their research proposal and thesis plan including motivation, context, problem definition, objectives, activity schedule up to the final thesis project, the state-of-the-art of the addressed topic, and partial advances of their research proposal including how they propose to use/create/improve/apply new or existing algorithms.

Final: Thesis plan at an advanced stage, thesis progress including chapters on thesis proposal and preliminary experiments.

6. COMPETENCES

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

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

7. TOPICS

Unit 1: Thesis Project (30 hours)	
Competences Expected: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12	
Topics	Learning Outcomes
• Thesis Project.	• Description of the format used by the University for the thesis plan [Evaluar] • Complete the thesis project plan[Evaluar] • Present the state-of-the-art of the thesis topic (50%)[Evaluar]
Readings : [IEE08], [ACM08], [Cit08]	

Unit 2: Thesis Progress (30 hours)	
Competences Expected: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12	
Topics	Learning Outcomes
• Thesis Progress.	• Description of the format used by the University for the thesis[Evaluar] • Complete the Theoretical Framework chapter of the Thesis[Evaluar] • Complete the Related Works chapter (35%)[Evaluar] • Plan, develop and present results (formal or statistical) of experiments oriented to their thesis topic (35%)[Evaluar]
Readings : [IEE08], [ACM08], [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

- [ACM08] ACM. *Digital Libray*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.
- [Cit08] CiteSeer.IST. *Scientific Literature Digital Libray*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.
- [IEE08] IEEE-Computer Society. *Digital Libray*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

AI365. Advanced Generative AI Models (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: AI365. Advanced Generative AI Models
2.2 Semester	: 9 th 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	: AI264. Deep Learning. (7 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course explores modern generative models (GANs, Diffusion Models, LLMs) for multimodal content creation, covering both mathematical foundations and practical applications with ethical considerations for synthetic media.

5. GOALS

- Implement image/text/video generation pipelines
- Design AI-assisted editing systems
- Evaluate synthetic content risks

6. COMPETENCES

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

7. TOPICS

Unit 1: Generative Fundamentals (12 hours)	
Competences Expected: 6,AG-C12	
Topics	Learning Outcomes
• GANs theory (minimax, Wasserstein) • Autoregressive models (PixelCNN, Transformers) • Evaluation metrics (FID, Inception Score)	• Train basic GANs [Usar] • Analyze generation failure modes [Evaluar]
Readings : [Goo+14]	

Unit 2: Diffusion Models (24 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
• Diffusion processes (forward/reverse) • Stable Diffusion and latent architectures • Control techniques (ControlNet, LoRA)	• Implement diffusion samplers [Usar] • Fine-tune for domain-specific generation [Evaluar]
Readings : [SSE23], [Rom+22]	

Unit 3: Synthetic Media Ethics (8 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• Deepfakes and misinformation • Copyright in generated content • Authenticity marking (C2PA, watermarking)	• Detect synthetic content [Familiarizarse] • Implement traceability systems [Usar]
Readings : [SSE23]	

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

- [Goo+14] Ian Goodfellow et al. "Generative Adversarial Networks". In: *NeurIPS*. 2014.
- [Rom+22] Robin Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *CVPR* (2022).
- [SSE23] Jascha Sohl-Dickstein, Yang Song, and Stefano Ermon. *Diffusion Models: A Comprehensive Practical Guide*. AI Press, 2023.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CB309. Bioinformatics (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CB309. Bioinformatics
2.2 Semester	: 9 th 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	: BI101FCCS. Biology. (7 th 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

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Usage)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Usage)

7. TOPICS

Unit 1: Introduction to Molecular Biology (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Review of organic chemistry: molecules and macro-molecules, sugars, nucleic acids, nucleotides, RNA, DNA, proteins, amino acids and levels of structure in proteins. - The Dogma of Life: From DNA to Proteins, Transcription, Translation, Protein Synthesis. - Genome study: Maps and sequences, specific techniques	- Achieve a general knowledge of the most important topics in Molecular Biology. [Familiarizarse] - Understand that biological problems are a challenge to the computational world. [Evaluar]
Readings : [CB00], [SM97]	

Unit 2: Sequence Comparison (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Sequences of nucleotides and amino acid sequences. - Sequence alignment, paired alignment problem, exhaustive search, Dynamic programming, global alignment, local alignment, gaps penalty - Comparison of multiple sequences: sum of pairs, complexity analysis by dynamic programming, alignment heuristics, star algorithm, progressive alignment algorithms.	- Understand and solve the problem of aligning a pair of sequences. [Usar] - Understand and solve the problem of multiple sequence alignment. [Usar] - Know the various algorithms for aligning existing sequences in the literature . [Familiarizarse]
Readings : [CB00], [SM97], [Pev00]	

Unit 3: Phylogenetic Trees (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Phylogeny: Introduction and phylogenetic relations - Phylogenetic trees: definition, type of trees, problem of search and reconstruction of trees - Reconstruction methods: parsimony methods, distance methods, maximum likelihood methods, confidence of reconstructed trees	- Understand the concept of phylogeny, phylogenetic trees and the methodological difference between biology and molecular biology. [Familiarizarse] - Understand the problem of the reconstruction of phylogenetic trees, to know and apply the main algorithms for the reconstruction of phylogenetic trees. [Evaluar]
Readings : [CB00], [SM97], [Pev00]	

Unit 4: DNA Sequence Assembling (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Biological basis: ideal case, difficulties, alternative methods for DNA sequencing • Formal Assembly Models: Shortest Common Superstring, Reconstruction, Multicontig • Algorithms for sequence assembly: representation of overlaps, paths to create superstrings, voracious algorithm, acyclic graphs. • Assembly heuristics: search for overlays, ordering fragments, alignments and consensus.	• Understand the computational challenge of the Sequence Assembly problem. [Familiarizarse] • Understand the principle of formal model for assembly. [Evaluar] • Know the main heuristics for the problem of assembly of DNA sequences[Usar]
Readings : [SM97], [Alu06]	

Unit 5: Secondary and tertiary structures (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Molecular structures: primary, secondary, tertiary, quaternary. • Prediction of secondary structures of RNA: formal model, pair energy, structures with independent bases, solution with Dynamic Programming, structures with loops. • <i>Protein folding</i>: Estructuras en proteínas, problema de protein folding. • <i>Protein Threading</i>: Definitions, Branch Bound Algorithm, Branch Bound for protein threading. • <i>Structural Alignment</i>: Definitions, DALI algorithm	• Know the protein structures and the necessity of computational methods for the prediction of the geometry. [Familiarizarse] • Know the algorithms for solving prediction problems of secondary structures RNA, and structures in proteins. [Evaluar]
Readings : [SM97], [CB00], [Alu06]	

Unit 6: Probabilistic Models in Molecular Biology (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Probability: Random Variables, Markov Chains, Metropoli-Hasting Algorithm, Markov Random Fields, and Gibbs Sampler, Maximum Likelihood. • 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. • Probabilistic phylogeny: probabilistic models of evolution, likelihood of alignments, likelihood for inference, comparison of probailistic and non-probabilistic methods	• Review concepts of Probabilistic Models and understand their importance in Computational Molecular Biology. [Evaluar] • Know and apply Hidden Markov Models for various analyzes in Molecular Biology.. [Usar] • Know the application of probabilistic models in Phylogeny and to compare them with non-probabilistic models[Evaluar]
Readings : [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 2026-I

1. COURSE

CS351-CS2023. Topics in Computer Graphics (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS351-CS2023. Topics in Computer Graphics
2.2 Semester	: 9 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS251-CS2023. Computer graphics . (7 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Advanced course covering modern rendering techniques (real-time ray tracing, path tracing), advanced shaders, and GPU computing. Includes applications in games, VR/AR, and scientific visualization with focus on DirectX 12 Ultimate and Vulkan Ray Tracing.

5. GOALS

- Implement advanced rendering algorithms using modern APIs (Vulkan, DX12)
- Design complex shaders (ray tracing, compute shaders)
- Optimize graphics pipelines for current hardware (RTX, RDNA3)
- Analyze AI techniques applied to graphics (DLSS, Neural Rendering)

6. COMPETENCES

- 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)

- 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)

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

Unit 1: Advanced Rendering (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Real-Time Ray Tracing (DXR, Vulkan RT) • Differential Path Tracing • AI Denoising (DLSS, XeSS) • Pipeline State Objects (PSOs)	• Implement global illumination with RTX [Usar] • Integrate neural upscalers [Evaluar] • Optimize pipelines for multi-GPU [Evaluar]
Readings : [MSW21], [PM23]	

Unit 2: GPU Computing (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Modern GPU Architectures (Ada Lovelace, RDNA3) • Compute Shaders (Wave Ininsics) • Advanced Culling (Mesh Shaders) • Rasterization-Free Rendering (Nanite, Lumen)	• Program parallel compute kernels [Usar] • Design geometry culling systems [Evaluar] • Profile GPU code with Nsight/RGP [Usar]
Readings : [KG23], [LN22]	

Unit 3: Graphics Frontier (20 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Neural Rendering (NeRFs, Gaussian Splatting) • Extended Reality (OpenXR) • Probabilistic Shaders • Future Hardware (Chiplets, Photonics)	• Implement basic NeRF models [Usar] • Evaluate XR techniques [Evaluar] • Discuss physical limits of rendering [Familiarizarse]
Readings : [HA23], [YR23]	

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

- [MSW21] Adam Marrs, Peter Shirley, and Ingo Wald. *Ray Tracing Gems II*. Advanced RT techniques with DLSS and hybrid rendering. Apress, 2021.
- [LN22] Timothy Lottes and NVIDIA. “RTX Accelerated Wavelet Noise”. In: *SIGGRAPH Advances in Real-Time Rendering*. Procedural texturing with RT cores. 2022.

-
- [HA23] Eric Haines and Tomas Akenine-Möller. “The State of Ray Tracing in 2023”. In: *Journal of Computer Graphics Techniques* 11.2 (2023).
- [KG23] Brian Karis and Epic Games. “Nanite and Lumen Technical Deep Dive”. In: *Unreal Engine Whitepapers* (2023). Virtualized geometry and global illumination in UE5.
- [PM23] Nuno Pereira and Microsoft. *DirectX Raytracing Documentation*. Tech. rep. Official DXR programming guide. Microsoft, 2023.
- [YR23] Lingjie Yan and Google Research. “Neural Radiance Fields for Real-Time View Synthesis”. In: *ACM Transactions on Graphics* 42.4 (2023). Real-time NeRF optimizations.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS353-CS2023. Quantum Computing (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS353-CS2023. Quantum Computing
2.2 Semester	: 9 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS3I1-CS2023. Computer Security. (8 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamental principles of quantum computing, including qubits, superposition, entanglement, quantum algorithms, and hardware. Students will contrast classical and quantum models, exploring applications in cryptography, optimization, physical simulation, and quantum machine learning, using frameworks like Qiskit or Cirq. Includes a practical module on hardware architectures (superconductors, ion traps) and their impact on algorithm design.

5. GOALS

- Understand the postulates of quantum mechanics applied to computing.
- Implement basic quantum circuits and optimization algorithms.
- Analyze the ethical, technical, and commercial impact of quantum computing.
- Evaluate hardware limitations in real-world applications.

6. COMPETENCES

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)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Usage)

AG-C10) Investigation: Studies complex computing problems using information science methods. (Familiarity)

7. TOPICS

Unit 1: Fundamentals of Quantum Mechanics (12 hours)	
Competences Expected: 1,AG-C08	
Topics	Learning Outcomes
• Qubits and Bloch sphere representation • Postulates of quantum mechanics • Quantum gates (Hadamard, CNOT, Toffoli)	• Mathematically model a qubit [Usar] • Simulate basic operations in Qiskit [Evaluar]
Readings : [NC10], [Tea23]	

Unit 2: Quantum Hardware (12 hours)	
Competences Expected: AG-C07	
Topics	Learning Outcomes
• Architectures: superconductors (IBM, Google) vs. ion traps (IonQ) • Metrics: coherence, error rates, qubit connectivity • Circuit compilation for real hardware	• Compare pros/cons of hardware platforms [Familiarize] • Adapt circuits to hardware constraints [Usar]
Readings : [MV23], [23]	

Unit 3: Quantum Algorithms (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Deutsch-Jozsa algorithm • Grover's search (applied to combinatorial optimization) • Shor's factorization and its impact on RSA	• Implement Grover for SAT problems [Usar]
Readings : [Pre18], [Tea23]	

Unit 4: Optimization Applications (8 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Quantum Approximate Optimization Algorithm (QAOA) • Logistics/finance applications (Portfolio Optimization) • Benchmarks vs. classical algorithms (Simulated Annealing)	• Design a QAOA for routing problems [Evaluar]
Readings : [QMLBook], [QuantumOptimizationSurvey]	

Unit 5: Ethics and Applications (4 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Post-quantum cryptography (NIST PQC Standardization) • Quantum supremacy: technical/ethical debate • Patents and quantum geopolitics	• Propose post-quantum migration strategies [Familiarizarse]
Readings : [Tea19], [PostQuantumCrypto]	

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>.
- [23] *IBM Quantum Hardware Roadmap*. Tech. rep. IBM Research, 2023. URL: <https://research.ibm.com/quantum-hardware>.
- [MV23] Mikko Möttönen and Juha Vartiainen. *Quantum Computing: From Qubits to Quantum Machines*. Springer, 2023.
- [Tea23] IBM Quantum Team. *Qiskit Textbook*. Guía práctica con ejemplos en Qiskit. 2023. URL: <https://qiskit.org/textbook>.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS369-CS2023. Robotics (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS369-CS2023. Robotics
2.2 Semester	: 9 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS361-CS2023. Computational Vision. (8 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course integrates principles of artificial intelligence, control systems, and programming to design autonomous robotic solutions, emphasizing complex problem analysis, teamwork, and professional ethics.

5. GOALS

- Design robotic algorithms applying computational theory (ABET6/AG-C12).
- Evaluate ethical and social impacts of robotics (ABET4/AG-C02).
- Function as a leader or member of a multidisciplinary team (ABET5/AG-C03).

6. COMPETENCES

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)

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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (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)

7. TOPICS

Unit 1: Fundamentals of Robotics (12 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Kinematics and dynamics of robots. • Modeling with λ-calculus.	• Apply computational theory to robotic design [Usar].
Readings : [Sic+09], [Mur00]	

Unit 2: Ethics and Society (12 hours)	
Competences Expected: 4,AG-C02	
Topics	Learning Outcomes
• Ethical dilemmas in robotic autonomy. • Social impact of robotics.	• Analyze professional responsibilities [Evaluar].
Readings : [Mur00], [TBF05]	

Unit 3: Team Project (24 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Collaborative prototype development. • Technical documentation and presentation.	• Lead teams in robotic projects [Evaluar].
Readings : [Sic+09], [LaV06]	

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

- [Mur00] Robin R. Murphy. *Introduction to AI Robotics*. MIT Press, 2000.
- [TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [LaV06] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006. URL: <http://planning.cs.uiuc.edu/>.
- [Sic+09] Bruno Siciliano et al. *Robotics: Modelling, Planning and Control*. Springer, 2009. DOI: 10.1007/978-1-84628-642-1.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS392-CS2023. Tópicos en Ingeniería de Software (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS392-CS2023. Tópicos en Ingeniería de Software
2.2 Semester	: 9 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS391-CS2023. Software Engineering III. (8 th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

AG-C05) Project Management: Applies project management principles in computing to manage projects. (Familiarity)

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 (<i>3-tier</i>), <i>pipe-and-filter</i>, 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
• La participación del equipo: – Procesos elemento del equipo, incluyendo responsabilidades de tarea, la estructura de reuniones y horario de trabajo – Roles y responsabilidades en un equipo de software – Equipo de resolución de conflictos – Los riesgos asociados con los equipos virtuales (comunicación, la percepción, la estructura) • Estimación de esfuerzo (a nivel personal) • Riesgo. – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • Gestión de equipos: – Organización de equipo y la toma de decisiones – Roles de identificación y asignación – Individual y el desempeño del equipo de evaluación • Gestión de proyectos: – Programación y seguimiento de elementos – Herramientas de gestión de proyectos – Análisis de Costo/Beneficio • Software de medición y técnicas de estimación. • Aseguramiento de la calidad del software y el rol de las mediciones. • Riesgo. – El papel del riesgo en el ciclo de vida – Categorías elemento de riesgo, incluyendo la seguridad, la seguridad, mercado, finanzas, tecnología, las personas, la calidad, la estructura y el proceso de • En todo el sistema de aproximación al riesgo, incluyendo riesgos asociados con herramientas.	• Discutir los comportamientos comunes que contribuyen al buen funcionamiento de un equipo [Usar] • Crear y seguir un programa para una reunión del equipo [Usar] • Identificar y justificar las funciones necesarias en un equipo de desarrollo de software [Usar] • Entender las fuentes, obstáculos y beneficios potenciales de un conflicto de equipo [Usar] • Aplicar una estrategia de resolución de conflictos en un ambiente de equipo [Usar] • Utilizar un método ad hoc para estimar el esfuerzo de desarrollo del software (ejemplo, tiempo) y comparar con el esfuerzo actual requerido [Usar] • Listar varios ejemplos de los riesgos del software [Usar] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar] • Describir las diferentes categorías de riesgo en los sistemas de software [Usar] • Demostrar a través de la colaboración de proyectos de equipo los elementos centrales de la construcción de equipos y gestión de equipos [Usar] • Describir como la elección de modelos de procesos afectan la estructura organizacional de equipos y procesos de toma de decisiones [Usar] • Crear un equipo mediante la identificación de los roles apropiados y la asignación de funciones a los miembros del equipo [Usar] • Evaluar y retroalimentar a los equipos e individuos sobre su desempeño en un ambiente de equipo [Usar] • 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] • Realizar el seguimiento del progreso de alguna etapa de un proyecto que utiliza métricas de proyectos apropiados [Usar] • Comparar las técnicas simples de tamaño de software y estimación de costos [Usar] • Usar una herramienta de gestión de proyectos para ayudar en la asignación y rastreo de tareas en un proyecto de desarrollo de software [Usar] • Describir el impacto del riesgo en el ciclo de vida de desarrollo de software [Usar]

Unit 3: ITIL (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Administración del servicio como práctica. • Ciclo de vida del servicio. • Definiciones y conceptos genéricos. • Modelos y principios claves. • Procesos. • Tecnología y arquitectura. • Competencia y entrenamiento.	• Utilizar y aplicar correctamente ITIL en el proceso de software. [Usar]
Readings : [Som17], [PM15]	

Unit 4: COBIT (14 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Fundamentos e Introducción. • Frameworks de Control y IT Governance.	• Utilizar y aplicar correctamente COBIT en el proceso de software. [Usar]
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 2026-I

1. COURSE

CS3P3-CS2023. Internet of Things (Elective)

2. GENERAL INFORMATION

2.1 Course	: CS3P3-CS2023. Internet of Things
2.2 Semester	: 9 th 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	: Elective
2.7 Learning modality	: Face to face
2.8 Prerequisites	: CS3P1-CS2023. Parallel and Distributed Computing . (8 th 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

- 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)

- 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)

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

Unit 1: Fundamentos de paralelismo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Procesamiento Simultáneo Múltiple. • Metas del Paralelismo (ej. rendimiento) frente a Concurrencia (ej. control de acceso a recursos compartidos) • Paralelismo, comunicación, y coordinación: – Paralelismo, comunicación, y coordinación – Necesidad de Sincronización • Errores de Programación ausentes en programación secuencial: – Tipos de Datos (lectura/escritura simultánea o escritura/escritura compartida) – Tipos de Nivel más alto (interleavings violating program intention, no determinismo no deseado) – Falta de vida/progreso (deadlock, starvation)	• Distinguir el uso de recursos computacionales para una respuesta mas rápida para administrar el acceso eficiente a un recurso compartido [Familiarizarse] • Distinguir múltiples estructuras de programación suficientes para la sincronización que pueden ser interimplementables pero tienen ventajas complementarias [Familiarizarse] • Distinguir datos de carrera (<i>data races</i>) a partir de carreras de mas alto nivel [Familiarizarse]
Readings : [Pac11], [Mat14], [Qui03]	

Unit 2: Arquitecturas paralelas (12 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Procesadores mutlinúcleo. • Memoria compartida vs memoria distribuida. • Multiprocesamiento simétrico. • SIMD, procesamiento de vectores. • GPU, coprocesamiento. • Taxonomía de Flynn. • Soporte a nivel de instrucciones para programación paralela. – Instrucciones atómicas como Compare/Set (Comparar / Establecer) • Problemas de Memoria: – Caches multiprocesador y coherencia de cache – Acceso a Memoria no uniforme (NUMA) • Topologías. – Interconexiones – Clusters – Compartir recursos (p.e., buses e interconexiones)	• Explicar las diferencias entre memoria distribuida y memoria compartida [Evaluar] • Describir la arquitectura SMP y observar sus principales características [Evaluar] • Distinguir los tipos de tareas que son adecuadas para máquinas SIMD [Usar] • Describir las ventajas y limitaciones de GPUs vs CPUs [Usar] • Explicar las características de cada clasificación en la taxonomía de Flynn [Usar] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse] • Describir los desafíos para mantener la coherencia de la caché [Familiarizarse]
Readings : [Pac11], [KH13], [SK10]	

Unit 3: Descomposición en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Necesidad de Comunicación y coordinación/sincronización. • Independencia y Particionamiento. • Conocimiento Básico del Concepto de Descomposición Paralela. • Descomposición basada en tareas: – Implementación de estrategias como hebras • Descomposición de Información Paralela – Estrategias como SIMD y MapReduce • Actores y Procesos Reactivos (solicitud de gestores)	• Explicar por qué la sincronización es necesaria en un programa paralelo específico [Usar] • Identificar oportunidades para particionar un programa serial en módulos paralelos independientes [Familiarizarse] • Escribir un algoritmo paralelo correcto y escalable [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición basada en tareas [Usar] • Paralelizar un algoritmo mediante la aplicación de descomposición datos en paralelo [Usar] • Escribir un programa usando actores y/o procesos reactivos [Usar]
Readings : [Pac11], [Mat14], [Qui03]	

Unit 4: Comunicación y coordinación (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Memoria Compartida. • La consistencia, y su papel en los lenguaje de programación garantías para los programas de carrera libre. • Pasos de Mensaje: – Mensajes Punto a Punto versus multicast (o basados en eventos) – Estilos para enviar y recibir mensajes Blocking vs non-blocking – Buffering de mensajes • Atomicidad: – Especificar y probar atomicidad y requerimientos de seguridad – Granularidad de accesos atómicos y actualizaciones, y uso de estructuras como secciones críticas o transacciones para describirlas – Exclusión mutua usando bloques, semáforos, monitores o estructuras relacionadas * Potencial para fallas y bloqueos (<i>deadlock</i>) (causas, condiciones, prevención) – Composición * Componiendo acciones atómicas granulares más grandes usando sincronización * Transacciones, incluyendo enfoques optimistas y conservadores • Consensos: – (Cíclicos) barreras, contadores y estructuras relacionadas • Acciones condicionales: – Espera condicional (p.e., empleando variables de condición)	• Usar exclusión mútua para evitar una condición de carrera [Usar] • 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] • Dar un ejemplo de un escenario en el que el bloqueo de mensajes enviados pueden dar <i>deadlock</i> [Usar] • Explicar cuándo y por qué mensajes de multidifusión (<i>multicast</i>) o basado en eventos puede ser preferible a otras alternativas [Familiarizarse] • Escribir un programa que termine correctamente cuando todo el conjunto de procesos concurrentes hayan sido completados [Usar] • 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] • Usar semáforos o variables de condición para bloquear hebras hasta una necesaria precondition de mantenga [Usar]
Readings : [Pac11], [Mat14], [Qui03]	

Unit 5: Análisis y programación de algoritmos paralelos (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Caminos críticos, el trabajo y la duración y la relación con la ley de Amdahl. • Aceleración y escalabilidad. • Naturalmente (vergonzosamente) algoritmos paralelos. • Patrones Algoritmicos paralelos (divide-y-conquista, map/reduce, amos-trabajadores, otros) – Algoritmos específicos (p.e., MergeSort paralelo) • Algoritmos de grafos paralelo (por ejemplo, la ruta más corta en paralelo, árbol de expansión paralela) • Cálculos de matriz paralelas. • Productor-consumidor y algoritmos paralelos segmentados. • Ejemplos de algoritmos paralelos no-escalables.	• Definir: camino crítico, trabajo y <i>span</i> [Familiarizarse] • Calcular el trabajo y el <i>span</i> y determinar el camino crítico con respecto a un diagrama de ejecución paralela. [Usar] • Definir <i>speed-up</i> y explicar la noción de escalabilidad de un algoritmo en este sentido [Familiarizarse] • Identificar tareas independientes en un programa que debe ser paralelizado [Usar] • Representar características de una carga de trabajo que permita o evite que sea naturalmente paralelizable [Familiarizarse] • Implementar un algoritmo dividir y conquistar paralelo (y/o algoritmo de un grafo) y medir empíricamente su desempeño relativo a su analogo secuencial [Usar] • 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] • Proporcionar un ejemplo de un problema que se corresponda con el paradigma productor-consumidor [Usar] • Dar ejemplos de problemas donde el uso de <i>pipelining</i> sería un medio eficaz para la paralelización [Usar] • Implementar un algoritmo de matriz paralela [Usar] • Identificar los problemas que surgen en los algoritmos del tipo productor-consumidor y los mecanismos que pueden utilizarse para superar dichos problemas [Usar]
Readings : [Mat14], [Qui03]	

Unit 6: Desempeño en paralelo (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Equilibrio de carga. • La medición del desempeño. • Programación y contención. • Evaluación de la comunicación de arriba. • Gestión de datos: – Costos de comunicación no uniforme debidos a proximidad – Efectos de Cache (p.e., false sharing) – Manteniendo localidad espacial • Consumo de energía y gestión.	• Detectar y corregir un desbalanceo de carga [Usar] • Calcular las implicaciones de la ley de Amdahl para un algoritmo paralelo particular [Usar] • Describir como la distribución/disposición de datos puede afectar a los costos de comunicación de un algoritmo [Familiarizarse] • Detectar y corregir una instancia de uso compartido falso (<i>false sharing</i>) [Usar] • Explicar el impacto de la planificación en el desempeño paralelo [Familiarizarse] • Explicar el impacto en el desempeño de la localidad de datos [Familiarizarse] • Explicar el impacto y los puntos de equilibrio relacionados al uso de energía en el desempeño paralelo [Familiarizarse]
Readings : [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>.

Chapter 10

Tenth Semester



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS3P2-CS2023. Cloud Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS3P2-CS2023. Cloud Computing
2.2 Semester	: 10 th 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-CS2023. Big Data. (9 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course covers modern cloud computing fundamentals including virtualization, containerization, distributed processing, and serverless architectures. Students will learn to implement scalable solutions using AWS, Azure, and GCP platforms, focusing on current industry practices.

5. GOALS

- Design cloud architectures following best practices (Well-Architected Framework)
- Implement scalable solutions using managed services (AWS Lambda, Azure Functions)
- Manage container clusters with Kubernetes and Docker
- Optimize costs and performance in cloud environments

6. COMPETENCES

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)

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)

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

Unit 1: Cloud Computing Fundamentals (12 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Service models (IaaS, PaaS, SaaS, FaaS) • Global providers (AWS, Azure, GCP) • Resilient architectures (Multi-AZ, Region Failover) • Security and identity (IAM, RBAC, Zero-Trust)	• Compare cloud service models [Usar] • Design VPC/subnet networks [Evaluar] • Implement security policies [Usar]
Readings : [VC23], [Ser23]	

Unit 2: Large-Scale Data Processing (15 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Spark 3.0 and Delta Lake • Stream processing (Kafka, Flink) • Data Lakes vs Warehouses (Snowflake, Redshift) • Serverless ETL (Glue, Dataflow)	• Implement batch/streaming pipelines [Usar] • Optimize distributed queries [Evaluar] • Manage data metastores [Usar]
Readings : [Dam+23], [KK18]	

Unit 3: Virtualization and Containerization (15 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Advanced Docker (buildx, compose) • Kubernetes (ARK, Operators) • Service Mesh (Istio, Linkerd) • Serverless Containers (AWS Fargate, Cloud Run)	• Deploy EKS/GKE clusters [Usar] • Implement CI/CD for containers [Evaluar] • Monitor with Prometheus/Grafana [Usar]
Readings : [BB23], [Mic23]	

Unit 4: Cloud Computing Trends (12 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Infrastructure as Code (Terraform, Pulumi) • GitOps (ArgoCD, Flux) • Edge Computing (AWS Outposts, Azure Arc) • Quantum Computing in the cloud (Braket, Qiskit)	• Automate deployments with IaC [Usar] • Implement GitOps patterns [Evaluar] • Evaluate hybrid architectures [Usar]
Readings : [Clo23], [Cen23]	

Unit 5: Sistemas distribuidos (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
- Fallos: - Fallos basados en red (incluyendo particiones) y fallos basados en nodos - Impacto en garantías a nivel de sistema (p.e., disponibilidad) - Envío de mensajes distribuido: - Conversión y transmisión de datos - Sockets - Secuenciamiento de mensajes - Almacenando <i>Buffering</i>, reenviando y desechando mensajes - Compensaciones de diseño para Sistemas Distribuidos: - Latencia versus rendimiento - Consistencia, disponibilidad, tolerancia de particiones - Diseño de Servicio Distribuido: - Protocolos y servicios Stateful versus stateless - Diseños de Sesión (basados en la conexión) - Diseños reactivos (provocados por E/S) y diseños de múltiples hilos - Algoritmos de Distribución de Núcleos: - Elección, descubrimiento	- Distinguir las fallas de red de otros tipos de fallas [Familiarizarse] - Explicar por qué estructuras de sincronización como cerraduras simples (<i>locks</i>) no son útiles en la presencia de fallas distribuidas [Familiarizarse] - Escribir un programa que realiza cualquier proceso de <i>marshalling</i> requerido y la conversión en unidades de mensajes, tales como paquetes, para comunicar datos importantes entre dos <i>hosts</i> [Usar] - Medir el rendimiento observado y la latencia de la respuesta a través de los <i>hosts</i> en una red dada [Usar] - Explicar por qué un sistema distribuido no puede ser simultáneamente Consistente (<i>Consistent</i>), Disponible (<i>Available</i>) y Tolerante a fallas (<i>Partition tolerant</i>). [Familiarizarse] - Implementar un servidor sencillo - por ejemplo, un servicio de corrección ortográfica [Usar] - Explicar las ventajas y desventajas entre: <i>overhead</i>, escalabilidad y tolerancia a fallas entre escoger un diseño sin estado (<i>stateless</i>) y un diseño con estado (<i>stateful</i>) para un determinado servicio [Familiarizarse] - Describir los desafíos en la escalabilidad, asociados con un servicio creciente para soportar muchos clientes, así como los asociados con un servicio que tendrá transitoriamente muchos clientes [Familiarizarse] - Dar ejemplos de problemas donde algoritmos de consenso son requeridos, por ejemplo, la elección de líder [Usar]
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. Addison-Wesley, 2011.
- [KK18] Holden Karau and Andy Konwinski. *Learning Spark*. O'Reilly, 2018.
- [BB23] Brendan Burns and Joe Beda. *Kubernetes: Up and Running*. 3rd. O'Reilly, 2023.
- [Cen23] IBM Cloud Architecture Center. *Hybrid Cloud Strategies*. Tech. rep. IBM, 2023.
- [Clo23] Google Cloud. *Serverless Architectures with Cloud Functions*. Tech. rep. Google Cloud, 2023.
- [Dam+23] Jules Damji et al. *Learning Spark: Lightning-Fast Data Analytics*. 2nd. O'Reilly, 2023.
- [Mic23] Microsoft. *Containerized Applications on Azure*. Tech. rep. Microsoft Docs, 2023.
- [Ser23] Amazon Web Services. *AWS Well-Architected Framework*. Tech. rep. Official cloud best practices guide. AWS, 2023.
- [VC23] Diego Villegas and Mark Cresswell. *Cloud Native Architectures*. Covers multi-cloud patterns with case studies. Packt, 2023.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS404. Research Workshop (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS404. Research Workshop
2.2 Semester	: 10 th 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	: CS403. Capstone Project II. (9 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course aims to enable students to properly complete their thesis draft.

5. GOALS

- For the student to complete this course with their thesis developed with sufficient quality for immediate defense.
- For the student to formally present the thesis draft to the faculty authorities.
- The deliverables for this course are:

Midterm: Progress of the thesis project including in the document: introduction, theoretical framework, state-of-the-art, proposal, analysis and/or experiments, and updated bibliography.

Final: Complete thesis document ready for defense within no more than fifteen days.

6. COMPETENCES

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

AG-C07) Computing Knowledge: Applies appropriate knowledge of mathematics, science, and computing. (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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Assessment)

- 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. (Assessment)

AG-C02) Ethics: Applies ethical principles and commits to professional ethics and standards of computing practice. (Assessment)

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

AG-C03) Individual and Teamwork: Performs effectively as an individual and as a member or leader in diverse teams. (Assessment)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (Assessment)

AG-C12) Applies computer science theory and software development fundamentals to produce computer-based solutions. (Assessment)

7. TOPICS

Unit 1: Writing the Undergraduate Thesis Draft (60 hours)	
Competences Expected: 1,2,4,5,6,AG-C02,AG-C03,AG-C07,AG-C09,AG-C12	
Topics	Learning Outcomes
• Writing and correction of the undergraduate thesis work	• Completed experimental part (if appropriate for the project) [Evaluar] • Verify that the document complies with the program's thesis format [Evaluar] • Delivery of the finalized thesis draft considered ready for public defense (approval requirement) [Evaluar]
Readings : [IEE08], [ACM08], [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

- [ACM08] ACM. *Digital Library*. <http://portal.acm.org/dl.cfm>. Association for Computing Machinery, 2008.
- [Cit08] CiteSeer.IST. *Scientific Literature Digital Library*. <http://citeseer.ist.psu.edu>. College of Information Sciences and Technology, Penn State University, 2008.
- [IEE08] IEEE-Computer Society. *Digital Library*. <http://www.computer.org/publications/dlib>. IEEE-Computer Society, 2008.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS367-CS2023. Topics in Artificial Intelligence (Mandatory)

2. GENERAL INFORMATION

- 2.1 Course** : CS367-CS2023. Topics in Artificial Intelligence
- 2.2 Semester** : 10th 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** : AI264. Deep Learning. (7th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

AG-C10) Investigation: Studies complex computing problems using information science methods. (Familiarity)

7. TOPICS

Unit 1: (60 hours)	
Competences Expected: 2,6,AG-C09	
Topics	Learning Outcomes
• Advanced topics in AI	• Develop expertise in diverse Artificial Intelligence techniques [Usar]
Readings : [Russell03], [Haykin99], [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 E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Massachusetts: Addison-Wesley Publishing Co., 1989.



National University of Engineering (UNI)
School of Computer Science
Syllabus 2026-I

1. COURSE

CS368-CS2023. Evolutionary Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS368-CS2023. Evolutionary Computing
2.2 Semester	: 10 th 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	: AI264. Deep Learning. (7 th 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

- 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)

- 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)

AG-C09) Solution Design and Development: Designs, implements, and evaluates solutions for complex computing problems. (Usage)

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

AG-C08) Problem Analysis: Identifies, formulates, and analyzes complex computing problems. (Familiarity)

7. TOPICS

Unit 1: Genetic Algorithms (16 hours)	
Competences Expected: 1,AG-C08	
Topics	Learning Outcomes
• Chromosomal representation (binary, real-valued) • Selection operators (roulette, tournament) • Crossover and mutation	• Design representations for discrete problems [Usar] • Compare selection operators [Evaluar]
Readings : [Gol18], [Tea23]	

Unit 2: Evolutionary Strategies (16 hours)	
Competences Expected: 2,AG-C09	
Topics	Learning Outcomes
• $(\mu + \lambda)$ and (μ, λ) algorithms • Parameter self-adaptation • Continuous optimization applications	• Implement strategies for real-world problems [Usar] • Tune self-adaptation parameters [Evaluar]
Readings : [Bey01]	

Unit 3: Advanced Applications (16 hours)	
Competences Expected: 6	
Topics	Learning Outcomes
• Genetic programming • Co-evolution • Multi-objective optimization (NSGA-II)	• Develop multi-objective solutions [Usar] • Evaluate Pareto front trade-offs [Evaluar]
Readings : [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 2026-I

1. COURSE

FG350. Leadership and Performance (Mandatory)

2. GENERAL INFORMATION

- 2.1 Course : FG350. Leadership and Performance
- 2.2 Semester : 10th Semester
- 2.3 Credits : 2
- 2.4 Horas : 2 HT;
- 2.5 Duration of the period : 16 weeks
- 2.6 Type of course : Mandatory
- 2.7 Learning modality : Face to face
- 2.8 Prerequisites : FG106. Theater. (8th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

En la actualidad las diferentes organizaciones en el mundo exigen a sus integrantes el ejercicio de liderazgo, esto significa asumir los retos asignados con eficacia y afán de servicio, siendo estas exigencias necesarias para la búsqueda de una sociedad más justa y reconciliada. Este desafío, pasa por la necesidad de formar a nuestros alumnos con un recto conocimiento de sí mismos, con capacidad de juzgar objetivamente la realidad y de proponer orientaciones que busquen modificar positivamente el entorno.

5. GOALS

- Desarrollar conocimientos, criterios, capacidades y actitudes para ejercer liderazgo, con el objeto de lograr la eficacia y servicio en los retos asignados, contribuyendo así en la construcción de una mejor sociedad.

6. COMPETENCES

- 3) Communicate effectively in a variety of professional contexts.. (Usage)

AG-C04 Communication: Communicates effectively in complex computing activities. (Usage)

7. TOPICS

Unit 1: (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Teorías de Liderazgo: • Definición de Liderazgo. • Fundamentos de Liderazgo. • Visión integral del Ser Humano y Motivos de la acción. • La práctica de la Virtud en el ejercicio de Liderazgo.	• Analizar y comprender las bases teóricas del ejercicio de Liderazgo.[Familiarizarse] • En base a lo comprendido, asumir la actitud correcta para llevarlo a la práctica.[Familiarizarse] • Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo.[Familiarizarse]
Readings : [CL02], [FA09], [Dia09], [Ant], [Sonnenfeld]	

Unit 2: (15 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Teoría de las Competencias • Reconocimiento de Competencias • Plan de Desarrollo • Modelos Mentales • Necesidades Emocionales • Perfiles Emocionales • Vicios Motivacionales	• Conocer y Desarrollar competencias de Liderazgo, centradas en lograr la eficacia, sin dejar de lado el deber de servicio con los demás.[Familiarizarse] • Reconocer las tendencias personales y grupales necesarias para el ejercicio de Liderazgo.[Familiarizarse]
Readings : [CW09], [Hue08], [CL02], [CM07]	

Unit 3: (18 hours)	
Competences Expected:	
Topics	Learning Outcomes
• La relación personal con el equipo • Liderazgo integral • Acompañamiento y discipulado • Fundamentos de unidad	• Desarrollar habilidades para el trabajo en equipo[Familiarizarse]
Readings : [Gol12], [CP], [HBJ98], [Hun10], [Haw12], [GS10]	

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

- [HBJ98] P. Hersey, K. H. Blanchard, and D. E. Johnson. *Administración del comportamiento organizacional: liderazgo situacional*. 1998.
- [CL02] Pablo Cardona and Pilar García Lombardi. *Cómo desarrollar las Competencias de Liderazgo*. PAD Lima- Perú, Tercera Edición, 2002.
- [CM07] Nuria Chinchilla and Maruja Moragas. *Dueños de Nuestro Destino*. Editorial Ariel, 2007.
- [Hue08] Luis Huete. *Construye tu Sueño*. LID Editorial Empresarial, 2008.
- [CW09] Pablo Cardona and Helen Wilkinson. *Creciendo como Líder*. Ediciones Universidad de Navarra S.A (EUNSA), Primera Edición, 2009.
- [Dia09] Alexandre Dianine-Havard. *Perfil del Líder. Hacia un Liderazgo Virtuoso*. Ediciones Urano S.A, 2009.

-
- [FA09] Pablo Ferreiro and Manuel Alcázar. *Gobierno de Personas en la Empresa*. Ediciones Universidad de Navarra EUNSA, 2009.
- [GS10] G. Ginebra and G. G. Serrabou. *Gestión de incompetentes. Libros de Cabecera*. 2010.
- [Hun10] Phil Hunsaker. *El nuevo arte de gestionar equipos: Un enfoque actual para guiar y motivar con éxito*. 2010.
- [Gol12] D. Goleman. *Inteligencia emocional*. Editorial Kairós, 2012.
- [Haw12] Peter Hawkins. *Coaching y liderazgo de equipos: coaching para un liderazgo con capacidad de transformación*. Ediciones Granica, 2012.
- [Ant] SJ Anthony D' Souza. *Descubre tu Liderazgo*. Editorial Sal Terrae.
- [CP] P. Cardona and C. R. Peña. *Dirección por misiones: Cómo generar empresas de alto rendimiento*.