

**National University of San Marcos (UNMSM)
Faculty of Mathematical Sciences
Department of Mathematics
School of Scientific Computing**



Book of Syllabi

– 2026-I –

Lima: November 26, 2025

Equipo de Trabajo

Dra. Maria Zegarra-Garay

Decana de la Facultad de Ciencias Matemáticas

email: *mzegarrag@unmsm.edu.pe*

Ernesto Cuadros-Vargas (Editor)

Orador distinguido para la *Association of Computing Machinery* (ACM)
Miembro del Directorio de Gobernadores de la Sociedad de Computación del
IEEE (2020-2023)

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

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

Presidente de la Sociedad Peruana de Computación (SPC) 2001-2007, 2009
email: *ecuadros@spc.org.pe*

Contents

1	First Semester	5
1.1	CS111. Introduction to Programming	6
1.2	MA111. Calculus I	14
1.3	MA121. Linear Algebra I	18
1.4	ID101. Technical and professional English I	21
2	Second Semester	25
2.1	CS100. Introduction to Computer Science	26
2.2	CS112. Objects oriented programming I	40
2.3	CS1D1. Discrete Structures I	48
2.4	MA112. Calculus II	53
2.5	MA122. Linear Algebra II	56
2.6	ID102. Technical and professional English II	59
3	Third Semester	63
3.1	CS113. Objects oriented programming II	64
3.2	CS1D2. Discrete Structures II	68
3.3	CS2B1. Platform Based Development	72
3.4	AI161. Applied AI	77
3.5	MA211. Calculus III	81
3.6	ID201. Technical and professional English III	84
4	Fourth Semester	89
4.1	CS210. Algorithms and Data Structures	90
4.2	MA212. Differential Equations	93
4.3	MA351. Statistics and Probabilities	96
4.4	ID202. Technical and professional English IV	99
5	Fifth Semester	103
5.1	MA221. Numerical Analysis I	104
5.2	MA222. Real Analysis I	107
5.3	MA223. Graph Theory	111
5.4	ID203. Technical and professional English V	114
6	Sixth Semester	119
6.1	MA321. Abstract algebra	120
6.2	MA325. Numerical Analysis II	123
6.3	MA352. Stochastic processes	127

7	Seventh Semester	131
7.1	MA322. Computational Number Theory	132
7.2	MA341. Optimization I	135
7.3	MA361. Mathematical Modeling	138
8	Eighth Semester	143
8.1	MA331. Scientific Computing	144
8.2	MA332. Fourier and Wavelet Analysis	148
8.3	MA355. Statistical Learning Theory	151
9	Ninth Semester	155
9.1	MA401. Capstone Project I	156
10	Tenth Semester	159
10.1	MA402. Capstone Project II	160

Chapter 1

First Semester



National University of San Marcos (UNMSM)
School of Scientific Computing
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) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

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

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

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

7. 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 (<i>Familiarity</i>)] • Identificar las contribuciones de varios pioneros en el campo de la computación [Familiarizarse (<i>Familiarity</i>)] • Discutir el contexto histórico de los paradigmas de diversos lenguajes de programación [Familiarizarse (<i>Familiarity</i>)] • Comparar la vida diaria antes y después de la llegada de los ordenadores personales y el Internet [Evaluar (<i>Assessment</i>)]
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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse (<i>Familiarity</i>)] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar (<i>Usage</i>)] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar (<i>Usage</i>)] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar (<i>Usage</i>)]
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 (<i>Assessment</i>)] • Identifica y describe el uso de tipos de datos primitivos [Familiarizarse (<i>Familiarity</i>)] • Escribe programas que usan tipos de datos primitivos [Usar (<i>Usage</i>)] • Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar (<i>Usage</i>)] • 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 (<i>Usage</i>)] • Escribe un programa que usa E/S de archivos para brindar persistencia a través de ejecuciones múltiples [Usar (<i>Usage</i>)] • Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Familiarizarse (<i>Familiarity</i>)] • Describe el concepto de recursividad y da ejemplos de su uso [Evaluar (<i>Assessment</i>)] • Identifica el caso base y el caso general de un problema basado en recursividad [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Familiarizarse (<i>Familiarity</i>)] • Indique la definición formal de Big O [Familiarizarse (<i>Familiarity</i>)] • Use la notación formal de la Big O para dar límites superiores asintóticos en la complejidad de tiempo y espacio de los algoritmos [Usar (<i>Usage</i>)] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] - Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar (<i>Assessment</i>)] - Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar (<i>Usage</i>)] - Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse (<i>Familiarity</i>)] - Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse (<i>Familiarity</i>)] - 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 (<i>Familiarity</i>)] - Explicar como el balanceamiento del arbol afecta la eficiencia de varias operaciones de un arbol de búsqueda binaria [Familiarizarse (<i>Familiarity</i>)] - Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar (<i>Usage</i>)] - 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 (<i>Assessment</i>)] - Describir la propiedad del heap y el uso de heaps como una implementación de colas de prioridad [Familiarizarse (<i>Familiarity</i>)] - 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 (<i>Usage</i>)] - Trazar y/o implementar un algoritmo de comparación de string [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Use encapsulation to create objects [Familiarizarse (<i>Familiarity</i>)].
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 (<i>Familiarity</i>)] • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse (<i>Familiarity</i>)] • Crea algoritmos para resolver problemas simples [Usar (<i>Usage</i>)] • Usa un lenguaje de programación para implementar, probar, y depurar algoritmos para resolver problemas simples [Usar (<i>Usage</i>)] • Implementa, prueba, y depura funciones recursivas simples y sus procedimientos [Usar (<i>Usage</i>)] • Determina si una solución iterativa o recursiva es la más apropiada para un problema [Evaluar (<i>Assessment</i>)] • Implementa un algoritmo de divide y vencerás para resolver un problema [Usar (<i>Usage</i>)] • Aplica técnicas de descomposición para dividir un programa en partes más pequeñas [Usar (<i>Usage</i>)] • Identifica los componentes de datos y el comportamiento de múltiples tipos de datos abstractos [Usar (<i>Usage</i>)] • Implementa un tipo de dato abstracto coherente, con la menor pérdida de acoplamiento entre componentes y comportamientos [Usar (<i>Usage</i>)] • Identifica las fortalezas y las debilidades relativas entre múltiples diseños e implementaciones de un problema [Evaluar (<i>Assessment</i>)]
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 (<i>Familiarity</i>)]
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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA111-UNMSM-FCM. Calculus I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA111-UNMSM-FCM. Calculus 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

This course introduces the fundamental concepts of differential and integral calculus of single-variable real functions. It provides the mathematical foundations necessary for analyzing problems in science and engineering, developing logical reasoning and problem-solving skills through limits, derivatives, and integrals. The course emphasizes both conceptual understanding and practical application of these concepts.

5. GOALS

- Understand the fundamental concepts of limits, continuity, and derivatives
- Apply differentiation techniques to optimization problems and rates of change
- Develop skills in function integration and application of definite integrals
- Solve practical problems using differential and integral calculus concepts

6. COMPETENCES

AG-Ci09 Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10 Formula hipótesis matemáticas. (Usage)

AG-Ci11 Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Aritmética Computacional y Análisis de Error (25 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [Ste16], [LE18]	

Unit 2: Búsqueda de Raíces y Optimización Numérica Básica (25 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [Ste16], [TWH14]	

Unit 3: Diferenciación e Integración Numérica (25 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) - Extrapolación de Richardson - Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) - Cuadratura gaussiana - Análisis de error para integración numérica	- Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] - Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] - Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] - Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] - Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [Ste16], [TWH14]	

Unit 4: Interpolación y Aproximación de Funciones (25 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
- Interpolación polinomial (formas de Lagrange, Newton) - Interpolación por splines (splines cúbicos) - Aproximación por mínimos cuadrados - Fenómeno de Runge y nodos de Chebyshev - Análisis de error para interpolación	- Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] - Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] - Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] - Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] - Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [Ste16], [TWH14]	

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

[TWH14] George B. Thomas, Maurice D. Weir, and Joel Hass. *Cálculo de una variable*. 13th. Pearson, 2014.

- [Ste16] James Stewart. *Cálculo de una variable: Trascendentes tempranas*. 8th. Cengage Learning, 2016.
- [LE18] Ron Larson and Bruce H. Edwards. *Cálculo*. 10th. Cengage Learning, 2018.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA121-UNMSM-FCM. Linear Algebra I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	:	MA121-UNMSM-FCM. Linear Algebra 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

This course introduces the fundamental concepts of linear algebra, providing the mathematical foundations for the study of vector spaces, linear transformations, and systems of linear equations. It develops abstract thinking and problem-solving skills through matrices, determinants, and vectors. The course is fundamental for applications in science, engineering, and computing.

5. GOALS

- Understand the concepts of vector spaces and subspaces
- Master matrix operations and determinants
- Solve systems of linear equations using different methods
- Apply linear transformations and compute eigenvalues and eigenvectors

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Álgebra Lineal Numérica (25 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Descomposición LU y sus aplicaciones • Descomposición QR y problemas de mínimos cuadrados • Descomposición en Valores Singulares (SVD) y sus aplicaciones • Número de condición y estabilidad numérica de sistemas lineales • Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)	• Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar (<i>Usage</i>)] • Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse (<i>Familiarity</i>)] • Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar (<i>Usage</i>)] • Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar (<i>Assessment</i>)] • Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar (<i>Assessment</i>)]
Readings : [LLM16], [Str16]	

Unit 2: Aritmética Computacional y Análisis de Error (25 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [LLM16], [Str16]	

Unit 3: Búsqueda de Raíces y Optimización Numérica Básica (25 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [LLM16], [Str16]	

Unit 4: Interpolación y Aproximación de Funciones (25 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [LLM16], [AR14]	

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

- [AR14] Howard Anton and Chris Rorres. *Álgebra Lineal con Aplicaciones*. 11th. Limusa Wiley, 2014.
- [LLM16] David C. Lay, Steven R. Lay, and Judi J. McDonald. *Álgebra Lineal y sus Aplicaciones*. 5th. Pearson, 2016.
- [Str16] Gilbert Strang. *Introduction to Linear Algebra*. 5th. Wellesley-Cambridge Press, 2016.



National University of San Marcos (UNMSM)

School of Scientific Computing

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) Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

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

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

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

7. 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 San Marcos (UNMSM)
School of Scientific Computing
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) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

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

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

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

7. 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 (<i>Usage</i>)] • Identify distortions of Computing in real-life situations. [Usar (<i>Usage</i>)] • Clearly identify at least 3 contexts of using the word“Engineer” in English. [Evaluar (<i>Assessment</i>)] • Identify the limitations of humans in solving computational problems. [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Understand the internal representation of characters in the ASCII and UTF-8 tables. [Familiarizarse (<i>Familiarity</i>)] • Understand the representation of colors in an image. [Familiarizarse (<i>Familiarity</i>)] • Apply Divide and Conquer algorithmic strategy. [Usar (<i>Usage</i>)] • Determine basic analysis of algorithmic complexity. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • 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 (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Usar (<i>Usage</i>)] • Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Explicar como los protocolos de intercambio de claves trabajan y como es que pueden fallar [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse (<i>Familiarity</i>)] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse (<i>Familiarity</i>)] • Describa como el uso de jerarquía de memoria (caché, memoria virtual) es aplicado para reducir el retardo efectivo en la memoria [Familiarizarse (<i>Familiarity</i>)] • Explique el efecto del retardo de la memoria en tiempo de ejecución [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Analizar las ventajas y desventajas inherentes en el diseño de un sistema operativo [Usar (<i>Usage</i>)] • Describir las funciones de un sistema operativo contemporáneo respecto a conveniencia, eficiencia, y su habilidad para evolucionar [Familiarizarse (<i>Familiarity</i>)] • Discutir acerca de sistemas operativos cliente-servidor, en red, distribuidos y cómo se diferencian de los sistemas operativos de un solo usuario [Familiarizarse (<i>Familiarity</i>)] • Identificar amenazas potenciales a sistemas operativos y las características del diseño de seguridad para protegerse de ellos [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Listar y definir la terminología de red apropiada [Familiarizarse (<i>Familiarity</i>)] • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse (<i>Familiarity</i>)] • Identificar los diferentes tipos de complejidad en una red (bordes, núcleo, etc.) [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Listar los factores que afectan al rendimiento de los protocolos de entrega fiables [Familiarizarse (<i>Familiarity</i>)] • Diseñar e implementar un protocolo confiable simple [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • En el contexto de a algoritmos específicos, identifique las características de data y/o otras condiciones o suposiciones que lleven a diferentes comportamientos [Evaluar (<i>Assessment</i>)] • Determine informalmente el tiempo y el espacio de complejidad de simples algoritmos [Usar (<i>Usage</i>)] • Indique la definición formal de Big O [Familiarizarse (<i>Familiarity</i>)] • Lista y contraste de clases estándares de complejidad [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Assessment</i>)] • Da ejemplos que ilustran las compensaciones entre espacio y tiempo que se dan en los algoritmos [Familiarizarse (<i>Familiarity</i>)] • Usar la notación formal Big O para dar límites de casos esperados en el tiempo de complejidad de los algoritmos [Usar (<i>Usage</i>)] • Explicar el uso de la notación theta grande, omega grande y o pequeña para describir la cantidad de trabajo hecho por un algoritmo [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] - Resolver problemas usando algoritmos básicos de grafos, incluyendo búsqueda por profundidad y búsqueda por amplitud [Usar (<i>Usage</i>)] - Implementar algoritmos numéricos básicos [Usar (<i>Usage</i>)] - Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Evaluar (<i>Assessment</i>)] - Ser capaz de implementar algoritmos de ordenamiento comunes cuadráticos y $O(N \log N)$ [Usar (<i>Usage</i>)] - Describir la implementación de tablas hash, incluyendo resolución y el evitamiento de colisiones [Familiarizarse (<i>Familiarity</i>)] - Discutir el tiempo de ejecución y eficiencia de memoria de los principales algoritmos de ordenamiento, búsqueda y hashing [Familiarizarse (<i>Familiarity</i>)] - 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 (<i>Familiarity</i>)] - Explicar como el balanceamiento del árbol afecta la eficiencia de varias operaciones de un árbol de búsqueda binaria [Familiarizarse (<i>Familiarity</i>)] - 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 (<i>Assessment</i>)]
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 (<i>Familiarity</i>)] • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse (<i>Familiarity</i>)] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse (<i>Familiarity</i>)] • Cita las metas básicas, funciones y modelos de un sistema de bases de datos [Familiarizarse (<i>Familiarity</i>)] • Describe los componentes de un sistema de bases de datos y da ejemplos de su uso [Familiarizarse (<i>Familiarity</i>)] • Identifica las funciones principales de un SGBD y describe sus roles en un sistema de bases de datos [Familiarizarse (<i>Familiarity</i>)] • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse (<i>Familiarity</i>)] • Usa un lenguaje de consulta declarativo para recoger información de una base de datos [Usar (<i>Usage</i>)] • Describe las capacidades que las bases de datos brindan al apoyar estructuras y/o la secuencia de flujo de datos, ejm. texto [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Usage</i>)] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar (<i>Usage</i>)] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar (<i>Usage</i>)] • 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 (<i>Assessment</i>)] • 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 (<i>Familiarity</i>)] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar (<i>Usage</i>)] • 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 (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Familiarizarse (<i>Familiarity</i>)] • Diferenciar entre las fases de desarrollo de software [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Explicar el papel de los modelos de madurez de procesos en la mejora de procesos [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Usage</i>)] • Definir la calidad del software y describir el papel de las actividades de aseguramiento de la calidad en el proceso de software [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)]
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 (<i>Usage</i>)] • Describe el rol de las heurísticas y describe los intercambios entre completitud, óptimo, complejidad de tiempo, y complejidad de espacio [Familiarizarse (<i>Familiarity</i>)] • Describe el problema de la explosión combinatoria del espacio de búsqueda y sus consecuencias [Familiarizarse (<i>Familiarity</i>)] • Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar (<i>Usage</i>)] • Selecciona e implementa un apropiado algoritmo de búsqueda no informado para un problema, y describe sus complejidades de tiempo y espacio [Usar (<i>Usage</i>)] • Evalúa si una heurística dada para un determinado problema es admisible/puede garantizar una solución óptima [Evaluar (<i>Assessment</i>)] • Compara y contrasta tópicos de búsqueda básica con temas jugabilidad de juegos [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Identificar ejemplos de tareas de clasificación, considerando las características de entrada disponibles y las salidas a ser predecidas [Familiarizarse (<i>Familiarity</i>)] • Describir el sobre ajuste (<i>overfitting</i>) en el contexto de un problema [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Explicar cómo las limitaciones en la percepción humana afectan la selección de la representación digital de señales analógicas [Usar (<i>Usage</i>)] • 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 (<i>Familiarity</i>)] • Describir modelos de color y su uso en los dispositivos de visualización de gráficos [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Describir cómo el doble buffer puede eliminar el parpadeo de la animación [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Describir la tubería básica gráficos y cómo el factor de representación va hacia adelante y atrás en esta [Familiarizarse (<i>Familiarity</i>)] • Crear un programa para visualizar modelos 3D de imágenes gráficas simples [Usar (<i>Usage</i>)] • Derivar la perspectiva lineal de triángulos semejantes por conversión de puntos (x,y,z) a puntos (x/z, y/z, 1) [Usar (<i>Usage</i>)] • Obtener puntos en 2-dimensiones y 3-dimensiones por aplicación de transformaciones afín [Usar (<i>Usage</i>)] • 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 (<i>Usage</i>)] • Explicar la dualidad de rastreo de rayos/rasterización para el problema de visibilidad [Familiarizarse (<i>Familiarity</i>)] • Implementar simples procedimientos que realicen la transformación y las operaciones de recorte de imágenes simples en 2 dimensiones [Usar (<i>Usage</i>)] • Calcular las necesidades de espacio en base a la resolución y codificación de color [Evaluar (<i>Assessment</i>)] • Calcular los requisitos de tiempo sobre la base de las frecuencias de actualización, técnicas de rasterización [Evaluar (<i>Assessment</i>)]
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 (<i>Usage</i>)] • Correctly apply data structures to solve the problem [Usar (<i>Usage</i>)] • Apply concepts related to algorithmic complexity in a search engine [Usar (<i>Usage</i>)].
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 San Marcos (UNMSM)
School of Scientific Computing
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) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

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

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

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

7. 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 (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Diferenciar emulación y el aislamiento [Familiarizarse (<i>Familiarity</i>)] • Evaluar virtualización de compensaciones [Evaluar (<i>Assessment</i>)]
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 (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Describir ejemplos de errores de programa detectadas por un sistema de tipos [Familiarizarse (<i>Familiarity</i>)] • Para múltiples lenguajes de programación, identificar propiedades de un programa con verificación estática y propiedades de un programa con verificación dinámica [Usar (<i>Usage</i>)] • Dar un ejemplo de un programa que no verifique tipos en un lenguaje particular y sin embargo no tenga error cuando es ejecutado [Familiarizarse (<i>Familiarity</i>)] • Usar tipos y mensajes de error de tipos para escribir y depurar programas [Usar (<i>Usage</i>)] • Explicar como las reglas de tipificación definen el conjunto de operaciones que legales para un tipo [Familiarizarse (<i>Familiarity</i>)] • Escribir las reglas de tipo que rigen el uso de un particular tipo compuesto [Usar (<i>Usage</i>)] • Explicar por qué indecidibilidad requiere sistemas de tipo para conservadoramente aproximar el comportamiento de un programa [Familiarizarse (<i>Familiarity</i>)] • Definir y usar piezas de programas (tales como, funciones, clases, métodos) que usan tipos genéricos, incluyendo para colecciones [Usar (<i>Usage</i>)] • Discutir las diferencias entre, genéricos (<i>generics</i>), subtipo y sobrecarga [Familiarizarse (<i>Familiarity</i>)] • Explicar múltiples beneficios y limitaciones de tipificación estática en escritura, mantenimiento y depuración de un software [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Assessment</i>)] • Identifica y describe el uso de tipos de datos primitivos [Familiarizarse (<i>Familiarity</i>)] • Escribe programas que usan tipos de datos primitivos [Usar (<i>Usage</i>)] • Modifica y expande programas cortos que usen estructuras de control condicionales e iterativas así como funciones [Usar (<i>Usage</i>)] • 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 (<i>Usage</i>)] • Escoje estructuras de condición y repetición adecuadas para una tarea de programación dada [Evaluar (<i>Assessment</i>)]
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 (<i>Usage</i>)] • Understand and apply the concept of parameter passing to a function, both by value and by reference. [Usar (<i>Usage</i>)] • Identify and apply the concept of function overloading. [Usar (<i>Usage</i>)] • Describe el concepto de recursividad y da ejemplos de su uso [Familiarizarse (<i>Familiarity</i>)] • Design, implement, and apply the concept of templates to create generic functions. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Design and apply the concept of multidimensional arrays. [Usar (<i>Usage</i>)] • Understand and apply the concept of references and pointers. [Familiarizarse (<i>Familiarity</i>)] • Understand, apply, and evaluate the relationship between pointers and arrays. [Evaluar (<i>Assessment</i>)] • Understand and implement dynamic memory management. Differentiate between heap and stack memory regions. [Evaluar (<i>Assessment</i>)] • Design, implement, and evaluate concepts like pointer-to-pointer, pointer-to-function, among other advanced pointer concepts. [Evaluar (<i>Assessment</i>)]
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 (<i>Usage</i>)] • Demonstrate the layout of an array in memory and how pointers are manipulated within those memory spaces. [Usar (<i>Usage</i>)] • Demonstrate the use of pointer arithmetic with arrays.[Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Demonstrate the use of functions and operators for dynamic memory allocation and deallocation.[Usar (<i>Usage</i>)] • Understand the implications of returning pointers from functions. [Usar (<i>Usage</i>)] • Use pointers to functions as parameters. [Usar (<i>Usage</i>)] • Understand the implications of dynamic memory usage and memory leaks. [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Understand the use of pointers to static members of a class. [Usar (<i>Usage</i>)] • Introduce the node structure and its use in a simple data structure. [Usar (<i>Usage</i>)] • Introduce data structures, showing a simple implementation of linked lists.[Usar (<i>Usage</i>)]
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 estandar. • Asignación dinámica: definición de método de llamada.	• Diseñar e implementar una clase [Usar (<i>Usage</i>)] • Usar subclase para diseñar una jerarquía simple de clases que permita al código ser reusable por diferentes subclases [Usar (<i>Usage</i>)] • Razonar correctamente sobre el flujo de control en un programa mediante el envío dinámico [Usar (<i>Usage</i>)] • Comparar y contrastar (1) el enfoque procedur/funcional- definiendo una función por cada operación con el cuerdo de la función proporcionando un caso por cada variación de dato - y (2) el enfoque orientado a objetos - definiendo una clase por cada variación de dato con la definición de la clase proporcionando un método por cada operación. Entender ambos enfoques como una definición de variaciones y operaciones de una matriz [Evaluar (<i>Assessment</i>)] • 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 (<i>Familiarity</i>)] • Usar mecanismos de encapsulación orientada a objetos, tal como interfaces y miembros privados [Usar (<i>Usage</i>)] • Definir y usar iteradores y otras operaciones sobre agregaciones, incluyendo operaciones que tienen funciones como argumentos, en múltiples lenguajes de programación, seleccionar la forma mas natural por cada lenguaje [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Implement and create new generic data types. [Usar (<i>Usage</i>)] • Understand the basic structures of the STL. [Familiarizarse (<i>Familiarity</i>)] • Use basic data structures like stack, queue, list, and vector from the STL. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Implement the overloading of allowed operators in the programming language. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Create programs to read and write files. [Usar (<i>Usage</i>)]
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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

CS1D1-CS2023. Discrete Structures I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS1D1-CS2023. Discrete Structures 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	: 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) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

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

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

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

7. TOPICS

Unit 1: Funciones, relaciones y conjuntos (22 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Conjuntos: – Diagramas de Venn – Unión, intersección, complemento – Producto Cartesiano – Potencia de conjuntos – Cardinalidad de Conjuntos finitos • Relations: – Reflexivity, simmetry, 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 (<i>Assessment</i>)] • Realizar las operaciones asociadas con conjuntos, funciones y relaciones [Evaluar (<i>Assessment</i>)] • Relacionar ejemplos prácticos para conjuntos funciones o modelos de relación apropiados e interpretar la asociación de operaciones y terminología en contexto [Evaluar (<i>Assessment</i>)]
Readings : [Gri03], [Ros07], [Vel06]	

Unit 2: Lógica básica (14 hours)	
Competences Expected:	
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 (<i>Usage</i>)] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar (<i>Usage</i>)] • Usar reglas de inferencia para construir demostraciones en lógica proposicional y de predicados [Usar (<i>Usage</i>)] • 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 (<i>Familiarity</i>)] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Usar (<i>Usage</i>)] • Describir las fortalezas y limitaciones de la lógica proposicional y de predicados [Usar (<i>Usage</i>)]
Readings : [Ros07], [Gri03], [Vel06]	

Unit 3: Técnicas de demostración (14 hours)	
Competences Expected:	
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 (<i>Assessment</i>)] • Describir la estructura básica de cada técnica de demostración (demostración directa, demostración por contradicción e inducción) descritas en esta unidad [Usar (<i>Usage</i>)] • Aplicar las técnicas de demostración (demostración directa, demostración por contradicción e inducción) correctamente en la construcción de un argumento solido [Usar (<i>Usage</i>)] • Determine que tipo de demostración es la mejor para un problema dado [Evaluar (<i>Assessment</i>)] • Explicar el paralelismo entre ideas matemáticas y/o inducción estructural para la recursión y definir estructuras recursivamente [Familiarizarse (<i>Familiarity</i>)] • Explicar la relación entre inducción fuerte y débil y dar ejemplos del apropiado uso de cada uno [Evaluar (<i>Assessment</i>)] • Enunciar el principio del buen-orden y su relación con la inducción matemática [Familiarizarse (<i>Familiarity</i>)]
Readings : [Ros07], [Vel06], [Sch12], [Vel06]	

Unit 4: Data Representation (10 hours)	
Competences Expected:	
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 (<i>Assessment</i>)]. • Carry out arithmetic operations using different kinds of representations. [Evaluar (<i>Assessment</i>)]. • Explain the floating point standard IEEE-754 [Familiarizarse (<i>Familiarity</i>)].
Readings : [Ros07], [Gri03], [Vel06]	

8. WORKPLAN

8.1 Methodology

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

8.2 Theory Sessions

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

8.3 Practical Sessions

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

9. EVALUATION SYSTEM

***** EVALUATION MISSING *****

10. BASIC BIBLIOGRAPHY

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



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA112-UNMSM-FCM. Calculus II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA112-UNMSM-FCM. Calculus II
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	: MA111-UNMSM-FCM. Calculus I. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course continues the study of calculus with functions of several variables, integrating concepts of vector calculus, partial derivatives, multiple integrals, and geometric and physical applications. It develops skills for modeling and solving problems in three dimensions, providing the foundations for advanced areas such as mathematical analysis, physics, and engineering.

5. GOALS

- Understand functions of several variables and their partial derivatives
- Apply differential and integral calculus in three dimensions
- Solve optimization problems with multiple variables
- Master multiple integrals and their applications

6. COMPETENCES

AG-Ci09 Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10 Formula hipótesis matemáticas. (Usage)

AG-Ci11 Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Diferenciación e Integración Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [Ste16], [TWH14]	

Unit 2: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de Euler y sus variantes (explícito, implícito) • Métodos de Runge-Kutta • Métodos multipaso (Adams-Bashforth, Adams-Moulton) • Análisis de estabilidad y convergencia para solucionadores de EDOs • Sistemas de EDOs y EDOs de orden superior	• Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar (<i>Usage</i>)] • Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar (<i>Assessment</i>)] • Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar (<i>Assessment</i>)] • Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar (<i>Usage</i>)]
Readings : [Ste16], [TWH14]	

Unit 3: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [Ste16], [TWH14]	

Unit 4: Búsqueda de Raíces y Optimización Numérica Básica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [Ste16], [TWH14]	

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

[TWH14] George B. Thomas, Maurice D. Weir, and Joel Hass. *Cálculo de varias variables*. 13th. Pearson, 2014.

[Ste16] James Stewart. *Cálculo de varias variables: Trascendentes tempranas*. 8th. Cengage Learning, 2016.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA122-UNMSM-FCM. Linear Algebra II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA122-UNMSM-FCM. Linear Algebra II
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	: MA121-UNMSM-FCM. Linear Algebra I. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course deepens advanced concepts of linear algebra, focusing on inner product spaces, advanced linear transformations, matrix decompositions, and computational applications. It develops skills to analyze complex algebraic structures and apply matrix methods to problems in data science, signal processing, and optimization.

5. GOALS

- Master inner product spaces and their properties
- Apply advanced matrix decompositions (SVD, QR)
- Analyze linear transformations in inner product spaces
- Solve approximation and least squares problems

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Álgebra Lineal Numérica (16 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Descomposición LU y sus aplicaciones • Descomposición QR y problemas de mínimos cuadrados • Descomposición en Valores Singulares (SVD) y sus aplicaciones • Número de condición y estabilidad numérica de sistemas lineales • Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)	• Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar (<i>Usage</i>)] • Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse (<i>Familiarity</i>)] • Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar (<i>Usage</i>)] • Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar (<i>Assessment</i>)] • Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar (<i>Assessment</i>)]
Readings : [Str16], [LLM16]	

Unit 2: Aritmética Computacional y Análisis de Error (16 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [Str16], [LLM16]	

Unit 3: Interpolación y Aproximación de Funciones (16 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [Str16], [LLM16]	

Unit 4: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (16 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [Str16], [TI97]	

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

- [TI97] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM, 1997.
- [LLM16] David C. Lay, Steven R. Lay, and Judi J. McDonald. *Álgebra Lineal y sus Aplicaciones*. 5th. Pearson, 2016.
- [Str16] Gilbert Strang. *Introduction to Linear Algebra*. 5th. Wellesley-Cambridge Press, 2016.



National University of San Marcos (UNMSM)

School of Scientific Computing

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) Comunicarse efectivamente en diversos contextos profesionales. (Usage)

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

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

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

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

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

7. 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 San Marcos (UNMSM)
School of Scientific Computing
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) Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (Assessment)

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

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

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

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

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

7. 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 (<i>Usage</i>)] • Implement programs that use STL adapters. [Usar (<i>Usage</i>)] • Apply advanced STL algorithms. [Usar (<i>Usage</i>)] • Use functors and predicates with the STL. [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Understand and use SFINAE for template selection. [Usar (<i>Usage</i>)] • Use Perfect Forwarding for efficient argument passing. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Implement variadic functions. [Usar (<i>Usage</i>)] • Design classes with variadic methods. [Usar (<i>Usage</i>)] • Create classes with variadic templates. [Usar (<i>Usage</i>)] • Implement inheritance with variable lists of variadic templates. [Usar (<i>Usage</i>)] • Apply variadic templates to solve real-world problems. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] • Define and use rvalue references. [Usar (<i>Usage</i>)] • Analyze the performance implications of using move semantics. [Evaluar (<i>Assessment</i>)] • Implement move constructors and move assignment operators for custom classes. [Usar (<i>Usage</i>)] • Apply move semantics to optimize resource management in C++ programs. [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Understand and apply structural design patterns: Adapter, Decorator, Facade. [Usar (<i>Usage</i>)]
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 (<i>Usage</i>)] • Using Functors as parameters to functions and classes. [Usar (<i>Usage</i>)] • Using Functors in functions and classes through templates. [Usar (<i>Usage</i>)]
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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

CS1D2-CS2023. Discrete Structures II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: CS1D2-CS2023. Discrete Structures 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	: CS1D1-CS2023. Discrete Structures I. (2 nd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

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

5. GOALS

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

6. COMPETENCES

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

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

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

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

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

7. TOPICS

Unit 1: Digital Logic and Data Representation (10 hours)	
Competences Expected:	
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 (<i>Assessment</i>)]. • Explain the algebraic structures of reticulum and its types [Evaluar (<i>Assessment</i>)]. • Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Evaluar (<i>Assessment</i>)]. • Explain the properties that satisfies a Boolean algebra [Evaluar (<i>Assessment</i>)]. • Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Evaluar (<i>Assessment</i>)]. • Find the canonical forms of a Boolean function [Evaluar (<i>Assessment</i>)]. • Represent a Boolean function as a Boolean circuit using logic gates [Evaluar (<i>Assessment</i>)]. • Minimize a Boolean function. [Evaluar (<i>Assessment</i>)].
Readings : [Ros07], [Gri03]	

Unit 2: Fundamentos de conteo (40 hours)	
Competences Expected:	
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 (<i>Familiarity</i>)] • Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse (<i>Familiarity</i>)] • Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse (<i>Familiarity</i>)] • Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse (<i>Familiarity</i>)] • Realizar cálculos que involucren aritmética modular [Familiarizarse (<i>Familiarity</i>)]
Readings : [Gri97]	

Unit 3: Árboles y Grafos (40 hours)	
Competences Expected:	
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 (<i>Familiarity</i>)] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse (<i>Familiarity</i>)] • 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 (<i>Familiarity</i>)] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse (<i>Familiarity</i>)] • Explicar como construir un árbol de expansión de un grafo [Familiarizarse (<i>Familiarity</i>)] • Determinar si dos grafos son isomorfos [Familiarizarse (<i>Familiarity</i>)]
Readings : [Joh99]	

8. WORKPLAN

8.1 Methodology

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

8.2 Theory Sessions

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

8.3 Practical Sessions

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

9. EVALUATION SYSTEM

***** EVALUATION MISSING *****

10. BASIC BIBLIOGRAPHY

- [Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. 2007.



National University of San Marcos (UNMSM)

School of Scientific Computing

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

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

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

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

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

7. 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 (<i>Familiarity</i>)] • Listar las características de lenguajes de plataforma [Familiarizarse (<i>Familiarity</i>)] • Escribir y ejecutar un programa simple basado en plataforma [Familiarizarse (<i>Familiarity</i>)] • Listar las ventajas y desventajas de la programación con restricciones de plataforma [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Familiarity</i>)] • Describir las limitaciones que la web pone a los desarrolladores [Familiarizarse (<i>Familiarity</i>)] • Comparar y contrastar la programación web con la programación de propósito general [Familiarizarse (<i>Familiarity</i>)] • Describir las diferencias entre software como un servicio y productos de software tradicionales [Familiarizarse (<i>Familiarity</i>)] • Discutir cómo los estándares de web impactan el desarrollo de software [Familiarizarse (<i>Familiarity</i>)] • Revisar una aplicación web existente con un estándar web actual [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Usage</i>)] - Web programming approach using embedded python. [Usar (<i>Usage</i>)] - Accessing and Manipulating MySQL. [Usar (<i>Usage</i>)] - The Ajax web application development approach. [Usar (<i>Usage</i>)] - DOM and CSS used in JavaScript. [Usar (<i>Usage</i>)] - Asynchronous Content Update Technologies. [Usar (<i>Usage</i>)] - XMLHttpRequest objects use to communicate between clients and servers. [Usar (<i>Usage</i>)] - XML and JSON. [Usar (<i>Usage</i>)] - XSLT and XPath as mechanisms for transforming XML documents. [Usar (<i>Usage</i>)] - Web services and APIs (especially Google Maps). [Usar (<i>Usage</i>)] - Macros Ajax for the development of contemporary web applications. [Usar (<i>Usage</i>)] - Design patterns used in web applications. [Usar (<i>Usage</i>)]
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 (<i>Familiarity</i>)] - Discutir las limitaciones que las plataformas móviles ponen a los desarrolladores [Familiarizarse (<i>Familiarity</i>)] - Discutir el rendimiento vs pérdida de potencia [Familiarizarse (<i>Familiarity</i>)] - Compare y contraste la programación móvil con la programación de proposito general [Familiarizarse (<i>Familiarity</i>)]
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 (<i>Usage</i>)] • 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 (<i>Usage</i>)] • Students build applications that require starting multiple activities through both standard and custom methods. [Usar (<i>Usage</i>)] • Students build applications that require standard and custom permissions. [Usar (<i>Usage</i>)] • Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usar (<i>Usage</i>)] • 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 (<i>Usage</i>)] • Students build an application that uses location information to collect latitude, length of places they visit. [Usar (<i>Usage</i>)]
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 San Marcos (UNMSM)

School of Scientific Computing

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

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

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

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

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

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

7. 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 (<i>Usage</i>)]. • Analyze global intelligent systems and their distinctive characteristics [Evaluar (<i>Assessment</i>)]. • Critically evaluate AI outputs across different cultural contexts [Evaluar (<i>Assessment</i>)].
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 (<i>Assessment</i>)]. • Develop cross-platform workflows that integrate different AI tools [Usar (<i>Usage</i>)]. • Implement iterative refinement strategies to optimize AI outputs [Usar (<i>Usage</i>)].
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 (<i>Usage</i>)]. - Design solutions that integrate multiple intelligent systems for complex tasks [Evaluar (<i>Assessment</i>)]. - Develop practical applications using Western and Chinese AI tools [Usar (<i>Usage</i>)]. - Implement workflows that leverage specific strengths of each platform [Evaluar (<i>Assessment</i>)].
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 (<i>Assessment</i>)]. - Apply professional ethical principles in global AI contexts [Usar (<i>Usage</i>)]. - Develop ethical guidelines to mitigate biases in AI systems [Evaluar (<i>Assessment</i>)]. - Evaluate ethical implications of AI deployment across different cultures [Usar (<i>Usage</i>)].
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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA211-UNMSM-FCM. Calculus III (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA211-UNMSM-FCM. Calculus III
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	: MA112-UNMSM-FCM. Calculus II. (2 nd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course advances the study of multivariable calculus, integrating concepts of vector calculus, vector fields, line and surface integrals, and fundamental theorems of vector calculus. It develops skills to analyze scalar and vector fields in space, providing foundations for applications in physics, engineering, and computational sciences.

5. GOALS

- Understand the concepts of vector fields and differential operators
- Apply line and surface integrals to physical problems
- Master the fundamental theorems of vector calculus
- Solve problems of flux, circulation, and vector potentials

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Diferenciación e Integración Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [MT03], [Ste16]	

Unit 2: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de Euler y sus variantes (explícito, implícito) • Métodos de Runge-Kutta • Métodos multipaso (Adams-Bashforth, Adams-Moulton) • Análisis de estabilidad y convergencia para solucionadores de EDOs • Sistemas de EDOs y EDOs de orden superior	• Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar (<i>Usage</i>)] • Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar (<i>Assessment</i>)] • Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar (<i>Assessment</i>)] • Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar (<i>Usage</i>)]
Readings : [MT03], [Ste16]	

Unit 3: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [MT03], [Ste16]	

Unit 4: Búsqueda de Raíces y Optimización Numérica Básica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [MT03], [Ste16]	

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

[MT03] Jerrold E. Marsden and Anthony J. Tromba. *Vector Calculus*. 5th. Pearson, 2003.

[Ste16] James Stewart. *Multivariable Calculus: Early Transcendentals*. 8th. Cengage Learning, 2016.



National University of San Marcos (UNMSM)

School of Scientific Computing

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 llation of ideas .

6. COMPETENCES

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

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

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

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

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

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

7. 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 San Marcos (UNMSM)
School of Scientific Computing
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) Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. (Assessment)

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

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

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

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

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

7. 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 (<i>Usage</i>)] • Develop knowledge to decide when it is better to use one implementation technique than another. [Usar (<i>Usage</i>)]
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 (<i>Assessment</i>)]
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 (<i>Assessment</i>)]
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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA212-UNMSM-FCM. Differential Equations (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA212-UNMSM-FCM. Differential Equations
2.2 Semester	: 4 th 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	: MA211-UNMSM-FCM. Calculus III. (3 rd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces fundamental methods for solving ordinary and partial differential equations. It develops skills for modeling dynamic phenomena in physics, engineering, and sciences through differential equations, providing mathematical foundations for analyzing continuous systems and simulating natural processes.

5. GOALS

- Understand fundamental concepts of differential equations
- Solve first and second order differential equations
- Apply analytical and numerical methods to initial and boundary value problems
- Model physical and biological phenomena using differential equations

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Método de Euler y sus variantes (explícito, implícito) • Métodos de Runge-Kutta • Métodos multipaso (Adams-Bashforth, Adams-Moulton) • Análisis de estabilidad y convergencia para solucionadores de EDOs • Sistemas de EDOs y EDOs de orden superior	• Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar (<i>Usage</i>)] • Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar (<i>Assessment</i>)] • Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar (<i>Assessment</i>)] • Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar (<i>Usage</i>)]
Readings : [BDM17], [Zil18]	

Unit 2: Búsqueda de Raíces y Optimización Numérica Básica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [BDM17], [Zil18]	

Unit 3: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [BDM17], [Zil18]	

Unit 4: Diferenciación e Integración Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [BDM17], [Zil18]	

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

[BDM17] William E. Boyce, Richard C. DiPrima, and Douglas B. Meade. *Ecuaciones Diferenciales y Problemas con Valores en la Frontera*. 11th. Limusa Wiley, 2017.

[Zil18] Dennis G. Zill. *Ecuaciones Diferenciales con Aplicaciones de Modelado*. 11th. Cengage Learning, 2018.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA351-UNMSM-FCM. Statistics and Probabilities (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA351-UNMSM-FCM. Statistics and Probabilities
2.2 Semester	: 4 th 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	: MA211-UNMSM-FCM. Calculus III. (3 rd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamentals of probability theory and statistics, providing mathematical foundations for data analysis and statistical inference. It develops skills for modeling random phenomena, conducting hypothesis testing, and applying statistical methods to problems in science, engineering, and social sciences.

5. GOALS

- Understand fundamental concepts of probability and random variables
- Apply probability distributions to real-world problems
- Perform statistical inference and hypothesis testing
- Analyze data using parametric and non-parametric statistical methods

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Probabilidad Avanzada y Procesos Estocásticos (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Fundamentos de probabilidad medida-teórica - Variables aleatorias y distribuciones - Caracterización de procesos estocásticos - Cadenas y procesos de Markov - Movimiento browniano y procesos de difusión	- Aplicar conceptos medida-teóricos a espacios de probabilidad [Familiarizarse (<i>Familiarity</i>)] - Caracterizar diferentes tipos de procesos estocásticos [Evaluar (<i>Assessment</i>)] - Analizar cadenas de Markov para comportamiento transitorio y en estado estacionario [Usar (<i>Usage</i>)] - Modelar sistemas usando procesos estocásticos en tiempo continuo [Usar (<i>Usage</i>)]
Readings : [Dev16], [Wal+12]	

Unit 2: Inferencia Estadística Computacional (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
- Estimación de Máxima Verosimilitud (MLE) - Métodos de inferencia bayesiana - Métodos de Monte Carlo mediante Cadenas de Markov (MCMC) - Algoritmo Expectación-Maximización (EM) - Remuestreo bootstrap y jackknife	- Implementar Estimación de Máxima Verosimilitud para varias distribuciones [Usar (<i>Usage</i>)] - Aplicar métodos bayesianos para estimación de parámetros [Usar (<i>Usage</i>)] - Implementar algoritmos MCMC para muestreo posterior [Usar (<i>Usage</i>)] - Usar el algoritmo EM para modelos con variables latentes [Usar (<i>Usage</i>)] - Aplicar métodos bootstrap para intervalos de confianza [Usar (<i>Usage</i>)]
Readings : [Dev16], [Wal+12]	

Unit 3: Métodos de Remuestreo y Validación (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Estimación bootstrap e intervalos de confianza - Técnicas de validación cruzada - Pruebas de permutación y aleatorización - Selección de modelos y criterios de información - Métodos de ensamblado y bagging	- Implementar bootstrap para estimación de error estándar [Usar (<i>Usage</i>)] - Aplicar validación cruzada para evaluación de modelos [Usar (<i>Usage</i>)] - Usar pruebas de permutación para contraste de hipótesis [Usar (<i>Usage</i>)] - Comparar modelos usando criterios de información (AIC, BIC) [Evaluar (<i>Assessment</i>)] - Implementar bagging para reducción de varianza [Usar (<i>Usage</i>)]
Readings : [Dev16], [Wal+12]	

Unit 4: Diseño de Experimentos Computacionales (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Diseños factoriales y factoriales fraccionados • Metodología de superficie de respuesta • Diseños de llenado de espacio (hipercubo latino) • Diseño óptimo de experimentos • Experimentos computacionales y modelado sustituto	• Diseñar experimentos factoriales para screening de factores [Usar (<i>Usage</i>)] • Aplicar metodología de superficie de respuesta para optimización [Usar (<i>Usage</i>)] • Implementar diseños de llenado de espacio para experimentos computacionales [Usar (<i>Usage</i>)] • Seleccionar diseños óptimos para objetivos específicos [Evaluar (<i>Assessment</i>)] • Construir modelos sustitutos para simulaciones costosas [Usar (<i>Usage</i>)]
Readings : [Dev16], [Wal+12]	

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

[Wal+12] Ronald E. Walpole et al. *Probabilidad y Estadística para Ingeniería y Ciencias*. 9th. Pearson, 2012.

[Dev16] Jay L. Devore. *Probabilidad y Estadística para Ingeniería y Ciencias*. 9th. Cengage Learning, 2016.



National University of San Marcos (UNMSM)

School of Scientific Computing

Syllabus 2026-I

1. COURSE

ID104. Technical and professional English IV (Mandatory)

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	: Mandatory
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) Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

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

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

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

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

7. 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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA221-UNMSM-FCM. Numerical Analysis I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA221-UNMSM-FCM. Numerical Analysis I
2.2 Semester	: 5 th Semester
2.3 Credits	: 5
2.4 Horas	: 3 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	: MA211-UNMSM-FCM. Calculus III. (3 rd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces fundamental numerical methods for solving mathematical problems through computational algorithms. It develops skills for implementing and analyzing numerical methods in linear algebra, nonlinear equations, and interpolation, with emphasis on error analysis and numerical stability.

5. GOALS

- Understand fundamentals of computational arithmetic and error analysis
- Implement numerical methods for linear systems of equations
- Solve nonlinear equations using iterative methods
- Apply function interpolation and approximation techniques

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Aritmética Computacional y Análisis de Error (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [BFB16], [CC15]	

Unit 2: Álgebra Lineal Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Descomposición LU y sus aplicaciones • Descomposición QR y problemas de mínimos cuadrados • Descomposición en Valores Singulares (SVD) y sus aplicaciones • Número de condición y estabilidad numérica de sistemas lineales • Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)	• Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar (<i>Usage</i>)] • Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse (<i>Familiarity</i>)] • Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar (<i>Usage</i>)] • Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar (<i>Assessment</i>)] • Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar (<i>Assessment</i>)]
Readings : [BFB16], [CC15]	

Unit 3: Búsqueda de Raíces y Optimización Numérica Básica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [BFB16], [CC15]	

Unit 4: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [BFB16], [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

- [CC15] Steven C. Chapra and Raymond P. Canale. *Métodos Numéricos para Ingenieros*. 7th. McGraw-Hill, 2015.
- [BFB16] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Análisis Numérico*. 10th. Cengage Learning, 2016.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA222-UNMSM-FCM. Real Analysis I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA222-UNMSM-FCM. Real Analysis I
2.2 Semester	: 5 th 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	: MA112-UNMSM-FCM. Calculus II. (2 nd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamentals of real mathematical analysis, providing rigorous foundations for differential and integral calculus. It develops logical reasoning and mathematical proof skills through the study of real numbers, sequences, series, and continuity, establishing theoretical foundations for advanced areas of mathematics.

5. GOALS

- Understand the structure of real numbers and their properties
- Analyze convergence of sequences and numerical series
- Master concepts of limits and continuity of functions
- Develop skills in rigorous mathematical proof

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Aritmética Computacional y Análisis de Error (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [Rud76], [BS11]	

Unit 2: Búsqueda de Raíces y Optimización Numérica Básica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Método de bisección e iteración de punto fijo • Método de Newton y método de la secante para búsqueda de raíces • Análisis de convergencia y orden de convergencia • Descenso de gradiente para optimización univariada • Tasa de convergencia y criterios de parada	• Implementar algoritmos de búsqueda de raíces (bisección, Newton, secante) y analizar su convergencia [Usar (<i>Usage</i>)] • Clasificar el orden de convergencia de métodos iterativos [Evaluar (<i>Assessment</i>)] • Aplicar descenso de gradiente para resolver problemas simples de optimización univariada [Usar (<i>Usage</i>)] • Diseñar criterios de parada apropiados para métodos iterativos [Usar (<i>Usage</i>)]
Readings : [Rud76], [BS11]	

Unit 3: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [Rud76], [BS11]	

Unit 4: Diferenciación e Integración Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [Rud76], [BS11]	

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

[Rud76] Walter Rudin. *Principles of Mathematical Analysis*. 3rd. McGraw-Hill, 1976.

[BS11] Robert G. Bartle and Donald R. Sherbert. *Introduction to Real Analysis*. 4th. Wiley, 2011.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA223-UNMSM-FCM. Graph Theory (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA223-UNMSM-FCM. Graph Theory
2.2 Semester	: 5 th 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	: CS1D2-CS2023. Discrete Structures II. (3 rd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamentals of graph theory, providing mathematical foundations for the study of discrete structures and their applications in computing, networks, and optimization. It develops skills for modeling real-world problems using graphs and applying algorithms to solve connectivity, traversal, and optimization problems in networks.

5. GOALS

- Understand fundamental concepts of graphs and their properties
- Apply graph traversal and search algorithms
- Solve shortest path and spanning tree problems
- Model combinatorial optimization problems using graph theory

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Algoritmos de Grafos y Optimización Combinatoria (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Representación de grafos y recorrido (BFS, DFS) • Algoritmos de camino más corto (Dijkstra, Bellman-Ford) • Árboles de expansión mínima (Prim, Kruskal) • Flujo en redes y emparejamiento • NP-completitud y algoritmos de aproximación	• Implementar algoritmos de recorrido de grafos [Usar (<i>Usage</i>)] • Resolver problemas de camino más corto usando algoritmos apropiados [Usar (<i>Usage</i>)] • Construir árboles de expansión mínima [Usar (<i>Usage</i>)] • Analizar complejidad de algoritmos de grafos [Evaluar (<i>Assessment</i>)] • Diseñar algoritmos de aproximación para problemas NP-difíciles [Usar (<i>Usage</i>)]
Readings : [BM08], [Wes01]	

Unit 2: Complejidad Algorítmica y Análisis Asintótico (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Medidas de complejidad y notación asintótica • Análisis del mejor caso, caso promedio y peor caso • Relaciones de recurrencia y técnicas de resolución • Análisis empírico de algoritmos • Complejidad espacial y compensaciones	• Analizar algoritmos usando notación asintótica (Big O, Omega, Theta) [Usar (<i>Usage</i>)] • Diferenciar entre complejidades del mejor caso, caso promedio y peor caso [Evaluar (<i>Assessment</i>)] • Resolver relaciones de recurrencia para análisis de algoritmos [Usar (<i>Usage</i>)] • Comparar algoritmos mediante mediciones empíricas [Evaluar (<i>Assessment</i>)]
Readings : [BM08], [Wes01]	

Unit 3: Estructuras de Datos Fundamentales y Avanzadas (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Arreglos, listas enlazadas y arreglos dinámicos • Pilas, colas y deque • Tablas hash y resolución de colisiones • Árboles y árboles de búsqueda binaria • Montículos y colas de prioridad	• Implementar estructuras de datos básicas desde cero [Usar (<i>Usage</i>)] • Analizar complejidad temporal/espacial de operaciones de estructuras de datos [Evaluar (<i>Assessment</i>)] • Seleccionar estructuras de datos apropiadas para problemas dados [Evaluar (<i>Assessment</i>)] • Manejar colisiones hash usando diferentes estrategias [Usar (<i>Usage</i>)]
Readings : [BM08], [Wes01]	

Unit 4: Algoritmos de Búsqueda y Ordenamiento (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Búsqueda lineal y binaria • Ordenamiento basado en comparación (QuickSort, MergeSort, HeapSort) • Ordenamiento sin comparación (RadixSort, CountingSort) • Algoritmos de selección y estadísticas de orden • Ordenamiento externo y procesamiento a gran escala	• Implementar y comparar algoritmos de búsqueda [Usar (<i>Usage</i>)] • Analizar complejidad temporal/espacial de algoritmos de ordenamiento [Evaluar (<i>Assessment</i>)] • Seleccionar algoritmo de ordenamiento apropiado para características de datos [Evaluar (<i>Assessment</i>)] • Implementar algoritmos de estadísticas de orden [Usar (<i>Usage</i>)]
Readings : [BM08], [Wes01]	

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

[Wes01] Douglas B. West. *Introduction to Graph Theory*. 2nd. Prentice Hall, 2001.

[BM08] J. A. Bondy and U. S. R. Murty. *Graph Theory*. 1st. Springer, 2008.



National University of San Marcos (UNMSM)

School of Scientific Computing

Syllabus 2026-I

1. COURSE

ID105. Technical and professional English V (Mandatory)

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	: Mandatory
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) Comunicarse efectivamente en diversos contextos profesionales. (Assessment)

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

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

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

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

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

7. 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 • Patronos 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 San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA321-UNMSM-FCM. Abstract algebra (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA321-UNMSM-FCM. Abstract algebra
2.2 Semester	: 6 th 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

This course introduces the fundamental concepts of abstract algebra, providing foundations for the study of algebraic structures such as groups, rings, and fields. It develops abstract thinking and mathematical proof skills through the analysis of algebraic properties and their applications in number theory, cryptography, and geometry.

5. GOALS

- Understand fundamental concepts of groups and their properties
- Analyze ring and field structures
- Apply fundamental theorems of abstract algebra
- Develop skills in proving algebraic theorems

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Teoría de Grupos y Algoritmos Algebraicos (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Fundamentos de teoría de grupos y representación computacional - Algoritmos algebraicos para operaciones de grupos - Grupos de permutación y simetría computacional - Aplicaciones en criptografía y teoría de códigos - Complejidad de algoritmos algebraicos	- Implementar operaciones fundamentales de grupos computacionalmente [Usar (<i>Usage</i>)] - Analizar algoritmos algebraicos para teoría de grupos [Evaluar (<i>Assessment</i>)] - Aplicar grupos de permutación a problemas de simetría [Usar (<i>Usage</i>)] - Usar teoría de grupos en aplicaciones criptográficas [Usar (<i>Usage</i>)]
Readings : [DF04], [Rot10]	

Unit 2: Teoría de Números Computacional y Criptografía (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
- Aritmética modular y algoritmos - Pruebas de primalidad y factorización - Criptografía de curva elíptica - Problema del logaritmo discreto - Fundamentos de criptografía post-cuántica	- Implementar algoritmos de aritmética modular [Usar (<i>Usage</i>)] - Aplicar métodos de prueba de primalidad [Usar (<i>Usage</i>)] - Usar curvas elípticas en protocolos criptográficos [Usar (<i>Usage</i>)] - Analizar sistemas basados en logaritmo discreto [Evaluar (<i>Assessment</i>)]
Readings : [DF04], [Rot10]	

Unit 3: Sistemas de Computación Simbólica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Sistemas de álgebra computacional (Mathematica, SymPy) - Integración y diferenciación simbólica - Resolución de ecuaciones y simplificación - Demostración de teoremas y razonamiento automatizado - Aplicaciones en computación científica	- Usar sistemas de álgebra computacional para computación matemática [Usar (<i>Usage</i>)] - Implementar algoritmos de computación simbólica [Usar (<i>Usage</i>)] - Aplicar técnicas de demostración de teoremas [Usar (<i>Usage</i>)] - Conectar computación simbólica y numérica [Evaluar (<i>Assessment</i>)]
Readings : [DF04], [Rot10]	

Unit 4: Lógica Computacional y Verificación Formal (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Lógica proposicional y de predicados • Solucionadores SAT y satisfacibilidad • Verificación de modelos y lógica temporal • Verificación de programas y lógica de Hoare • Demostradores de teoremas interactivos	• Aplicar lógica computacional a resolución de problemas [Usar (<i>Usage</i>)] • Usar solucionadores SAT para satisfacción de restricciones [Usar (<i>Usage</i>)] • Implementar algoritmos de verificación de modelos [Usar (<i>Usage</i>)] • Verificar programas usando métodos formales [Usar (<i>Usage</i>)]
Readings : [DF04], [Rot10]	

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

[DF04] David S. Dummit and Richard M. Foote. *Abstract Algebra*. 3rd. Wiley, 2004.

[Rot10] Joseph J. Rotman. *Advanced Modern Algebra*. 3rd. American Mathematical Society, 2010.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA325-UNMSM-FCM. Numerical Analysis II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA325-UNMSM-FCM. Numerical Analysis II
2.2 Semester	: 6 th Semester
2.3 Credits	: 5
2.4 Horas	: 3 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	: MA221-UNMSM-FCM. Numerical Analysis I. (5 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course advances the study of numerical methods for advanced mathematical problems, focusing on differential equations, numerical integration, and eigenvalue problems. It develops skills for implementing and analyzing sophisticated numerical algorithms with applications in engineering and computational sciences, with emphasis on stability and computational efficiency.

5. GOALS

- Solve ordinary and partial differential equations numerically
- Apply advanced numerical integration and differentiation methods
- Analyze eigenvalue problems and matrix decompositions
- Evaluate stability and convergence of advanced numerical methods

6. COMPETENCES

AG-Ci09 Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10 Formula hipótesis matemáticas. (Usage)

AG-Ci11 Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Método de Euler y sus variantes (explícito, implícito) • Métodos de Runge-Kutta • Métodos multipaso (Adams-Bashforth, Adams-Moulton) • Análisis de estabilidad y convergencia para solucionadores de EDOs • Sistemas de EDOs y EDOs de orden superior	• Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar (<i>Usage</i>)] • Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar (<i>Assessment</i>)] • Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar (<i>Assessment</i>)] • Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar (<i>Usage</i>)]
Readings : [BFB16], [CC15]	

Unit 2: Diferenciación e Integración Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [BFB16], [CC15]	

Unit 3: Álgebra Lineal Numérica (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Descomposición LU y sus aplicaciones • Descomposición QR y problemas de mínimos cuadrados • Descomposición en Valores Singulares (SVD) y sus aplicaciones • Número de condición y estabilidad numérica de sistemas lineales • Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)	• Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar (<i>Usage</i>)] • Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse (<i>Familiarity</i>)] • Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar (<i>Usage</i>)] • Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar (<i>Assessment</i>)] • Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar (<i>Assessment</i>)]
Readings : [BFB16], [CC15]	

Unit 4: Interpolación y Aproximación de Funciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [BFB16], [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

[CC15] Steven C. Chapra and Raymond P. Canale. *Métodos Numéricos para Ingenieros*. 7th. McGraw-Hill, 2015.

- [BFB16] Richard L. Burden, J. Douglas Faires, and Annette M. Burden. *Análisis Numérico*. 10th. Cengage Learning, 2016.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA352-UNMSM-FCM. Stochastic processes (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA352-UNMSM-FCM. Stochastic processes
2.2 Semester	: 6 th 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	: MA351-UNMSM-FCM. Statistics and Probabilities. (4 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the theory of stochastic processes, providing mathematical foundations for the study of systems that evolve randomly over time. It develops skills for modeling random phenomena in finance, engineering, and sciences through Markov chains, Poisson processes, and Brownian motion, with applications in simulation and analysis of stochastic systems.

5. GOALS

- Understand fundamental concepts of stochastic processes
- Model systems using discrete and continuous time Markov chains
- Analyze Poisson processes and their applications
- Apply stochastic processes to problems in finance and sciences

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Probabilidad Avanzada y Procesos Estocásticos (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Fundamentos de probabilidad medida-teórica - Variables aleatorias y distribuciones - Caracterización de procesos estocásticos - Cadenas y procesos de Markov - Movimiento browniano y procesos de difusión	- Aplicar conceptos medida-teóricos a espacios de probabilidad [Familiarizarse (<i>Familiarity</i>)] - Caracterizar diferentes tipos de procesos estocásticos [Evaluar (<i>Assessment</i>)] - Analizar cadenas de Markov para comportamiento transitorio y en estado estacionario [Usar (<i>Usage</i>)] - Modelar sistemas usando procesos estocásticos en tiempo continuo [Usar (<i>Usage</i>)]
Readings : [Ros19], [KT14]	

Unit 2: Simulación de Sistemas Discretos (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
- Teoría de colas y modelos - Autómatas celulares y modelos basados en agentes - Métodos de simulación Monte Carlo - Cadenas y procesos de Markov - Simulación de eventos discretos	- Formular y analizar modelos simples de colas [Usar (<i>Usage</i>)] - Implementar autómatas celulares y simulaciones basadas en agentes [Usar (<i>Usage</i>)] - Aplicar métodos Monte Carlo para simulación probabilística [Usar (<i>Usage</i>)] - Analizar cadenas de Markov para transiciones de estado y comportamiento en estado estacionario [Evaluar (<i>Assessment</i>)] - Diseñar simulaciones de eventos discretos para análisis de sistemas [Usar (<i>Usage</i>)]
Readings : [Ros19], [KT14]	

Unit 3: Inferencia Estadística Computacional (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
- Estimación de Máxima Verosimilitud (MLE) - Métodos de inferencia bayesiana - Métodos de Monte Carlo mediante Cadenas de Markov (MCMC) - Algoritmo Expectación-Maximización (EM) - Remuestreo bootstrap y jackknife	- Implementar Estimación de Máxima Verosimilitud para varias distribuciones [Usar (<i>Usage</i>)] - Aplicar métodos bayesianos para estimación de parámetros [Usar (<i>Usage</i>)] - Implementar algoritmos MCMC para muestreo posterior [Usar (<i>Usage</i>)] - Usar el algoritmo EM para modelos con variables latentes [Usar (<i>Usage</i>)] - Aplicar métodos bootstrap para intervalos de confianza [Usar (<i>Usage</i>)]
Readings : [Ros19], [KT14]	

Unit 4: Simulación de Sistemas Continuos (20 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Modelos de Ecuaciones Diferenciales Ordinarias (EDOs) • Modelos de Ecuaciones Diferenciales Parciales (EDPs) • Métodos de diferencias finitas para EDPs • Selección e implementación de solucionadores • Sistemas stiff y métodos adaptativos	• Formular modelos de EDOs y EDPs para sistemas físicos [Usar (<i>Usage</i>)] • Implementar métodos de diferencias finitas para EDPs simples [Usar (<i>Usage</i>)] • Seleccionar solucionadores numéricos apropiados para diferentes tipos de EDOs/EDPs [Evaluar (<i>Assessment</i>)] • Manejar sistemas stiff usando métodos numéricos apropiados [Usar (<i>Usage</i>)]
Readings : [Ros19], [KT14]	

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

[KT14] Samuel Karlin and Howard M. Taylor. *A First Course in Stochastic Processes*. 2nd. Academic Press, 2014.

[Ros19] Sheldon M. Ross. *Introduction to Probability Models*. 12th. Academic Press, 2019.

Chapter 7

Seventh Semester



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA322-UNMSM-FCM. Computational Number Theory (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA322-UNMSM-FCM. Computational Number Theory
2.2 Semester	: 7 th 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	: MA321-UNMSM-FCM. Abstract algebra. (6 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamentals of computational number theory, integrating advanced mathematical concepts with algorithmic methods to solve problems in cryptography, computer security, and digital communications. It develops skills for implementing modular arithmetic, primality, and factorization algorithms with applications in modern cryptographic systems.

5. GOALS

- Understand fundamental concepts of modular arithmetic and congruences
- Implement algorithms for primality testing and factorization
- Apply number theory methods in public key cryptography
- Analyze security of cryptographic systems based on number theory

6. COMPETENCES

AG-Ci09 Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci10 Formula hipótesis matemáticas. (Usage)

AG-Ci11 Reconoce la estructura formal de problemas matemáticos. (Usage)

7. TOPICS

Unit 1: Teoría de Números Computacional y Criptografía (16 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Aritmética modular y algoritmos • Pruebas de primalidad y factorización • Criptografía de curva elíptica • Problema del logaritmo discreto • Fundamentos de criptografía post-cuántica	• Implementar algoritmos de aritmética modular [Usar (<i>Usage</i>)] • Aplicar métodos de prueba de primalidad [Usar (<i>Usage</i>)] • Usar curvas elípticas en protocolos criptográficos [Usar (<i>Usage</i>)] • Analizar sistemas basados en logaritmo discreto [Evaluar (<i>Assessment</i>)]
Readings : [Sti06], [HPS08]	

Unit 2: Criptografía Simétrica (16 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Cifrados de bloque y modos de operación • Cifrados de flujo • Funciones hash criptográficas • Códigos de autenticación de mensajes • Técnicas de criptoanálisis	• Implementar algoritmos de cifrado de bloque [Usar (<i>Usage</i>)] • Aplicar modos de operación para cifrado [Usar (<i>Usage</i>)] • Usar funciones hash criptográficas [Usar (<i>Usage</i>)] • Analizar ataques de criptoanálisis [Evaluar (<i>Assessment</i>)]
Readings : [Sti06], [HPS08]	

Unit 3: Criptografía Asimétrica (16 hours)	
Competences Expected: AG-Ci09,AG-Ci11	
Topics	Learning Outcomes
• Principios de criptografía de clave pública • Criptosistema RSA • Criptografía de curva elíptica • Intercambio de clave Diffie-Hellman • Firmas digitales	• Implementar cifrado y descifrado RSA [Usar (<i>Usage</i>)] • Aplicar criptografía de curva elíptica [Usar (<i>Usage</i>)] • Usar Diffie-Hellman para intercambio seguro de claves [Usar (<i>Usage</i>)] • Implementar esquemas de firma digital [Usar (<i>Usage</i>)]
Readings : [Sti06], [HPS08]	

Unit 4: Computación con Preservación de la Privacidad (16 hours)	
Competences Expected: AG-Ci09,AG-Ci10	
Topics	Learning Outcomes
• Privacidad diferencial • Cifrado homomórfico • Computación multipartita segura • Privacidad en aprendizaje federado • Métricas y análisis de privacidad	• Aplicar mecanismos de privacidad diferencial [Usar (<i>Usage</i>)] • Usar cifrado homomórfico para computación privada [Usar (<i>Usage</i>)] • Implementar protocolos de computación multipartita segura [Usar (<i>Usage</i>)] • Analizar garantías de privacidad de algoritmos [Evaluar (<i>Assessment</i>)]
Readings : [Sti06], [HPS08]	

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

[Sti06] Douglas R. Stinson. *Cryptography: Theory and Practice*. 3rd. Chapman and Hall/CRC, 2006.

[HPS08] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. 2nd. Springer, 2008.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA341-UNMSM-FCM. Optimization I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA341-UNMSM-FCM. Optimization I
2.2 Semester	: 7 th Semester
2.3 Credits	: 5
2.4 Horas	: 3 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	: MA221-UNMSM-FCM. Numerical Analysis I. (5 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the mathematical foundations of optimization, covering both unconstrained and constrained problems. Classical methods of continuous optimization, optimality conditions, numerical algorithms, and applications in various fields are studied. The course provides the theoretical and computational foundations for formulating and solving optimization problems in science, engineering, and economics, developing analytical skills to characterize and find optimal solutions.

5. GOALS

- Understand the fundamental concepts of optimization and classify different types of optimization problems.
- Apply optimality conditions to characterize solutions in continuous optimization problems.
- Implement numerical algorithms to solve unconstrained and constrained optimization problems.
- Analyze the convergence and efficiency of different optimization methods.
- Formulate and solve real-world optimization problems in various applied contexts.

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

AG-Ci12) Demuestra formal y correctamente declaraciones matemáticas haciendo uso de diversos métodos. (Usage)

7. TOPICS

Unit 1: Optimización Convexa y No Convexa (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci12	
Topics	Learning Outcomes
• Conjuntos y funciones convexas • Condiciones de optimalidad • Métodos de gradiente para optimización convexa • Desafíos de optimización no convexa • Análisis de convergencia	• Caracterizar problemas de optimización convexa [Evaluar (<i>Assessment</i>)] • Aplicar condiciones de optimalidad para verificar soluciones [Usar (<i>Usage</i>)] • Implementar métodos basados en gradiente para problemas convexos [Usar (<i>Usage</i>)] • Analizar desafíos en optimización no convexa [Evaluar (<i>Assessment</i>)]
Readings : [NW06], [BV04]	

Unit 2: Optimización con Restricciones (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci12	
Topics	Learning Outcomes
• Programación lineal (método simplex) • Programación entera • Multiplicadores de Lagrange y condiciones KKT • Métodos de penalización y barrera • Teoría de dualidad	• Resolver problemas de programación lineal [Usar (<i>Usage</i>)] • Formular y resolver problemas de programación entera [Usar (<i>Usage</i>)] • Aplicar condiciones KKT a problemas con restricciones [Usar (<i>Usage</i>)] • Usar métodos de penalización para optimización con restricciones [Usar (<i>Usage</i>)]
Readings : [NW06], [BV04]	

Unit 3: Optimización para Aprendizaje Automático (20 hours)	
Competences Expected: AG-Ci09	
Topics	Learning Outcomes
• Descenso de gradiente estocástico • Momentum y métodos adaptativos (Adam, RM-SProp) • Regularización y dispersidad • Optimización distribuida para AA • Métodos de segundo orden para AA	• Implementar variantes de descenso de gradiente estocástico [Usar (<i>Usage</i>)] • Aplicar métodos de optimización adaptativos [Usar (<i>Usage</i>)] • Usar regularización para generalización de modelos [Usar (<i>Usage</i>)] • Diseñar estrategias de optimización distribuidas [Usar (<i>Usage</i>)]
Readings : [NW06], [Ber16]	

Unit 4: Metaheurísticas y Optimización Bioinspirada (20 hours)	
Competences Expected: AG-Ci09	
Topics	Learning Outcomes
• Algoritmos genéticos y estrategias evolutivas • Inteligencia de enjambre (PSO, colonia de hormigas) • Temple simulado • Búsqueda tabú • Metaheurísticas híbridas	• Implementar algoritmos genéticos [Usar (<i>Usage</i>)] • Aplicar métodos de inteligencia de enjambre [Usar (<i>Usage</i>)] • Usar temple simulado para optimización global [Usar (<i>Usage</i>)] • Diseñar enfoques metaheurísticos híbridos [Usar (<i>Usage</i>)]
Readings : [Tal09], [Mir19]	

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

- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. 1st. Cambridge University Press, 2004.
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2nd. Springer, 2006.
- [Tal09] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. 1st. Wiley, 2009.
- [Ber16] Dimitri P. Bertsekas. *Nonlinear Programming*. 3rd. Athena Scientific, 2016.
- [Mir19] Seyedali Mirjalili. *Evolutionary Algorithms and Neural Networks: Theory and Applications*. 1st. Springer, 2019.



National University of San Marcos (UNMSM)

School of Scientific Computing

Syllabus 2026-I

1. COURSE

MA361-UNMSM-FCM. Mathematical Modeling (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA361-UNMSM-FCM. Mathematical Modeling
2.2 Semester	: 7 th Semester
2.3 Credits	: 5
2.4 Horas	: 3 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	: MA325-UNMSM-FCM. Numerical Analysis II. (6 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the fundamental principles of mathematical modeling to represent, analyze, and predict the behavior of systems in various areas such as science, engineering, and business. It emphasizes the formulation of models from real-world problems, their solution using mathematical and computational tools, and the critical interpretation of results. The course develops abstraction skills and the application of advanced mathematics to solve complex real-world problems.

5. GOALS

- Understand the fundamental principles and stages of the mathematical modeling process.
- Formulate mathematical models from concrete problems in different scientific and technological contexts.
- Apply analytical and numerical methods to solve the proposed models.
- Critically interpret simulation results and validate models with real data.
- Develop skills to effectively communicate models and their conclusions.

6. COMPETENCES

AG-Ci07) Conduce estudios de problemas científicos utilizando métodos de investigación, incluyendo el conocimiento basado en la investigación, el diseño y la conducción de experimentos, el análisis y la interpretación de datos, y la síntesis de información para producir conclusiones válidas. (Usage)

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci14) Implementa procesos matemáticos en la computadora. (Usage)

7. TOPICS

Unit 1: Formulación de Modelos Matemáticos (16 hours)	
Competences Expected: AG-Ci07,AG-Ci09,AG-Ci14	
Topics	Learning Outcomes
• Proceso y ciclo de modelado • Modelos continuos vs. discretos • Modelos deterministas vs. estocásticos • Identificación de supuestos y técnicas de simplificación • Análisis dimensional y escalado	• Describir el ciclo completo de modelado desde la definición del problema hasta la interpretación de la solución [Familiarizarse (<i>Familiarity</i>)] • Clasificar modelos como continuos/discretos y deterministas/estocásticos [Evaluar (<i>Assessment</i>)] • Identificar y justificar supuestos apropiados para simplificación de modelos [Evaluar (<i>Assessment</i>)] • Aplicar análisis dimensional para reducir la complejidad de modelos físicos [Usar (<i>Usage</i>)]
Readings : [GFH13], [LS09]	

Unit 2: Simulación de Sistemas Continuos (16 hours)	
Competences Expected: AG-Ci07,AG-Ci09,AG-Ci14	
Topics	Learning Outcomes
• Modelos de Ecuaciones Diferenciales Ordinarias (EDOs) • Modelos de Ecuaciones Diferenciales Parciales (EDPs) • Métodos de diferencias finitas para EDPs • Selección e implementación de solucionadores • Sistemas stiff y métodos adaptativos	• Formular modelos de EDOs y EDPs para sistemas físicos [Usar (<i>Usage</i>)] • Implementar métodos de diferencias finitas para EDPs simples [Usar (<i>Usage</i>)] • Seleccionar solucionadores numéricos apropiados para diferentes tipos de EDOs/EDPs [Evaluar (<i>Assessment</i>)] • Manejar sistemas stiff usando métodos numéricos apropiados [Usar (<i>Usage</i>)]
Readings : [BCD13], [GFH13]	

Unit 3: Simulación de Sistemas Discretos (16 hours)	
Competences Expected: AG-Ci07,AG-Ci09,AG-Ci14	
Topics	Learning Outcomes
• Teoría de colas y modelos • Autómatas celulares y modelos basados en agentes • Métodos de simulación Monte Carlo • Cadenas y procesos de Markov • Simulación de eventos discretos	• Formular y analizar modelos simples de colas [Usar (<i>Usage</i>)] • Implementar autómatas celulares y simulaciones basadas en agentes [Usar (<i>Usage</i>)] • Aplicar métodos Monte Carlo para simulación probabilística [Usar (<i>Usage</i>)] • Analizar cadenas de Markov para transiciones de estado y comportamiento en estado estacionario [Evaluar (<i>Assessment</i>)] • Diseñar simulaciones de eventos discretos para análisis de sistemas [Usar (<i>Usage</i>)]
Readings : [GFH13], [Mee13]	

Unit 4: Análisis de Sensibilidad y Robustez (16 hours)	
Competences Expected: AG-Ci07,AG-Ci09,AG-Ci14	
Topics	Learning Outcomes
• Análisis de sensibilidad local (derivadas parciales) • Análisis de sensibilidad global (índices de Sobol) • Método de Morris y técnicas de screening • Análisis de robustez y escenarios del peor caso • Propagación de incertidumbre	• Realizar análisis de sensibilidad local usando métodos basados en derivadas [Usar (<i>Usage</i>)] • Aplicar técnicas de análisis de sensibilidad global (ej., índices de Sobol) [Usar (<i>Usage</i>)] • Usar el método de Morris para screening de factores [Usar (<i>Usage</i>)] • Evaluar la robustez del modelo bajo variaciones de parámetros [Evaluar (<i>Assessment</i>)] • Cuantificar la propagación de incertidumbre a través de modelos [Evaluar (<i>Assessment</i>)]
Readings : [Ban+10], [GFH13]	

Unit 5: Validación y Verificación de Modelos (16 hours)	
Competences Expected:	
Topics	Learning Outcomes
• Verificación: corrección del código numérico • Validación: comparación con datos experimentales • Cuantificación de Incertidumbre (UQ) en modelos • Calibración de modelos y estimación de parámetros • Pruebas y métricas de bondad de ajuste	• Distinguir entre verificación y validación en modelado [Familiarizarse (<i>Familiarity</i>)] • Aplicar técnicas de verificación para garantizar la corrección del código [Usar (<i>Usage</i>)] • Validar modelos usando datos experimentales y pruebas estadísticas [Usar (<i>Usage</i>)] • Calibrar parámetros del modelo usando técnicas de optimización [Usar (<i>Usage</i>)] • Aplicar métodos de cuantificación de incertidumbre para evaluar la confiabilidad del modelo [Usar (<i>Usage</i>)]
Readings : [GFH13], [LS09], [Mee13]	

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

[LS09] C. C. Lin and L. A. Segel. *Mathematics Applied to Deterministic Problems in the Natural Sciences*. 1st. Society for Industrial and Applied Mathematics, 2009.

-
- [Ban+10] Jerry Banks et al. *Discrete-Event System Simulation*. 5th. Prentice Hall, 2010.
- [BCD13] Martin Braun, Courtney S. Coleman, and Donald A. Drew. *Differential Equation Models*. 1st. Springer, 2013.
- [GFH13] Frank R. Giordano, William P. Fox, and Steven B. Horton. *A First Course in Mathematical Modeling*. 5th. Cengage Learning, 2013.
- [Mee13] Mark M. Meerschaert. *Mathematical Modeling*. 4th. Academic Press, 2013.

Chapter 8

Eighth Semester



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA331-UNMSM-FCM. Scientific Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA331-UNMSM-FCM. Scientific Computing
2.2 Semester	: 8 th Semester
2.3 Credits	: 5
2.4 Horas	: 3 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	: MA361-UNMSM-FCM. Mathematical Modeling. (7 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course provides the foundations of modern scientific computing, integrating numerical methods, algorithm analysis, and efficient programming to solve complex mathematical problems. It focuses on the practical implementation of numerical algorithms, error analysis, and computational optimization. The course develops skills to select, implement, and validate appropriate numerical methods for different types of scientific and engineering problems.

5. GOALS

- Understand the fundamental principles of computational arithmetic and error analysis in numerical calculations.
- Implement and analyze numerical algorithms for linear algebra, interpolation, and integration.
- Develop computational solutions for ordinary differential equations and initial value problems.
- Apply optimization techniques and root-finding methods to scientific problems.
- Evaluate the stability, convergence, and efficiency of numerical methods in practical applications.

6. COMPETENCES

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

AG-Ci12) Demuestra formal y correctamente declaraciones matemáticas haciendo uso de diversos métodos. (Usage)

AG-Ci14) Implementa procesos matemáticos en la computadora. (Usage)

7. TOPICS

Unit 1: Aritmética Computacional y Análisis de Error (16 hours)	
Competences Expected: AG-Ci11,AG-Ci12,AG-Ci14	
Topics	Learning Outcomes
• Aritmética de punto flotante y el estándar IEEE • Fuentes de error: truncamiento, redondeo y discretización • Propagación de error y número de condición • Estabilidad de algoritmos numéricos • Error absoluto, relativo y directo/inverso	• Definir los diferentes tipos de errores en computación numérica (redondeo, truncamiento, discretización) [Familiarizarse (<i>Familiarity</i>)] • Explicar los principios de representación y aritmética de punto flotante según el estándar IEEE [Familiarizarse (<i>Familiarity</i>)] • Calcular el error absoluto y relativo de una aproximación [Usar (<i>Usage</i>)] • Analizar la propagación de errores en una secuencia de operaciones aritméticas [Usar (<i>Usage</i>)] • Distinguir entre la condición de un problema y la estabilidad de un algoritmo [Evaluar (<i>Assessment</i>)] • Calcular el número de condición de un problema simple (ej., multiplicación matriz-vector) [Usar (<i>Usage</i>)]
Readings : [Hig02], [Gol91]	

Unit 2: Álgebra Lineal Numérica (16 hours)	
Competences Expected: AG-Ci11,AG-Ci12,AG-Ci14	
Topics	Learning Outcomes
• Descomposición LU y sus aplicaciones • Descomposición QR y problemas de mínimos cuadrados • Descomposición en Valores Singulares (SVD) y sus aplicaciones • Número de condición y estabilidad numérica de sistemas lineales • Métodos iterativos para sistemas lineales (Jacobi, Gauss-Seidel, Gradiente Conjugado)	• Implementar algoritmos de descomposición LU y QR para resolver sistemas lineales [Usar (<i>Usage</i>)] • Explicar el significado geométrico y algebraico de la Descomposición en Valores Singulares [Familiarizarse (<i>Familiarity</i>)] • Usar SVD para aplicaciones como aproximación matricial y análisis de componentes principales [Usar (<i>Usage</i>)] • Analizar la estabilidad de un sistema lineal usando su número de condición [Evaluar (<i>Assessment</i>)] • Seleccionar e implementar un método iterativo apropiado para resolver sistemas lineales grandes y dispersos [Evaluar (<i>Assessment</i>)]
Readings : [GL13], [TI97]	

Unit 3: Interpolación y Aproximación de Funciones (16 hours)	
Competences Expected: AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
• Interpolación polinomial (formas de Lagrange, Newton) • Interpolación por splines (splines cúbicos) • Aproximación por mínimos cuadrados • Fenómeno de Runge y nodos de Chebyshev • Análisis de error para interpolación	• Construir polinomios interpoladores usando diferentes métodos [Usar (<i>Usage</i>)] • Implementar interpolación por splines cúbicos [Usar (<i>Usage</i>)] • Aplicar el método de mínimos cuadrados para ajuste de datos [Usar (<i>Usage</i>)] • Explicar el fenómeno de Runge y estrategias para mitigarlo [Familiarizarse (<i>Familiarity</i>)] • Analizar el error de un esquema de interpolación [Evaluar (<i>Assessment</i>)]
Readings : [QSS07], [SB02]	

Unit 4: Diferenciación e Integración Numérica (16 hours)	
Competences Expected: AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
• Aproximaciones por diferencias finitas (adelantadas, atrasadas, centradas) • Extrapolación de Richardson • Fórmulas de Newton-Cotes (regla del trapecio, regla de Simpson) • Cuadratura gaussiana • Análisis de error para integración numérica	• Calcular derivadas de funciones usando esquemas de diferencias finitas [Usar (<i>Usage</i>)] • Aplicar extrapolación de Richardson para mejorar la precisión de aproximaciones numéricas [Usar (<i>Usage</i>)] • Implementar reglas de integración numérica (Trapecio, Simpson, Gaussiana) [Usar (<i>Usage</i>)] • Comparar la precisión y eficiencia de diferentes métodos de integración numérica [Evaluar (<i>Assessment</i>)] • Analizar el error para fórmulas de diferenciación e integración numérica [Evaluar (<i>Assessment</i>)]
Readings : [QSS07], [SB02]	

Unit 5: Métodos Numéricos para Ecuaciones Diferenciales Ordinarias (EDOs) (16 hours)	
Competences Expected: AG-Ci14	
Topics	Learning Outcomes
• Método de Euler y sus variantes (explícito, implícito) • Métodos de Runge-Kutta • Métodos multipaso (Adams-Bashforth, Adams-Moulton) • Análisis de estabilidad y convergencia para solucionadores de EDOs • Sistemas de EDOs y EDOs de orden superior	• Implementar solucionadores básicos de EDOs (Euler, Runge-Kutta) [Usar (<i>Usage</i>)] • Analizar las propiedades de estabilidad y convergencia de métodos numéricos para EDOs [Evaluar (<i>Assessment</i>)] • Seleccionar un método numérico apropiado para una EDO dada basado en requisitos de estabilidad y precisión [Evaluar (<i>Assessment</i>)] • Resolver sistemas de EDOs y convertir EDOs de orden superior a sistemas de primer orden [Usar (<i>Usage</i>)]
Readings : [AP98], [HNW93]	

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

- [Gol91] David Goldberg. "What Every Computer Scientist Should Know About Floating-Point Arithmetic". In: *ACM Computing Surveys* 23.1 (1991), pp. 5–48.
- [HNW93] Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. 2nd. Springer, 1993.
- [TI97] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. 1st. SIAM, 1997.
- [AP98] Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. 1st. SIAM, 1998.
- [Hig02] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. 2nd. SIAM, 2002.
- [SB02] Josef Stoer and Roland Bulirsch. *Introduction to Numerical Analysis*. 3rd. Springer, 2002.
- [QSS07] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. 2nd. Springer, 2007.
- [GL13] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 4th. Johns Hopkins University Press, 2013.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA332-UNMSM-FCM. Fourier and Wavelet Analysis (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA332-UNMSM-FCM. Fourier and Wavelet Analysis
2.2 Semester	: 8 th 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	: MA221-UNMSM-FCM. Numerical Analysis I. (5 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course introduces the theories and applications of Fourier analysis and wavelets for signal processing and analysis. It covers from the fundamentals of Fourier series to modern wavelet transforms, providing the mathematical tools to analyze signals in time and frequency domains. The course emphasizes both theoretical aspects and practical applications in signal processing, data compression, and image analysis.

5. GOALS

- Understand the mathematical foundations of Fourier series and transforms in their different variants.
- Analyze signals in the frequency domain using Fourier techniques and their applications.
- Master the theory of wavelet transforms and multiresolution analysis.
- Implement algorithms for Fast Fourier Transforms (FFT) and wavelet transforms.
- Apply Fourier and wavelet techniques to real problems in signal and image processing.

6. COMPETENCES

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

AG-Ci14) Implementa procesos matemáticos en la computadora. (Usage)

7. TOPICS

Unit 1: Análisis de Fourier y Wavelets (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
- Series y transformadas de Fourier - Transformada de Fourier Discreta (DFT) y FFT - Transformadas wavelet y análisis multirresolución - Diseño de filtros y procesamiento de señales - Aplicaciones en análisis de señales e imágenes	- Implementar algoritmos FFT para procesamiento de señales [Usar (<i>Usage</i>)] - Aplicar transformadas wavelet para análisis multi-escala [Usar (<i>Usage</i>)] - Diseñar filtros digitales para procesamiento de señales [Usar (<i>Usage</i>)] - Analizar señales en el dominio de la frecuencia [Evaluar (<i>Assessment</i>)]
Readings : [Mal09], [Bra00]	

Unit 2: Procesamiento Digital de Imágenes y Visión Computacional (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
- Representación de imágenes y espacios de color - Filtrado y mejora de imágenes - Segmentación de imágenes y detección de características - Procesamiento morfológico de imágenes - Algoritmos de visión computacional	- Implementar algoritmos de filtrado de imágenes [Usar (<i>Usage</i>)] - Aplicar técnicas de segmentación de imágenes [Usar (<i>Usage</i>)] - Usar operaciones morfológicas para análisis de imágenes [Usar (<i>Usage</i>)] - Desarrollar aplicaciones de visión computacional [Usar (<i>Usage</i>)]
Readings : [OS99], [GW18]	

Unit 3: Visualización Científica y Computación Visual (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
- Principios de visualización y percepción - Visualización de flujo y campos vectoriales - Renderizado volumétrico e isosuperficies - Visualización de información - Analítica visual y exploración interactiva	- Diseñar visualizaciones científicas efectivas [Usar (<i>Usage</i>)] - Implementar técnicas de visualización de flujo [Usar (<i>Usage</i>)] - Aplicar renderizado volumétrico a datos científicos [Usar (<i>Usage</i>)] - Crear herramientas de analítica visual interactiva [Usar (<i>Usage</i>)]
Readings : [Bra00], [Bri88]	

Unit 4: Advanced Wavelets and Applications (20 hours)	
Competences Expected: AG-Ci09,AG-Ci11,AG-Ci14	
Topics	Learning Outcomes
• Transformadas wavelet y análisis multirresolución • Orthogonal and biorthogonal wavelets • Multiresolution analysis and filter banks • Daubechies wavelets and their properties • Applications in compression and denoising	• Aplicar transformadas wavelet para análisis multi-escala [Usar (<i>Usage</i>)] • Implement discrete wavelet transforms [Usar (<i>Usage</i>)] • Apply wavelets to image compression [Usar (<i>Usage</i>)] • Analyze signals using multiresolution analysis [Evaluar (<i>Assessment</i>)] • Compare different wavelet families [Evaluar (<i>Assessment</i>)]
Readings : [Mal09], [Dau92]	

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

- [Bri88] E. Oran Brigham. *The Fast Fourier Transform and Its Applications*. 1st. Prentice Hall, 1988.
- [Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*. 1st. SIAM, 1992.
- [OS99] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-Time Signal Processing*. 2nd. Prentice Hall, 1999.
- [Bra00] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. 3rd. McGraw-Hill, 2000.
- [Mal09] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd. Academic Press, 2009.
- [GW18] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. 4th. Pearson, 2018.



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA355-UNMSM-FCM. Statistical Learning Theory (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA355-UNMSM-FCM. Statistical Learning Theory
2.2 Semester	: 8 th 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	: MA351-UNMSM-FCM. Statistics and Probabilities. (4 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course provides the theoretical foundations of statistical learning, covering the mathematical principles that support machine learning algorithms. It focuses on the theoretical analysis of generalization capability, model complexity, and performance guarantees for various learning methods. The course connects theory with practice through the study of generalization bounds, bias-variance tradeoff, and foundations of computational learning theory.

5. GOALS

- Understand the fundamentals of statistical learning theory and its probabilistic foundations.
- Analyze the bias-variance tradeoff in the selection and evaluation of machine learning models.
- Apply Vapnik-Chervonenkis (VC) theory to analyze model complexity and generalization capability.
- Develop and demonstrate generalization bounds for different learning algorithms.
- Connect theoretical principles with the practical implementation of statistical learning algorithms.

6. COMPETENCES

AG-Ci10) Formula hipótesis matemáticas. (Usage)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Usage)

AG-Ci12) Demuestra formal y correctamente declaraciones matemáticas haciendo uso de diversos métodos. (Usage)

7. TOPICS

Unit 1: Fundamentos del Aprendizaje Estadístico (20 hours)	
Competences Expected: AG-Ci10,AG-Ci11,AG-Ci12	
Topics	Learning Outcomes
• Compensación sesgo-varianza • Teoría de Vapnik-Chervonenkis (VC) • Minimización de riesgo empírico • Cotas de generalización • Fundamentos de teoría de aprendizaje	• Analizar la compensación sesgo-varianza en selección de modelos [Evaluar (<i>Assessment</i>)] • Aplicar teoría VC para analizar complejidad de modelos [Usar (<i>Usage</i>)] • Implementar minimización de riesgo empírico [Usar (<i>Usage</i>)] • Interpretar cotas de generalización [Familiarizarse (<i>Familiarity</i>)]
Readings : [Vap98], [HTF09]	

Unit 2: Métodos de Aprendizaje Supervisado (20 hours)	
Competences Expected: AG-Ci10,AG-Ci12	
Topics	Learning Outcomes
• Modelos lineales y extensiones • Máquinas de Vectores Soporte (SVM) • Árboles de decisión y métodos de ensamblado • Métodos de kernel • Aprendizaje supervisado probabilístico	• Implementar y analizar modelos lineales [Usar (<i>Usage</i>)] • Aplicar SVM para clasificación y regresión [Usar (<i>Usage</i>)] • Construir métodos de ensamblado (bagging, boosting) [Usar (<i>Usage</i>)] • Usar métodos de kernel para problemas no lineales [Usar (<i>Usage</i>)]
Readings : [HTF09], [Jam+13]	

Unit 3: Métodos de Aprendizaje No Supervisado (20 hours)	
Competences Expected: AG-Ci10,AG-Ci11	
Topics	Learning Outcomes
• Algoritmos de clustering (k-means, jerárquico) • Reducción de dimensionalidad (PCA, t-SNE) • Estimación de densidad • Aprendizaje de variedades • Autoencoders y aprendizaje profundo no supervisado	• Aplicar algoritmos de clustering para exploración de datos [Usar (<i>Usage</i>)] • Implementar PCA y otros métodos de reducción de dimensionalidad [Usar (<i>Usage</i>)] • Usar estimación de densidad para aprendizaje no supervisado [Usar (<i>Usage</i>)] • Analizar técnicas de aprendizaje de variedades [Evaluar (<i>Assessment</i>)]
Readings : [Mur12], [Bis06]	

Unit 4: Evaluación y Selección de Modelos (20 hours)	
Competences Expected: AG-Ci10	
Topics	Learning Outcomes
• Estrategias de validación cruzada • Métricas de rendimiento para clasificación y regresión • Comparación de modelos y pruebas estadísticas • Optimización de hiperparámetros • Curvas de aprendizaje y herramientas de diagnóstico	• Implementar validación cruzada para evaluación de modelos [Usar (<i>Usage</i>)] • Seleccionar métricas de rendimiento apropiadas [Evaluar (<i>Assessment</i>)] • Comparar modelos usando pruebas estadísticas [Usar (<i>Usage</i>)] • Optimizar hiperparámetros usando métodos sistemáticos [Usar (<i>Usage</i>)]
Readings : [HTF09], [Jam+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

- [Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. 1st. Wiley, 1998.
- [Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 1st. Springer, 2006.
- [HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer, 2009.
- [Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. 1st. MIT Press, 2012.
- [Jam+13] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. 1st. Springer, 2013.

Chapter 9

Ninth Semester



National University of San Marcos (UNMSM)
School of Scientific Computing
Syllabus 2026-I

1. COURSE

MA401-UNMSM-FCM. Capstone Project I (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA401-UNMSM-FCM. Capstone Project I
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	: • MA331-UNMSM-FCM. Scientific Computing. (8th Sem) • MA355-UNMSM-FCM. Statistical Learning Theory. (8th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course represents the first semester of the capstone project in computational mathematics. Students develop an applied research project or technological development that integrates and applies the knowledge acquired during the degree program. The course emphasizes problem formulation, literature review, methodological design, and initial development of computational solutions under the supervision of a faculty advisor. Projects must demonstrate research capability, innovation, and application of computational mathematics to real-world problems.

5. GOALS

- Formulate and delimit a research problem in computational mathematics with theoretical or applied relevance.
- Conduct a systematic literature review and state-of-the-art analysis related to the research problem.
- Design an appropriate methodology to address the problem, selecting suitable computational techniques and tools.
- Develop the initial phases of the project, including prototypes and preliminary analyses.
- Effectively communicate project progress through technical reports and oral presentations.
- Demonstrate autonomous work capability and project management under tutorial supervision.

6. COMPETENCES

- AG-Ci01)** Analiza y evalúa el impacto de las soluciones a problemas científicos en el desarrollo sostenible de la sociedad, la economía, la sostenibilidad, la salud y la seguridad, los marcos legales y el medio ambiente. (Usage)
- AG-Ci02)** Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional, se adhiere a las leyes nacionales e internacionales relevantes y demuestra comprensión de la necesidad de la diversidad y la inclusión. (Usage)
- AG-Ci03)** Se desempeña efectivamente como individuo, y como miembro o líder en equipos diversos e inclusivos y en entornos multidisciplinarios, presenciales, remotos y sus combinaciones. (Usage)
- AG-Ci04)** Se comunica de forma efectiva e inclusiva con la comunidad científica y la sociedad en general, siendo capaz de comprender y redactar informes efectivos, y realizar presentaciones efectivas, teniendo en cuenta diferencias culturales, de idioma y de aprendizaje. (Usage)

AG-Ci05) Aplica los principios de gestión y la toma de decisiones económicas considerando eventuales riesgos, como miembro o líder de equipo, para gestionar proyectos científicos en entornos multidisciplinarios. (Usage)

AG-Ci06) Reconoce la necesidad y está preparado para: i) aprender de forma independiente y continua, ii) adaptarse a tecnologías nuevas y emergentes, y iii) aplicar el pensamiento crítico en el contexto más amplio de los cambios tecnológicos. (Usage)

AG-Ci07) Conduce estudios de problemas científicos utilizando métodos de investigación, incluyendo el conocimiento basado en la investigación, el diseño y la conducción de experimentos, el análisis y la interpretación de datos, y la síntesis de información para producir conclusiones válidas. (Usage)

AG-Ci14) Implementa procesos matemáticos en la computadora. (Usage)

7. TOPICS

Unit 1: Project Formulation and Planning (12 hours)	
Competences Expected: AG-Ci01,AG-Ci05,AG-Ci07	
Topics	Learning Outcomes
• Identification and delimitation of research problems • Systematic literature review and state-of-the-art analysis • Formulation of research objectives and hypotheses • Methodological design and work plan • Development of project proposal and schedule	• Formulate research problems with social impact [Evaluar (<i>Assessment</i>)] • Design management plans for scientific projects [Usar (<i>Usage</i>)] • Conduct problem studies using research methods [Usar (<i>Usage</i>)] • Evaluate technical and scientific feasibility of projects [Evaluar (<i>Assessment</i>)]
Readings : [Bab16], [Kot04]	

Unit 2: Methodological Development and Tools (12 hours)	
Competences Expected: AG-Ci02,AG-Ci03,AG-Ci04	
Topics	Learning Outcomes
• Selection and justification of computational methods • Design of experiments and validation protocols • Computational tools for project development • Version management and technical documentation • Ethical considerations in computational research	• Apply ethical principles in professional research [Usar (<i>Usage</i>)] • Work effectively in multidisciplinary teams [Usar (<i>Usage</i>)] • Communicate technical progress effectively [Usar (<i>Usage</i>)] • Select appropriate tools for the project [Evaluar (<i>Assessment</i>)]
Readings : [Oat06], [Pre14]	

Unit 3: Implementation and Initial Development (12 hours)	
Competences Expected: AG-Ci06,AG-Ci07,AG-Ci14	
Topics	Learning Outcomes
• Development of prototypes and proof of concepts • Implementation of algorithms and computational methods • Preliminary analysis of results and validation • Iteration and refinement of solutions • Technical documentation of development	• Learn independently and continuously [Usar (<i>Usage</i>)] • Conduct studies using research methods [Usar (<i>Usage</i>)] • Implement mathematical processes on computer [Usar (<i>Usage</i>)] • Adapt solutions based on preliminary results [Evaluar (<i>Assessment</i>)]
Readings : [Som15], [McC04]	

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

- [Kot04] C. R. Kothari. *Research Methodology: Methods and Techniques*. 2nd. New Age International, 2004.
- [McC04] Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*. 2nd. Microsoft Press, 2004.
- [Oat06] Brenda J. Oates. *Researching Information Systems and Computing*. 1st. SAGE Publications, 2006.
- [Pre14] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, 2014.
- [Som15] Ian Sommerville. *Software Engineering*. 10th. Pearson, 2015.
- [Bab16] Earl R. Babbie. *The Basics of Social Research*. 7th. Cengage Learning, 2016.

Chapter 10

Tenth Semester



National University of San Marcos (UNMSM)

School of Scientific Computing

Syllabus 2026-I

1. COURSE

MA402-UNMSM-FCM. Capstone Project II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	: MA402-UNMSM-FCM. Capstone Project II
2.2 Semester	: 10 th Semester
2.3 Credits	: 5
2.4 Horas	: 1 HT; 8 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	: MA401-UNMSM-FCM. Capstone Project I. (9 th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course constitutes the final phase of the capstone project in computational mathematics, where students complete, validate, and present an applied research project or technological development. It emphasizes the complete implementation of computational solutions, rigorous analysis of results, methodological validation, and professional communication of findings. The project must demonstrate mastery of computational mathematics, innovation capability, and contribution to knowledge or solution of real-world problems, culminating in professional-quality work and a public defense before a specialized jury.

5. GOALS

- Fully implement the designed computational solution, applying advanced computational mathematics techniques.
- Conduct comprehensive analysis of results and rigorous methodological validations.
- Professionally document the entire research and development process of the project.
- Evaluate the impact and limitations of the developed solution, proposing future improvements.
- Publicly defend the project before a specialized jury, demonstrating topic mastery and technical communication skills.
- Demonstrate autonomous work capability, project management, and professional ethics in applied research development.

6. COMPETENCES

AG-Ci01) Analiza y evalúa el impacto de las soluciones a problemas científicos en el desarrollo sostenible de la sociedad, la economía, la sostenibilidad, la salud y la seguridad, los marcos legales y el medio ambiente. (Assessment)

AG-Ci02) Aplica principios éticos y se compromete con la ética profesional y las normas de la práctica profesional, se adhiere a las leyes nacionales e internacionales relevantes y demuestra comprensión de la necesidad de la diversidad y la inclusión. (Assessment)

AG-Ci03) Se desempeña efectivamente como individuo, y como miembro o líder en equipos diversos e inclusivos y en entornos multidisciplinarios, presenciales, remotos y sus combinaciones. (Assessment)

AG-Ci04) Se comunica de forma efectiva e inclusiva con la comunidad científica y la sociedad en general, siendo capaz de comprender y redactar informes efectivos, y realizar presentaciones efectivas, teniendo en cuenta diferencias culturales, de idioma y de aprendizaje. (Assessment)

AG-Ci05) Aplica los principios de gestión y la toma de decisiones económicas considerando eventuales riesgos, como miembro o líder de equipo, para gestionar proyectos científicos en entornos multidisciplinarios. (Assessment)

AG-Ci06) Reconoce la necesidad y está preparado para: i) aprender de forma independiente y continua, ii) adaptarse a tecnologías nuevas y emergentes, y iii) aplicar el pensamiento crítico en el contexto más amplio de los cambios tecnológicos. (Assessment)

AG-Ci07) Conduce estudios de problemas científicos utilizando métodos de investigación, incluyendo el conocimiento basado en la investigación, el diseño y la conducción de experimentos, el análisis y la interpretación de datos, y la síntesis de información para producir conclusiones válidas. (Assessment)

AG-Ci09) Usa enunciados matemáticos para resolver problemas matemáticos. (Assessment)

AG-Ci11) Reconoce la estructura formal de problemas matemáticos. (Assessment)

AG-Ci12) Demuestra formal y correctamente declaraciones matemáticas haciendo uso de diversos métodos. (Assessment)

AG-Ci14) Implementa procesos matemáticos en la computadora. (Assessment)

7. TOPICS

Unit 1: Advanced Implementation and Optimization (20 hours)	
Competences Expected: AG-Ci07,AG-Ci12,AG-Ci14	
Topics	Learning Outcomes
• Complete implementation of algorithms and computational methods • Optimization and performance improvement of solutions • Technical validation and exhaustive testing • Handling of edge cases and complex situations • Iterative refinement based on results	• Conduct complete studies using research methods [Evaluar (<i>Assessment</i>)] • Formally demonstrate properties of solutions [Evaluar (<i>Assessment</i>)] • Implement complex mathematical processes on computer [Usar (<i>Usage</i>)] • Optimize computational solutions for efficiency [Evaluar (<i>Assessment</i>)]
Readings : [McC04], [Knu97]	

Unit 2: Results Analysis and Validation (20 hours)	
Competences Expected: AG-Ci01,AG-Ci07,AG-Ci11	
Topics	Learning Outcomes
• Rigorous statistical analysis of results • Methodological validation and comparison with benchmarks • Sensitivity and robustness analysis of solutions • Evaluation of model limitations and assumptions • Critical interpretation of findings and conclusions	• Analyze social impact of scientific solutions [Evaluar (<i>Assessment</i>)] • Conduct rigorous methodological validations [Evaluar (<i>Assessment</i>)] • Recognize formal structure of complex problems [Evaluar (<i>Assessment</i>)] • Interpret results in applied context [Usar (<i>Usage</i>)]
Readings : [Mon17], [BHH05]	

Unit 3: Professional Documentation and Communication (20 hours)	
Competences Expected: AG-Ci02,AG-Ci04,AG-Ci06	
Topics	Learning Outcomes
• Preparation of professional technical report • Preparation of scientific articles or technical reports • Design of effective presentations for defense • Communication of results to different audiences • Ethical considerations in results publication	• Apply ethical principles in scientific communication [Usar (Usage)] • Effectively communicate complex results [Evaluar (Assessment)] • Learn continuously through feedback [Usar (Usage)] • Document processes professionally [Evaluar (Assessment)]
Readings : [All96], [MM08]	

Unit 4: Defense and Final Evaluation (20 hours)	
Competences Expected: AG-Ci03,AG-Ci05,AG-Ci09	
Topics	Learning Outcomes
• Preparation for public project defense • Presentation strategies and technical argumentation • Handling jury questions and objections • Critical self-evaluation of own work • Planning of future work and improvements	• Perform effectively in public defense [Evaluar (Assessment)] • Manage complex scientific projects [Evaluar (Assessment)] • Solve advanced mathematical problems [Evaluar (Assessment)] • Respond to criticism constructively [Usar (Usage)]
Readings : [Tur18], [Anh06]	

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

- [All96] Michael Alley. *The Craft of Scientific Writing*. 3rd. Springer, 1996.
- [Knu97] Donald E. Knuth. *The Art of Computer Programming*. 3rd. Addison-Wesley, 1997.
- [McC04] Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*. 2nd. Microsoft Press, 2004.
- [BHH05] George E. P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. 2nd. Wiley, 2005.
- [Anh06] Robert R. H. Anholt. *Dazzle 'Em With Style: The Art of Oral Scientific Presentation*. 2nd. Academic Press, 2006.
- [MM08] Janice R. Matthews and Robert W. Matthews. *Successful Scientific Writing: A Step-by-Step Guide for the Biological and Medical Sciences*. 3rd. Cambridge University Press, 2008.

-
- [Mon17] Douglas C. Montgomery. *Design and Analysis of Experiments*. 9th. Wiley, 2017.
- [Tur18] Kate L. Turabian. *A Manual for Writers of Research Papers, Theses, and Dissertations*. 9th. University of Chicago Press, 2018.