National University of San Marcos Faculty of

Mathematical Sciences School of Scientific Computing



Book of Syllabi

– 2026-I–

Lima: October 28, 2026

Equipo de Trabajo

Dra. Maria Zegarra-Garay Decana de la Facultad de Ciencias Matemáticas email: *mzegarrag@unmsm.edu.pe*	**Mag. Jorge Icaro Condado Jáuregui** Director (e) de Computación Científica email: *jcondadoj@unmsm.edu.pe*
Dr. Frida Mereyda López Córdova Dra. Ing de Sistemas e Informática, UNMSM (2025) Docente del Departamento Académico de Matemáticas email: *flopezc@unmsm.edu.pe*	**Mg. Victoria Ysabel Rojas Rojas** Profesora de la Facultad de Ciencias Matemáticas de la UNMSM email: *vrojasr@unmsm.edu.pe*
Mg. Rubén Arbañil Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *rarbanilr@unmsm.edu.pe*	**Lic. Máximo Gerardo Gonzales Chávez** Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *mgonzalesc1@unmsm.edu.pe*
Mg. Oscar Benito Pacheco Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *obenitop@unmsm.edu.pe*	**Dra. Rosa Medina Aguilar** Profesora de la Facultad de Ciencias Matemáticas de la UNMSM email: *rmedinaa1@unmsm.edu.pe*
Mg. Luz Teresa Quispe Vega Profesora de la Facultad de Ciencias Matemáticas de la UNMSM email: *lquispev@unmsm.edu.pe*	**Mg. Miriam Gisell Pescorán Florencio** Profesora de la Facultad de Ciencias Matemáticas de la UNMSM email: *mpescoranf@unmsm.edu.pe*
Lic. Joseph Wilmer Simpe Laura Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *jsimpel@unmsm.edu.pe*	**Mg. Francisco Felipe Quiroz Chueca** Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *fquirozc@unmsm.edu.pe*
Mg. Rómulo Fernando Lomparte Alvarado Profesor de la Facultad de Ciencias Matemáticas de la UNMSM email: *rlompartea@unmsm.edu.pe*	**Dr. Omar Stevenson Guzmán Rea** Dr. Matemáticas. Universidad Federal de Pernambuco (2020) Docente del Departamento Académico de Matemáticas email: *oguzmanr@unmsm.edu.pe*

Ernesto Cuadros-Vargas
Dr. en Ciencia de la Computación, Universidad de Sao Paulo, Brasil (2004)
Orador distinguido para la *Association of Computing Machinery* (ACM) (2025-2027)
Miembro del Directorio de Gobernadores de la Sociedad de Computación del IEEE (2020-2023)
Miembro del *Steering Committee ACM/IEEE-CS Computing Curricula 2020 (CS2020)*
Miembro del *Steering Committee ACM/IEEE-CS Computing Curricula/Computer Science (CS2013)*
Secretario Ejecutivo del Centro Latinoamericano de Estudios en Informática (CLEI) 2008-2016
email: *ecuadros@spc.org.pe*

Representantes estudiantiles

Maria Fernanda Rupay Caballero
Estudiante de Computación Científica, UNMSM, 2024
email: *maria.rupayc@unmsm.edu.pe*

Sebastián Alfredo Borja Ramos
Estudiante de Computación Científica, UNMSM, 2024
email: *sebastian.borja@unmsm.edu.pe*

Angel Aarón Reyes Cáceres
Estudiante de Computación Científica, UNMSM, 2024
email: *angel.reyesc@unmsm.edu.pe*

Carlos Jesús López Bravo
Estudiante de Computación Científica, UNMSM, 2024
email: *carlos.lopez22@unmsm.edu.pe*

Julio André Carrillo Montero
Estudiante de Computación Científica, UNMSM, 2023
email: *julio.carrillom@unmsm.edu.pe*

Adrian Enrique Borstelmann Segura
Estudiante de Computación Científica, UNMSM, 2025
email: *adrian.borstelmanns@unmsm.edu.pe*

Jan Lucas Mancinelli Vite
Estudiante de Computación Científica, UNMSM, 2023
email: *jan.mancinelli@unmsm.edu.pe*

Adriano Emer Valdivia
Estudiante de Computación Científica, UNMSM, 2025
email: *adriano.valdivia@unmsm.edu.pe*

Cristian Josue Bernal Nuñez
Estudiante de Computación Científica, UNMSM, 2023
email: *cristian.bernal@unmsm.edu.pe*

Jhanilde Pozo Hernandez
Estudiante de Computación Científica, UNMSM, 2023
email: *jhanilde.pozo@unmsm.edu.pe*

Franco David Zevallos Silva
Estudiante de Computación Científica, UNMSM, 2025
email: *franco.zevalloss@unmsm.edu.pe*

Angello Andre Manyari Medina
Estudiante de Computación Científica, UNMSM, 2023
email: *angello.manyari@unmsm.edu.pe*

Contents

First Semester **7**
1.1 CS111. Introduction to Programming . . . 7
1.2 MA100. Basic Mathematics . . . 13
1.3 FG101. Language . . . 16
1.4 FG105. Gender and Society . . . 19
1.5 FG109. Environment and Sustainable Development . . . 22
1.6 FG112. Methods of managing university personnel . . . 25
1.7 FG140. Fundamentals of Scientific Research . . . 28

Second Semester **31**
2.1 CS100. Introduction to Computer Science . . . 31
2.2 CS112. Objects-oriented programming I . . . 48
2.3 AI161. Applied AI . . . 55
2.4 MA111. Calculus I . . . 59
2.5 FI101. General Physics . . . 61

Third Semester **64**
3.1 CS113. Objects-oriented programming II . . . 64
3.2 CS1D1. Discrete Structures I . . . 72
3.3 CS2B1. Platform Based Development . . . 74
3.4 MA112. Calculus II . . . 79
3.5 MA121. Linear Algebra I . . . 81
3.6 ID101. Technical and professional English I . . . 84

Fourth Semester **88**
4.1 CS1D2. Discrete Structures II . . . 88
4.2 CS210. Algorithms and Data Structures . . . 90
4.3 CS271. Data Management . . . 95
4.4 MA122. Linear Algebra II . . . 100
4.5 MA210. Calculus III . . . 103
4.6 ID102. Technical and professional English II . . . 106

Fifth Semester **110**
5.1 CS251. Computer graphics . . . 110
5.2 MA211. Calculus IV . . . 114
5.3 MA221. Numerical Analysis I . . . 117
5.4 MA222. Real Analysis I . . . 120
5.5 MA251. Introduction to Statistics . . . 123

5.6 MA281. Numbers theory . . . 125
5.7 ID201. Technical and professional English III . . . 128

Sixth Semester **132**
6.1 MA212. Ordinary Differential Equations . . . 132
6.2 MA325. Numerical Analysis II . . . 135
6.3 MA351. Statistics and Probabilities . . . 138
6.4 MA371. Functional analysis I . . . 144
6.5 ID202. Technical and professional English IV . . . 147

Seventh Semester **151**
7.1 CS261. Artificial Intelligence . . . 151
7.2 CS3P1. Parallel and Distributed Computing . . . 164
7.3 MA213. Partial Differential Equations . . . 179
7.4 MA326. Numerical Analysis III . . . 182
7.5 MA361. Mathematical Modeling . . . 185
7.6 ID203. Technical and professional English V . . . 188

Eighth Semester **192**
8.1 AI263. Introduction to Machine Learning . . . 192
8.2 MA401. Capstone Project I . . . 194
8.3 CS2S1. Operating systems . . . 197
8.4 CS3I1. Computer Security . . . 203
8.5 QC251. Quantum Mechanics . . . 211
8.6 CS253. Exploratory data analysis . . . 215
8.7 CS370. Big Data and Analytics . . . 219
8.8 STA252. Inferential Statistics . . . 223
8.9 MA327. Numerical Analysis IV . . . 226
8.10 MA352. Stochastic processes . . . 229
8.11 MA372. Functional analysis II . . . 232

Ninth Semester **235**
9.1 AI264. Deep Learning . . . 235
9.2 MA402. Capstone Project II . . . 238
9.3 CS231. Networking and Communication . . . 241
9.4 MA381. Algebraic Structures . . . 246
9.5 AI262. Natural Language Processing . . . 249
9.6 CS273. Time series analysis . . . 251
9.7 MA341. Optimization I . . . 255
9.8 MA355. Statistical Learning Theory . . . 258

Tenth Semester **261**
10.1 CS400. Pre-professional internships . . . 261
10.2 AI268. Computational Vision . . . 263
10.3 AI367. Topics in Artificial Intelligence . . . 273
10.4 MA403. Capstone Project III . . . 285
10.5 QC381. Introduction to Quantum Computing . . . 288
10.6 CS254. Advanced Data Visualization . . . 293



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Introduction to Programming
1.2 Code	:	CS111
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This is the first course in the sequence of introductory courses to Computer Science. This course introduces participants to the fundamental concepts of programming using Python. Topics include data types, control structures, functions, lists, recursion, and the mechanics of execution, testing, and debugging. Programming is one of the pillars of Computer Science; any professional in the area will need to program to materialize their models and proposals.

4. GOALS

- Introduce the fundamental concepts of programming.
- Develop the ability of abstraction using a programming language.
- Develop problem-solving skills through algorithmic thinking.
- Understand basic data structures and their applications.
- Master the Python programming language for problem solving.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Introduction to Programming and History (4 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Brief history of computing and programming languages. • Overview of Python programming language. • Programming environments and tools. • Basic program structure and execution. • Writing, running, and debugging simple programs.	• Identify important trends in the history of computing. [Familiarizarse] • Discuss the historical context of programming language paradigms. [Familiarizarse] • Compare daily life before and after personal computers and the Internet. [Evaluar] • Write, compile, and run a simple Python program. [Usar]
Readings : [**Zelle2010**], [**Guttag2013**], [**Brooks2019**]	

Unit 2: Type Systems Fundamentals (6 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• A type as a set of values together with a set of operations: – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records/structs, unions, arrays, lists, functions, references using set operations) • Type safety and errors caused by using values inconsistently given their intended types • Goals and limitations of static and dynamic typing: detecting and eliminating errors as early as possible. • Primitive data types in Python (int, float, str, bool). • Type conversion and casting.	• Describe, for both a primitive and a compound type, the values that have that type [Familiarizarse] • Describe examples of program errors detected by a type system [Familiarizarse] • Use types and type-error messages to write and debug programs [Usar] • Explain Python's dynamic typing system. [Familiarizarse]
Readings : [**Zelle2010**], [**Guttag2013**]	

Unit 3: Object-Oriented Programming Fundamentals (14 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Imperative programming as a subset of object-oriented programming. • Definition of classes: fields, methods, and constructors. • Dynamic vs static properties. • Variables, expressions, and assignments. • Input and output operations. • Conditional statements (if, elif, else). • Iteration structures (for loops, while loops). • Nested loops and loop control statements. • Exception handling basics.	• Compose a class through design, implementation, and testing to meet behavioral requirements [Usar] • Predict and validate control flow in a program using dynamic dispatch [Evaluar] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Usar] • Implement programs using conditional and iterative structures. [Usar] • Debug programs with syntax and logical errors. [Usar] • Design and implement simple algorithms using loops. [Usar]
Readings : [**Zelle2010**], [**Guttag2013**], [**Brooks2019**]	

Unit 4: Functions and Parameter Passing (10 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Function definition and invocation. • Parameter passing mechanisms (by value, by reference). • Return values and multiple returns. • Variable scope and lifetime (local, global, nonlocal). • Recursion fundamentals. • Lambda expressions and higher-order functions. • Functions as first-class objects. • Documentation and testing of functions.	• Define and implement functions with proper parameter passing. [Usar] • Explain the difference between pass-by-value and pass-by-reference. [Familiarizarse] • Implement recursive solutions to simple problems. [Usar] • Use lambda expressions for functional programming constructs. [Usar] • Design functions with appropriate scope and documentation. [Evaluar]
Readings : [**Zelle2010**], [**Guttag2013**], [**Pragmatic2009**]	

Unit 5: Basic Data Structures (10 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Lists and list operations (indexing, slicing, mutability). • Tuples and immutability. • Dictionaries and key-value mapping. • Sets and set operations. • Strings as sequences. • List comprehensions and generator expressions. • Basic searching and sorting algorithms. • Introduction to algorithmic complexity for basic operations.	• Implement programs using lists, dictionaries, and sets. [Usar] • Choose appropriate data structures for given problems. [Evaluar] • Explain the trade-offs between different data structures. [Familiarizarse] • Implement basic search and sort operations on lists. [Usar]
Readings : [**Zelle2010**], [**Guttag2013**]	

Unit 6: Object-Oriented Programming Fundamentals (10 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Subclasses, inheritance (including multiple inheritance), and method overriding. • Dynamic dispatch: definition of method-call. • Object-oriented idioms for encapsulation: – Privacy, data hiding, and visibility of class members. – Interfaces revealing only method signatures. – Abstract base classes, traits and mixins. • Classes and objects in Python. • Attributes and methods. • Inheritance and polymorphism basics. • Encapsulation and information hiding.	• Build a simple class hierarchy utilizing subclassing that allows code to be reused for distinct subclasses [Usar] • Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Familiarizarse] • Compare and contrast the benefits and costs/impact of using inheritance (subclasses) and composition (specifically, how to base composition on higher order functions) [Evaluar] • Design and implement simple classes in Python. [Usar] • Explain the benefits of OOP for code organization. [Familiarizarse]
Readings : [**Zelle2010**], [**Guttag2013**]	

Unit 7: Algorithmic Strategies (6 hours)

Competences Expected: ABET-1,ABET-2

Topics	**Learning Outcomes**
• Paradigms: – Brute-Force (e.g., linear search, selection sort, traveling salesperson, knapsack) – Decrease-and-Conquer: ∗ By a Constant (e.g., insertion sort, topological sort) ∗ By a Constant Factor (e.g., binary search) ∗ By a Variable Size (e.g., Euclid's) – Divide-and-Conquer (e.g., binary search, quicksort, mergesort, Strassen's) – Greedy (e.g., Dijkstra's, Kruskal's, Knapsack) – Transform-and-Conquer: ∗ Instance simplification (e.g., find duplicates via list presort) ∗ Representation change (e.g., heapsort) ∗ Problem reduction (e.g., least-common-multiple, linear programming) ∗ Dynamic programming (e.g., Floyd's, Marshall, Bellman-Ford) – Space vs time tradeoffs (e.g., hashing) • Iteration vs recursion (e.g., factorial, tree search) • Problem decomposition strategies. • Algorithm design techniques. • Debugging and testing strategies. • Tracing program execution.	• For each of the paradigms in this unit: – Explain its definitional characteristics – Explain an example that demonstrates the paradigm including how this example satisfies the paradigm's characteristics. [Familiarizarse] • Give examples of iterative and recursive algorithms that solve the same problem, explain the benefits and disadvantages of each approach [Usar] • Evaluate whether a greedy approach leads to an optimal solution [Evaluar] • Apply systematic debugging techniques to locate and fix errors. [Usar] • Analyze algorithm complexity for simple problems. [Familiarizarse]

Readings : [**Guttag2013**], [**Cormen2009**]

Unit 8: Modern Development Methods and Tool Usage (4 hours)

Competences Expected: ABET-1,ABET-2

Topics	**Learning Outcomes**
• Using modern programming environments (IDEs, notebooks). • Code search and reuse best practices. • Using standard libraries and APIs. • Debugging and testing techniques. • Version control basics. • Code documentation and style guidelines. • Virtual environments and package management.	• Use modern development tools effectively. [Usar] • Apply code reuse principles through libraries and APIs. [Usar] • Debug and test programs systematically. [Usar] • Follow coding standards and documentation practices. [Familiarizarse] • Manage Python packages and environments. [Familiarizarse]

Readings : [**Zelle2010**], [**Pragmatic2009**]

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Basic Mathematics
1.2 Code	:	MA100
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The course includes knowledge of Notions of Propositional Logic and set theory, Real number systems, Integers, the Principle of Mathematical Induction, and polynomials of one variable. Matrices, determinants, and their applications in problems involving living beings. Introduction to statistics.

4. GOALS

- Develop logical reasoning and critical thinking skills through the use of formal languages.
- Understand and apply the properties of number systems (real and complex) in problem solving.
- Manipulate polynomial and matrix structures to solve systems of equations and interpret results.
- Introduce basic concepts of statistics and probability for data analysis.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

6. TOPICS

Unit 1: Logic and Sets (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Propositional logic: Connectives, truth tables, and validity. • Logical-mathematical quantifiers. • Set theory: Operations, relations, and functions. • Families of sets and Cartesian product.	• Formalize natural language statements into propositional language [Familiarizarse]. • Operate with sets using algebraic laws and Venn diagrams [Usar]. • Prove basic properties of sets and logical relations [Evaluar].
Readings : [**Lipschutz1991**], [**Rojo1995**]	

Unit 2: Number Systems (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• The real number system: Field and order axioms. • Mathematical induction and Newton's binomial theorem. • Complex numbers: Binomial, polar, and exponential forms. • Roots of complex numbers and De Moivre's theorem.	• Identify the algebraic structure of real and complex numbers [Familiarizarse]. • Apply the principle of mathematical induction in proofs of summations [Usar]. • Solve equations in the complex number field using various representations [Evaluar].
Readings : [**Gentile1985**], [**Pinzon1973**]	

Unit 3: Polynomials and Matrices (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomials: Division algorithm and roots of polynomials. • Matrices: Types, operations, and determinants. • Inverse matrix and rank of a matrix. • Systems of linear equations: Gauss-Jordan method.	• Recognize the structure of polynomial rings and matrix algebra [Familiarizarse]. • Use elementary row operations to calculate the inverse of a matrix [Usar]. • Model and solve systems of linear equations applied to contextual problems [Evaluar].
Readings : [**Gentile1984**], [**Rojo1995**]	

Unit 4: Statistics and Probability (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Descriptive statistics: Measures of central tendency. • Dispersion measures and graphical representation. • Probability concepts: Sample space and events. • Conditional probability and independence.	• Describe data sets using basic statistical indicators [Familiarizarse]. • Calculate probabilities of simple and compound events in random experiments [Usar]. • Interpret statistical results for decision making in scientific environments [Evaluar].
Readings : [**Peterson2005**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Language
1.2 Code	:	FG101
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	2
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Language course is fundamental for the development of the university student's communicative skills. It focuses on mastering the rules and principles that regulate the use of standard Spanish, covering phonetic, syntactic, lexical, and pragmatic levels. Likewise, it seeks to strengthen the capacity for reading comprehension, synthesis, and academic writing under international standards, ensuring that the graduate can communicate their ideas with coherence, ethics, and academic rigor.

4. GOALS

- Know and apply Spanish writing standards in the drafting and editing of diverse texts.
- Understand the fundamental concepts of human communication and adapt the message to the academic reality.
- Apply basic research and writing knowledge to prepare structured academic reports.
- Execute formal speeches with responsibility, ethics, and mastery of oral expression techniques.

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Usage)

6. TOPICS

Unit 1: Spanish Language and Regulations (16 hours)	
Competences Expected: ABET-4	
Topics	**Learning Outcomes**
• Language, speech, and norm. • Functions of language and the linguistic reality of Peru. • Written accentuation: General and special rules. • Reading: Comprehension strategies and main ideas. • Orthography of numerical expressions.	• Recognize linguistic terms and language functions in various contexts [Familiarizarse]. • Correctly use orthography and accentuation rules in text production [Usar]. • Analyze the multilingual reality of Peru and characterize academic language [Evaluar].
Readings : [**RAE2010**], [**RAE2014**]	

Unit 2: Academic Writing (16 hours)	
Competences Expected: ABET-4	
Topics	**Learning Outcomes**
• The academic report: Planning and structure. • Direct and indirect citations, and paraphrasing according to APA standards. • The sentence as a syntactic unit: Subject and predicate. • Punctuation marks: Period, comma, and semicolon. • Summary of scientific articles.	• Identify the syntactic structure of simple and compound sentences [Familiarizarse]. • Employ citations and bibliographic references following APA regulations [Usar]. • Prepare coherent summaries of scientific articles and plan academic reports [Evaluar].
Readings : [**MartÃ■nez2002**], [**Nariola2007**]	

Unit 3: Text Drafting and Grammar (16 hours)	
Competences Expected: ABET-4	
Topics	**Learning Outcomes**
• Cases of redundancy, colloquialisms, and solecisms. • Subordinate compound sentences: Noun, adjective, and adverbial. • APA and Vancouver standards in scientific writing. • Textual coherence and cohesion: Dequeísmo and queísmo. • Review and editing of the academic report.	• Detect common writing errors such as lexical imprecision and language vices [Familiarizarse]. • Organize paragraphs with proper grammatical cohesion and coherence [Usar]. • Correct and co-evaluate academic texts to ensure standard language use [Evaluar].
Readings : [**MarÃ■n2008**], [**MartÃ■nez2002**]	

Unit 4: Oral Expression and Public Speaking (16 hours)	
Competences Expected: ABET-4	
Topics	**Learning Outcomes**
• The art of public speaking: Oratory and discourse. • Characteristics and types of oratory. • Preparation of visual aids (PPT) for presentations. • Ethical responsibilities of the speaker. • Final presentation of the academic report.	• Internalize fundamental discourse techniques and active listening [Familiarizarse]. • Design effective visual presentations that complement oral communication [Usar]. • Present academic reports with fluency, propriety, and ethics before an audience [Evaluar].
Readings : [**Verderber2010**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Gender and Society
1.2 Code	:	FG105
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	2
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Gender and Society course seeks to familiarize students with the category of gender as an analytical tool to understand social organization. The course critically addresses the implications of inequality and discrimination in the production of knowledge and in the socio-historical structure. Through this analysis, a reflective vision of the environment is promoted, fostering respect for diversity and the pursuit of equity in the exercise of citizenship and professional practice.

4. GOALS

- Understand the concept of gender and its importance for the analysis of social structures and critical thinking.
- Identify historical processes of domination and the formation of gender stereotypes in Western and Peruvian society.
- Analyze the relationship between gender, division of labor, and access to public and private power spaces.
- Evaluate equality policies and the national and international regulatory framework aimed at reducing gender gaps.

5. COMPETENCES

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Familiarity)

6. TOPICS

Unit 1: Fundamental Concepts of Gender (8 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Sex and gender: Conceptual and biological distinctions. • Gender as an analytical category in the social sciences. • Gender socialization and the construction of identities. • Stereotypes, prejudices, and gender roles today.	• Distinguish the biological bases of sex from the social and cultural construction of gender [Familiarizarse]. • Explain how gender stereotypes influence the perception of the environment and communication [Usar]. • Critically analyze the production of knowledge from an inclusive gender perspective [Evaluar].
Readings : [**Butler2007**], [**Lamas1996**]	

Unit 2: Gender, History, and Social Structure (8 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Historical evolution of gender relations. • The sexual division of labor: Public and private spaces. • Gender and coloniality in Latin America. • Patriarchy and systems of domination in Peruvian history.	• Identify historical milestones that have shaped gender gaps in Latin American society [Familiarizarse]. • Describe the impact of the division of labor on the economic and social development of women [Usar]. • Evaluate how historical power structures affect current sustainability and social equity [Evaluar].
Readings : [**Federici2010**], [**Quijano1988**]	

Unit 3: Gender, Power, and Citizenship (8 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Gender and the exercise of citizenship in the 21st century. • Political participation and access to decision-making positions. • Gender-based violence: Definitions, types, and statistics. • Masculinities and changes in power relations.	• Recognize the various forms of gender-based violence and existing prevention mechanisms [Familiarizarse]. • Apply ethical principles in the analysis of power relations between genders [Usar]. • Argue for the importance of diversity and inclusion in the development of civil society [Evaluar].
Readings : [**Fuller2004**], [**Pateman2016**]	

Unit 4: Public Policies and Regulatory Framework (8 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• International legal framework: CEDAW and the SDGs. • National Gender Equality Policy in Peru. • Gender perspective in public management and social projects. • Current challenges for substantive equality.	• Identify the main international treaties ratified by Peru regarding gender [Familiarizarse]. • Integrate the gender perspective in the planning and management of multidisciplinary projects [Usar]. • Evaluate the impact of public equality policies on national health, safety, and economy [Evaluar].
Readings : [**Durand2014**], [**MIMP2019**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Environment and Sustainable Development
1.2 Code	:	FG109
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Environment and Sustainable Development course is theoretical and practical in nature, aiming to raise student awareness regarding the importance of ecosystem conservation. The course covers the principles of ecology, the impact of human activities on the biosphere, and global strategies to achieve development that balances economic growth, social equity, and environmental protection. This knowledge is crucial for computer science professionals to design technological solutions with a focus on environmental responsibility and sustainability.

4. GOALS

- Understand the fundamental concepts of ecology and the functioning of biogeochemical cycles.
- Analyze the causes and consequences of major global environmental problems, such as climate change.
- Evaluate the national and international regulatory framework oriented towards environmental management and sustainability.
- Apply sustainability indicators in the evaluation of scientific and technological projects.

5. COMPETENCES

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Familiarity)

6. TOPICS

Unit 1: Ecology and Biodiversity (16 hours)	
Competences Expected: ABET-5	
Topics	**Learning Outcomes**
• Ecology: Basic concepts, ecosystems, and biomes. • Energy flow and biogeochemical cycles in nature. • Biodiversity: Importance, levels, and current threats. • Ecosystem services and valuation of natural capital.	• Identify the components of an ecosystem and the interdependence relationships among living beings [Familiarizarse]. • Explain the impact of biodiversity loss on the stability of natural systems [Usar]. • Evaluate the health of a local ecosystem using biological indicators and physicochemical parameters [Evaluar].
Readings : [**Miller2012**], [**Smith2001**]	

Unit 2: Environmental Impact and Climate Change (16 hours)	
Competences Expected: ABET-5	
Topics	**Learning Outcomes**
• Environmental pollution: Air, water, soil, and waste. • The greenhouse effect and global warming. • Impact of industrial and technological activities. • Ecological footprint and carbon footprint: Measurement and reduction.	• Recognize the main sources of pollution and their effects on public health and the climate [Familiarizarse]. • Calculate personal and institutional carbon footprints to propose mitigation measures [Usar]. • Critically analyze the environmental impact of data centers and technological infrastructure [Evaluar].
Readings : [**Miller2012**], [**Star2004**]	

Unit 3: Environmental Management and Regulations (16 hours)	
Competences Expected: ABET-5	
Topics	**Learning Outcomes**
• Environmental legal framework in Peru and the Ministry of Environment. • Environmental Impact Assessment (EIA) and certifications. • Natural resource management: Water, energy, and protected areas. • Environmental ethics and corporate social responsibility.	• Describe the functions of environmental regulatory bodies at the national level [Familiarizarse]. • Apply ethical and legal principles in the design of environmental management plans for projects [Usar]. • Argue for the importance of transparency and citizen participation in environmental issues [Evaluar].
Readings : [**Strange2012**], [**Primack1993**]	

Unit 4: Strategies for Sustainable Development (16 hours)	
Competences Expected: ABET-5	
Topics	**Learning Outcomes**
	• Recognize the 17 Sustainable Development Goals and their relationship with scientific computing [Familiarizarse]. • Integrate sustainability criteria and circular economy into the software life cycle [Usar]. • Propose innovative technological solutions to solve contemporary socio-environmental challenges [Evaluar].
Readings : [**Strange2012**], [**Miller2012**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Methods of managing university personnel
1.2 Code	:	FG112
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	2
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The course University Personal Management Methods is theoretical-practical in nature and aims to empower students through the development of intrapersonal competencies and intellectual work strategies. The course integrates self-knowledge and strategic planning processes with advanced learning techniques and digital tools. Through this comprehensive approach, students will be able to manage their behavior in the face of change, optimize their academic performance, and conduct research with ethical rigor, fostering a strategic vision to face the challenges of the contemporary university and professional environment.

4. GOALS

- Develop a personal strategic plan integrating autonomous learning techniques and SWOT analysis.
- Apply cognitive and metacognitive strategies to optimize the processing, retention, and communication of scientific information.
- Enhance leadership, ethics, and collaborative work skills in face-to-face and virtual environments.
- Produce monographic research papers and effective presentations following academic integrity standards and use of ICTs.

5. COMPETENCES

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: Strategic Management and Learning (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Basic concepts of management and university learning. • Self-knowledge and personal diagnosis (SWOT). • Personal vision, mission, and values. • Cognitive processes, attention, and learning strategies. • Metacognition: Learning to learn and self-regulation.	• Identify the principles of management and cognitive factors involved in academic success [Familiarizarse]. • Develop a strategic planning matrix oriented towards individual development [Usar]. • Evaluate the alignment between personal values and continuous learning processes [Evaluar].
Readings : [**Alcaide2021**], [**Bernardo2020**], [**Fischman2018**]	

Unit 2: Intellectual Work Techniques and ICTs (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Comprehensive reading, underlining, and note-taking. • Visual organizers: Concept and mind maps. • Time management and personal productivity. • Personal Learning Environment (PLE) and Web 2.0 tools. • Collaborative work in virtual environments.	• Recognize the usefulness of visual organizers and ICT tools in studying [Familiarizarse]. • Use synthesis techniques and time management to optimize performance [Usar]. • Design an efficient personal learning environment integrating digital and productivity tools [Evaluar].
Readings : [**Pallares2017**], [**Novak2019**], [**Lazzati2019**]	

Unit 3: Leadership, Ethics, and Research (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Leadership: Styles, theories, and conflict management. • The monograph: Structure, search, and source selection. • Ethical use of information and plagiarism prevention (Citation standards). • Social responsibility and professional ethics in project management.	• Understand the methodological structure of research and the principles of effective leadership [Familiarizarse]. • Select reliable information sources for solving management problems [Usar]. • Write academic documents respecting intellectual property and ethical principles [Evaluar].
Readings : [**Cortina2021**], [**Montenegro2022**], [**Hernandez2018**]	

Unit 4: Communication and Decision Making (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• The decision-making process: Models and risks. • Inclusive academic communication: Synchronous and asynchronous. • Oral presentations and preparation of visual aids. • Emotional intelligence and proactive adaptation to change.	• Identify the critical stages for decision-making and effective communication [Familiarizarse]. • Apply assertive communication strategies to lead and present project results [Usar]. • Resolve conflict situations and make decisions based on critical thinking and emotional intelligence [Evaluar].
Readings : [**Bernardo2020**], [**Lazzati2019**], [**Zimbardo2017**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Fundamentals of Scientific Research
1.2 Code	:	FG140
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Fundamentals of Scientific Research course provides the student with the basic knowledge and tools for the development of high-quality scientific research. The course addresses the problematization of reality, theoretical foundation, methodology, and ethics in research. It seeks to develop in the student the capacity for critical observation, academic rigor, and intellectual honesty necessary to produce new knowledge that contributes to the advancement of science and technology.

4. GOALS

- Understand the concept of science, the scientific method, and the stages of the research process.
- Identify and pose relevant research problems based on the observation of reality.
- Apply techniques for searching and managing scientific information in specialized databases.
- Know the standards for scientific writing and the ethical principles governing the publication of results.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Familiarity)

6. TOPICS

Unit 1: Science and Scientific Method (12 hours)	
Competences Expected: ABET-3	
Topics	**Learning Outcomes**
• Scientific knowledge: Characteristics and levels. • Science: Classification and objectives. • The scientific method: Stages and application. • Ethics in research and best practices.	• Differentiate scientific knowledge from empirical or everyday knowledge [Familiarizarse]. • Sequentially apply the stages of the scientific method in practical cases [Usar]. • Evaluate the importance of ethics and scientific integrity in project development [Evaluar].
Readings : [**Bunge2014**], [**Hernandez2014**]	

Unit 2: Scientific Information Management (12 hours)	
Competences Expected: ABET-3	
Topics	**Learning Outcomes**
• Information search: Strategies and databases. • Bibliographic management tools (Mendeley, Zotero). • Critical reading of scientific articles. • Citation and reference standards (APA, Vancouver, IEEE).	• Identify reliable information sources and high-impact academic databases [Familiarizarse]. • Use bibliographic managers to organize and cite sources in an academic document [Usar]. • Write a coherent and well-founded literature review on a specific topic [Evaluar].
Readings : [**Martinez2013**], [**SÃąnchez2011**]	

Unit 3: Research Project Planning (12 hours)	
Competences Expected: ABET-3	
Topics	**Learning Outcomes**
• Identification and delimitation of the research problem. • Formulation of the research question and objectives. • Justification and feasibility of the research. • Hypothesis formulation and definition of variables.	• Recognize the fundamental components of a scientific research problem [Familiarizarse]. • Formulate mathematical or logical hypotheses that can be experimentally tested [Usar]. • Design the preliminary methodological framework for a rigorous scientific study [Evaluar].
Readings : [**Hernandez2014**], [**Kothari2009**]	

Unit 4: Scientific Writing and Communication (12 hours)	
Competences Expected: ABET-3	
Topics	**Learning Outcomes**
• Structure of the scientific article (IMRaD). • Technical writing: Clarity, precision, and brevity. • Tables, figures, and data representation. • The publication process and peer review.	• Understand the standard structure of scientific reports and communications [Familiarizarse]. • Prepare effective reports using technical language and appropriate visual representations [Usar]. • Present and orally defend research results in an inclusive manner [Evaluar].
Readings : [**Olson2014**], [**Russey2006**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Introduction to Computer Science
1.2 Code	:	CS100
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Familiarity)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Familiarity)

6. TOPICS

Unit 1: Computational Thinking (Part I) (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Explanation of the evaluation system. • General course instructions. • Definition of Computing. • Computing as a Human-Computer binomial. • Distortions in the definition of computing. • Computing as the automation of abstraction. • Computing and Engineering: similarities and differences. • Algorithmic problem-solving. • Dynamics: Understanding the execution of an algorithm at human speed.	• Apply the fundamental concepts of computing in real-life situations. [Usar] • Identify distortions of Computing in real-life situations. [Usar] • Clearly identify at least 3 contexts of using the word"Engineer" in English. [Evaluar] • Identify the limitations of humans in solving computational problems. [Usar]
Readings : [**Brookshear2019**]	

Unit 2: Computational Thinking. Part II (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Decomposition (Breaking a problem in smaller pieces) • Abstraction (Focus on important things) • Pattern recongnition (Identify similar sequences based on previous problems) • Designing algorithms	• Apply Computational Thinking to real-world problems. [Usar]
Readings : [**Brookshear2019**]	

Unit 3: General concepts (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Binary vs. decimal numbering. • Character representation: the ASCII table. • Binary search. • Computational complexity of an algorithm.	• Apply various numbering systems to real-world problems. [Usar] • Understand the internal representation of characters in the ASCII and UTF-8 tables. [Familiarizarse] • Apply Divide and Conquer algorithmic strategy. [Usar] • Determine basic analysis of algorithmic complexity. [Usar]
Readings : [**Brookshear2019**]	

Unit 4: Machine-Level Data Representation (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Overview and history of computer architecture • Bits, bytes, and words • Unsigned, signed and two's complement representations • Numeric data representation and number bases: – Fixed-point – Floating-point • Representation of non-numeric data • Representation of records, arrays and UTF data types	• Discuss why everything in computers are data, including instructions [Debate] • Explain how fixed-length number representations can affect accuracy and precision [Explain] • Describe how negative integers are stored in sign-magnitude and two's-complement representations [Describe] • Discuss how different formats can represent numerical data [Debate] • Explain the bit-level representation of non-numeric data, such as characters, strings, records, and arrays [Explain] • Translate numerical data from one format to another [Translate] • Describe how a single adder (without overflow detection) can handle both signed (two's complement) and unsigned (binary) input without "knowing" which format a given input is using [Describe]
Readings : [**Brookshear2019**]	

Unit 5: Program Abstraction and Representation (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Programs that take (other) programs as input such as interpreters, compilers, type-checkers, documentation generators • Components of a language: – Definitions of alphabets, delimiters, sentences, syntax, and semantics – Syntax vs semantics • Program as a set of non-ambiguous meaningful sentences • Basic programming abstractions: constants, variables, declarations (including nested declarations), command, expression, assignment, selection, definite and indefinite iteration, iterators, function, procedure, modules, exception handling • Types of variables: static, local, nonlocal, global; need and issues with nonlocal and global variables. • Scope rules: static vs dynamic; visibility of variables; side-effects. • Environment vs store and their properties • Data and control abstraction • Mechanisms for information exchange between program units such as procedures, functions, and modules: nonlocal variables, global variables, parameter-passing, import-export between modules • Low level instruction representation such as virtual machine instructions, assembly language, and binary representation	• Illustrate the scope of variables and visibility using simple programs [Apply] • Illustrate different types of parameter-passing using simple pseudo programming language [Apply] • Explain side-effect using global and nonlocal variables and how to fix such programs [Explain] • Explain how programs that process other programs treat the other programs as their input data [Explain] • Describe a grammar and an abstract syntax tree for a small language [Describe]
Readings : [**Brookshear2019**]	

Unit 6: Cryptography Fundamentals (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Basic concepts on Cryptography • Public-keys, Private-keys	• Explain the role of cryptography in supporting security and privacy [Explain] • Discuss the risks of inventing one's own cryptographic methods [Debate] • Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Debate] • Implement and cryptanalyze classical ciphers [Implement]
Readings : [**Brookshear2019**]	

Unit 7: Memory Management (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Review of physical memory, address translation and memory management hardware See also: Memory Hierarchy, Discrete Mathematics • Impact of memory hierarchy including cache concept, cache lookup, and per-CPU caching on operating system mechanisms and policy See also: Memory Hierarchy, Latency, Cache and Memory Hierarchy, Virtualization and Isolation • Logical and physical addressing, address space virtualization. See also: Memory Hierarchy, Discrete Mathematics • Concepts of paging, page replacement, thrashing and allocation of pages and frames • Allocation/deallocation/storage techniques (algorithms and data structure) performance and flexibility – Arenas, slab allocators, free lists, size classes, heterogeneously sized pages (huge pages) • Memory caching and cache coherence and the effect of flushing the cache to avoid speculative execution vulnerabilities. See also: Functional Organization, Memory Hierarchy, Latency, Cache and Memory Hierarchy, Virtualization and Isolation • Virtual memory: leveraging virtual memory hardware for OS services and efficiency	• Explain memory hierarchy and cost-performance tradeoffs [Explain] • Summarize the principles of virtual memory as applied to caching and paging [Summarize] • Evaluate the tradeoffs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Evaluate] • Describe the reason for and use of cache memory (performance and proximity, how caches complicate isolation and virtual machine abstraction) [Describe]
Readings : [**Brookshear2019**]	

Unit 8: Role and Purpose of Operating Systems (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Operating systems mediate between general purpose hardware and application-specific software. • Universal operating system functions (e.g., process, user and device interfaces, persistence of data) • Influences of security, networking, multimedia, parallel and distributed computing	• Understand the objectives and functions of modern operating systems [Explain] • Evaluate the design issues in different usage scenarios (e.g., real time OS, mobile, server) [Evaluate] • Understand how evolution and stability are desirable and mutually antagonistic in operating systems function [Explain]
Readings : [**Brookshear2019**]	

Unit 9: Principles of Operating System (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Operating system software design and approaches (e.g., monolithic, layered, modular, micro-kernel, unikernel) • Abstractions, processes, and resources • The evolution of the link between hardware architecture and the operating system functions • Leveraging interrupts from hardware level: service routines and implementations. See also: Assembly Level Machine Organization – Timer interrupts for implementing time slices – I/O interrupts for putting blocking threads to sleep without polling	• Understand how the application of software design approaches to operating systems design/implementation (e.g., layered, modular, etc.) affects the robustness and maintainability of an operating system [Explain] • Categorize system calls by purpose [Categorize] • Apply OS techniques for isolation, protection, and throughput across OS functions (e.g., starvation similarities in process scheduling, disk request scheduling, semaphores, etc.) and beyond [Apply]
Readings : [**Brookshear2019**]	

Unit 10: Fundamentals of Networks and Communications (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Importance of networking in contemporary computing, and associated challenges. See also: Digital Equity and Accessibility, Social Impact and AI, Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation • Organization of the internet (e.g., users, Internet Service Providers, autonomous systems, content providers, content delivery networks) • Layers and their roles (application, transport, network, datalink, and physical) • Network elements (e.g., routers, switches, hubs, access points, and hosts) • Basic queueing concepts (e.g., relationship with latency, congestion, service levels, etc.)	• Articulate the organization of the internet [Articulate] • List and define the appropriate network terminology [List] • Describe the layered structure of a typical networked architecture [Describe]
Readings : [**Brookshear2019**]	

Unit 11: Networked Applications (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Naming and address schemes (e.g., DNS, and Uniform Resource Identifiers) • Distributed application paradigms (e.g., client/server, peer-to-peer, cloud, edge, and fog). See also: Communication, Coordination • Diversity of networked application demands (e.g., latency, bandwidth, and loss tolerance). See also: Communication, Sustainable Design and Pervasive Computing, Environmental Footprint of Computing Systems, Systemic Effects and Social Context, Digital Equity and Accessibility, Social Impact and AI • Coverage of application-layer protocols (e.g., HTTP) • Interactions with TCP, UDP, and Socket APIs. See also: Programs: Distribution, Programs: Execution Properties, GPU Programming, Programs: Implementation Mappings, Programs: Declarative Parallelism, Programs: Starting Activities	• Define the principles of naming, addressing, resource location [Define] • Analyze the needs of specific networked application demands [Analyze]
Readings : [**Brookshear2019**]	

Unit 12: Complexity Analysis Framework (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Complexity Analysis Framework: – Best, average, and worst-case performance of an algorithm – Empirical and relative (Order of Growth) measurements – Input size and primitive operations – Time and space efficiency • Empirical measurements of performance • Time and space tradeoffs in algorithms	• Prepare a presentation that explains to first year students the basic concepts of algorithmic complexity including best, average, and worst-case algorithm behavior, Big-O, Omega, and Theta notations, complexity classes, time and space tradeoffs, empirical measurement, and impact on practical problems [Explain] • For each algorithm in the Foundational Data Structures, Foundational Algorithms, Advanced Algorithms unit, explain its runtime complexity class and why it belongs to this class [Explain] • Informally evaluate the foundational complexity class of simple algorithms [Evaluate] • Develop empirical studies to determine and validate hypotheses about the runtime complexity of various algorithms by running algorithms on input of various sizes and comparing actual performance to the theoretical analysis [Create] • Explain examples that illustrate time-space tradeoffs of algorithms [Explain] • Explain how tree balance affects the efficiency of binary search tree operations [Explain]
Readings : [**Brookshear2019**]	

Unit 13: Asymptotic Notation and Complexity Classes (1 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Asymptotic complexity analysis (average and worst-case bounds): – Big-O, Big-Omega, and Big-Theta formal notations – Foundational Complexity Classes and Representative Examples/Problems: ∗ $O(1)$ Constant (e.g., array access) ∗ $O(\log_2 n)$ Logarithmic (e.g., binary search) ∗ $O(n)$ Linear (e.g., linear search) ∗ $O(n \log_2 n)$ Log Linear (e.g., mergesort) ∗ $O(n^2)$ Quadratic (e.g., selection sort) ∗ $O(n^c)$ Polynomial (e.g., $O(n^3)$ Gaussian elimination) ∗ $O(2^n)$ Exponential (e.g., Knapsack, Satisfiability (SAT), Traveling Sales-Person (TSP), all subsets) ∗ $O(n!)$ Factorial (e.g., Hamiltonian circuit, all permutations)	• Using examples, explain each of the foundational complexity classes in this unit [Explain] • For each foundational complexity class in this unit, explain an algorithm that demonstrates the associated runtime complexity [Explain] • Explain to a non-technical audience the significance of tractable versus intractable algorithms using an intuitive explanation of Big-O complexity [Explain]
Readings : [**Brookshear2019**]	

Unit 14: Complexity Analysis: Recursion, Amortization, and Tight Bounds (1 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Little-o, Little-Omega, and Little Theta notations	• Given a problem to program for which there may be several algorithmic approaches, evaluate them and determine which are feasible, and select one that is optimal in implementation and run-time behavior [Evaluate]
Readings : [**Brookshear2019**]	

Unit 15: Foundational Data Structures (3 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Abstract Data Type (ADT) and operations on an ADT: – Dictionary operations (insert, delete, find) • Arrays: – Numeric vs non-numeric, character strings – Single (vector) vs multidimensional (matrix) • Records/Structs/Tuples and Objects • Linked lists (for historical reasons): – Single vs Double and Linear vs Circular • Stacks • Queues and deques: – Heap-based priority queue • Hash tables/maps: – Collision resolution and complexity (e.g., probing, chaining, rehash) • Trees: – Binary, n-ary, and search trees – Balanced (e.g., AVL, Red-Black, Heap)	• For each ADT/Data-Structure in this unit: – Explain its definition, properties, representation(s), and associated ADT operations. – Explain step-by-step how the ADT operations associated with the data structure transform it. [Explain] • Explain how collision avoidance and collision resolution is handled in hash tables [Explain] • Explain the heap property and the use of heaps as an implementation of a priority queue [Explain]
Readings : [**Brookshear2019**]	

Unit 16: Foundational Algorithms (3 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Graphs (e.g., [un]directed, [a]cyclic, [un]connected, and [un]weighted): – Graph representation: adjacency list vs matrix • Search algorithms: – $O(n)$ complexity (e.g., linear/sequential array/list search) – $O(\log_2 n)$ complexity (e.g., binary search) – $O(\log_b n)$ complexity (e.g., uninformed depth/breadth-first tree search) • Sorting algorithms (e.g., stable, unstable): – $O(n^2)$ complexity (e.g., insertion, selection) – $O(n \log n)$ complexity (e.g., quicksort, merge, timsort) • Graph algorithms: – Shortest path (e.g., Dijkstra's, Floyd's) – Minimal spanning tree (e.g., Prim's, Kruskal's) • Sorting algorithms: – $O(n \log n)$ complexity heapsort – Pseudo $O(n)$ complexity (e.g., bucket, counting, radix) • Graph algorithms: – Transitive closure (e.g., Warshall's) – Topological sort	• For each algorithm in this unit explain step-by-step how the algorithm operates [Explain] • For each algorithmic approach (e.g., sorting) in this unit apply a prototypical example of the approach (e.g., merge sort) [Apply] • Given requirements for a problem, develop multiple solutions using various data structures and algorithms. Subsequently, evaluate the suitability, strengths, and weaknesses selecting an approach that best satisfies the requirements [Create] • Explain factors beyond computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Explain]
Readings : [**Brookshear2019**]	

Unit 17: Advanced Algorithms (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Cryptography algorithms (e.g., SHA-256) • Parallel algorithms	• An appreciation of quantum computation and its application to certain problems [Explain]
Readings : [**Brookshear2019**]	

Unit 18: Core Database System Concepts (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Purpose and advantages of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Transaction management • Normalization • Approaches for managing large volumes of data (e.g., NoSQL database systems, use of MapReduce) • Distributed databases/cloud-based systems • Use of a declarative query language	• Identify at least four advantages that using a database system provides [Analyze] • Enumerate the components of a (relational) database system [Enumerate] • Defend the value of data independence [Defend] • Compose a simple select-project-join query in SQL [Compose] • Describe the advantages for eliminating duplicate repeated data [Describe] • Outline how MapReduce uses parallelism to process data efficiently [Outline]
Readings : [**Brookshear2019**]	

Unit 19: Object-Oriented Programming Fundamentals (4 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Imperative programming as a subset of object-oriented programming. • Object-oriented design: – Decomposition into objects carrying state and having behavior. – Class-hierarchy design for modeling. • Definition of classes: fields, methods, and constructors. • Subclasses, inheritance (including multiple inheritance), and method overriding. • Dynamic dispatch: definition of method-call. • Object-oriented idioms for encapsulation: – Privacy, data hiding, and visibility of class members. – Interfaces revealing only method signatures. – Abstract base classes, traits and mixins.	• Enumerate the differences between imperative and object-oriented programming paradigms [Enumerate] • Compose a class through design, implementation, and testing to meet behavioral requirements [Compose] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Use]
Readings : [**Brookshear2019**]	

Unit 20: Version Control and CI/CD (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Software configuration management and version control: Software Development Practices – Configuration in version control, reproducible builds/configuration. – Version control branching strategies. Development branches vs release branches. Trunk-based development. – Merging/rebasing strategies, when relevant. • Release management. • Software process automation: – Build systems - the value of fast, hermetic, reproducible builds, compare/contrast approaches to building a project. – Continuous Integration (CI) - the use of automation and automated tests to do preliminary validation that the current head/trunk revision builds and passes (basic) tests. – Continuous Deployment (CD) - the use of automation to automatically release every change that passes the automated tests to the production environment, ensuring frequent and reliable deliveries. – Dependency management - updating external/upstream dependencies, package management, SemVer. • Tool integration concepts and mechanisms. Software Development Practices • Use of modern IDE facilities - debugging, refactoring, searching/indexing, ML-powered code assistants, etc. Software Development Practices	• Describe the difference between centralized and distributed software configuration management [Describe] • Describe how version control can be used to help manage software release management [Describe] • Identify configuration items and use a source code control tool in a small team-based project [Analyze] • Understand the use of CI/CD systems as a ground-truth for the state of the team's shared code (build and test success) [Explain] • Demonstrate the capability to use software tools in support of the development of a software product of medium size [Demonstrate]
Readings : [**Brookshear2019**]	

Unit 21: Requirements Engineering (2 hours)

Competences Expected: ABET-1,ABET-2,ABET-5

Topics	**Learning Outcomes**

• Describe functional requirements using, for example, use cases or user stories.
  – Using at least one method of documenting and structuring functional requirements.
  – Understanding how the method supports design and implementation.
  – Strengths and weaknesses of using a specific approach.
• Requirements elicitation.
  – Sources of requirements, for example, users, administrators, or support personnel.
  – Methods of requirement gathering, for example, surveys, interviews, or behavioral analysis.
• Non-functional requirements, for example, security, usability, or performance, also called as Quality Attributes. Sustainable Design and Pervasive Computing, Environmental Footprint of Computing Systems, Systemic Effects and Social Context
• Risk identification and management, including ethical considerations surrounding the proposed product. Ethics Codes and Legal Accountability, Professional Responsibility and Public Policy, Professional Values, Nature and Continuing Development, Ethical Dissent and Cultural Context, Workplace Ethics
• Communicating and/or formalizing requirement specifications.
• Prototyping a tool for both eliciting and validating/confirming requirements.
• Product evolution: when requirements change, how to understand what effect that has and what changes need to be made.
• Effort estimation:
  – Learning techniques for better estimating the effort required to complete a task;
  – Practicing estimation and comparing it to how long tasks take;
  – Effort estimation is quite difficult, so students are likely to be way off in many cases, but seeing the process play out with their own work is valuable.
	• Compare different methods of eliciting requirements along multiple axes [Compare]
	• Identify differences between two methods of describing functional requirements (e.g., customer interviews, user studies) and the situations where each would be preferred [Analyze]
	• Identify which behaviors are required, allowed, or barred from a given set of requirements and a list of candidate behaviors [Analyze]
	• Collect a set of requirements for a simple software system [Analyze]
	• Identify areas of a software system that need to be changed, given a description of the system and a set of new requirements to be implemented [Analyze]
	• Identify the functional and non-functional requirements in a set of requirements [Analyze]
	• Estimate the time to complete a set of tasks, then compare estimates to the actual time taken [Estimate]
	• Determine an implementation sequence for a set of tasks, adhering to dependencies between them, with a goal to retire risk as early as possible [Determine]
	• Write a requirement specification for a simple software system [Write]

Readings : [**Brookshear2019**]

Unit 22: Machine Learning Fundamentals (1 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Definition and examples of a broad variety of machine learning tasks: – Supervised learning: ∗ Classification ∗ Regression – Reinforcement learning – Unsupervised learning: ∗ Clustering • Fundamental ideas: – No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences. – Sources of error and undecidability in machine learning • A simple statistical-based supervised learning such as linear regression or decision trees: – Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly	• Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain [Explain] • Differentiate the terms of AI, machine learning, and deep learning [Evaluate] • Explain how machine learning works as an optimization/search process [Explain]
Readings : [**Brookshear2019**]	

Unit 23: Neural Networks and Ethics (1 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Basic neural networks: – Fundamentals of understanding how neural networks work and their training process, without details of the calculations – Basic introduction to generative neural networks (e.g., large language models) • Ethics for Machine Learning: – Focus on real data, real scenarios, and case studies – Dataset/algorithmic/evaluation bias and unintended consequences • Deep learning: – Deep feed-forward networks (intuition only, no mathematics) – Convolutional neural networks (intuition only, no mathematics) – Visualization of learned feature representations from deep nets – Other architectures (generative NN, recurrent NN, transformers, etc.) • Ethics for Machine Learning: – Continued focus on real data, real scenarios, and case studies – Privacy – Fairness – Intellectual property – Explainability	• Describe the neural network training process and resulting learned representations [Explain] • Visualize the training progress of a neural network through learning curves in a well-established toolkit (e.g., TensorBoard) and visualize the learned features of the network [Apply] • Given a real-world application of machine learning, describe ethical issues regarding the choices of data, preprocessing steps, algorithm selection, and visualization/presentation of results [Apply]
Readings : [**Brookshear2019**]	

Unit 24: Fundamental Concepts of Graphics and Interactive Techniques (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Overview of computer graphics pipeline and interactive systems • Basic rendering concepts: rasterization, ray tracing, shading models • Color theory and color spaces • Coordinate systems and transformations • Basic interaction techniques: input devices, event handling • Advanced rendering concepts: global illumination, physically based rendering • Graphics hardware and acceleration techniques • Human perception and visual cognition	• Explain the basic stages of the computer graphics pipeline [Explain] • Describe basic rendering techniques such as rasterization and ray tracing [Describe] • Apply color theory to design visually effective graphics [Apply] • Implement coordinate transformations for 2D and 3D graphics [Implement] • Design basic interactive applications using common input devices and event handling [Design] • Compare advanced rendering techniques such as global illumination and physically based rendering [Compare] • Analyze the role of graphics hardware in accelerating rendering tasks [Analyze] • Explain how human perception influences graphics and interaction design [Explain]
Readings : [**Brookshear2019**]	

Unit 25: Applied Rendering and Techniques (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Ray tracing: acceleration structures, sampling, shading • Texture mapping and filtering	• Implement rasterization algorithms for primitive rendering [Implement] • Implement basic ray tracing with acceleration structures [Implement]
Readings : [**Brookshear2019**]	

Unit 26: Closure Class: How Does a Search Engine Like Google Work? (2 hours)	
Competences Expected: ABET-1,ABET-2,ABET-5	
Topics	**Learning Outcomes**
• Problem analysis • The index does not grow linearly with the size of the indexed information. • Response time does not depend on the size of the "database." • Response time does not depend on the number of occurrences found. • Combining various data structures to reach a solution. • Analyzing the scalability of the solution.	• Understand the principles under which a search engine is created [Usar] • Correctly apply data structures to solve the problem [Usar] • Apply concepts related to algorithmic complexity in a search engine [Usar].
Readings : [**Brookshear2019**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Objects-oriented programming I
1.2 Code	:	CS112
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS111. Introduction to Programming. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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 in C++17/C++20, expanding on the foundations from the first course while introducing low-level programming concepts. The course provides a solid foundation for advanced topics covered in the subsequent course of the curriculum.

4. GOALS

- Introduce the student to the foundations of the object orientation paradigm in C++
- Develop skills in C++17/C++20 programming including templates and STL
- Understand memory management and low-level programming concepts
- Develop problem-solving skills using OOP principles
- Master intermediate C++ programming techniques

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Language Pragmatics (2 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Problem domains and programming paradigm • Criteria for good programming language design: – Principles of language design such as orthogonality – Defining control and iteration constructs – Modularization of large software • Brief review of programming paradigms • Comparison between functional and imperative programming • History of programming languages (emphasis on C and C++)	• Discuss the historical context of programming language paradigms [Familiarizarse]. • Evaluate language design trade-offs for specific applications [Evaluar].
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 2: Language Translation and Execution (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Execution models for JIT (Just-In-Time), compiler, interpreter • Use of intermediate code, e.g., bytecode • Execution as native code or within a virtual machine • Virtual machines and intermediate languages • Compilation vs. interpretation models	• Explain and understand the differences between compiled, JIT, and interpreted language implementations, including the benefits and limitations of each [Familiarizarse] • Differentiate syntax and parsing from semantics and evaluation [Usar] • Illustrate ambiguity in parsing using nested if-then-else/arithmetic expression and show resolution using precedence order [Evaluar]
Readings : [**Stroustrup2013**], [**Deitel2017**]	

Unit 3: Type Systems Fundamentals (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• A type as a set of values together with a set of operations: – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records/structs, unions, arrays, lists, functions, references using set operations) • Generic types (parametric polymorphism): – Definition and advantages of polymorphism: parametric, subtyping, overloading, and coercion – Comparison of monomorphic and polymorphic types – Comparison with ad-hoc polymorphism (overloading) and subtype polymorphism – Generic parameters and typing – Use of generic libraries such as collections – Comparison with ad hoc polymorphism (overloading) and subtype polymorphism – Prescriptive vs descriptive polymorphism – Implementation models of polymorphic types – Subtyping • Declaration models (linking, visibility, scope, lifetime) • Overview of type checking in C++ • Type conversion and casting in C++	• Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usar] • Discuss the differences among generics, subtyping, and overloading [Familiarizarse] • Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Evaluar]
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 4: Object-Oriented Programming Fundamentals (14 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Definition of classes: fields, methods, and constructors. • Subclasses, inheritance (including multiple inheritance), and method overriding. • Dynamic dispatch: definition of method-call. • Subtyping: – Subtype polymorphism; implicit upcasts in typed languages. – Notion of behavioral replacement: subtypes acting like supertype. – Relationship between subtyping and inheritance. • Classes and objects in C++ • Constructors and destructors • Access specifiers and encapsulation • Inheritance hierarchies • Polymorphism and virtual functions	• Compose a class through design, implementation, and testing to meet behavioral requirements [Usar] • Build a simple class hierarchy utilizing subclassing that allows code to be reused for distinct subclasses [Usar] • Predict and validate control flow in a program using dynamic dispatch [Evaluar] • Design C++ classes with proper encapsulation [Usar]. • Implement inheritance hierarchies for code reuse [Usar].
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 5: Systems Execution and Memory Model (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Direct, indirect, and indexed access to memory location • Run-time representation of data abstractions such as variables, arrays, vectors, records, pointer-based data elements such as linked-lists and trees, and objects • Low level allocation and accessing of high-level data structures such as basic data types, n-dimensional array, vector, record, and objects • Return from procedure as automatic deallocation mechanism for local data elements in the stack • Manual memory management: allocating, deallocating, and reusing heap memory • Automated memory management: garbage collection as an automated technique using the notion of reachability • Memory layout (stack, heap, static) • Pointer arithmetic and operations • Dynamic memory allocation (new/delete) • Memory leaks and dangling pointers • Smart pointers (unique_ptr, shared_ptr)	• Explain why memory leaks and dangling pointer problems occur, and what can be done by a programmer to avoid/fix them [Usar] • Explain how programming language implementations typically organize memory into global data, text, heap, and stack sections and how features such as recursion and memory management map to this memory model [Evaluar] • Explain how a core language construct, such as data abstractions and control abstractions, is executed [Evaluar] • Implement RAII for resource management [Usar]. • Debug memory-related errors in C++ programs [Usar].
Readings : [**Stroustrup2013**], [**Deitel2017**]	

Unit 6: Functional Programming (10 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Lambda expressions and evaluation: – Variable binding and scope rules. – Parameter-passing. – Nested lambda expressions and reduction order. • Using higher-order functions (taking, returning, and storing functions). • Function templates and class templates • Template specialization • Lambda expressions in C++11/14 • Standard Template Library concepts	• Develop useful functions that take and return other functions [Usar] • Correctly interpret variables and lexical scope in a program using function closures [Evaluar] • Use functional encapsulation mechanisms such as closures and modular interfaces [Usar] • Implement generic functions using templates [Usar]. • Use lambda expressions for functional programming [Usar].
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 7: Object-Oriented Programming: Encapsulation, Subtyping, and Reflection (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Collection classes, iterators, and other common library components. • Sequence containers (vector, list, deque) • Associative containers (set, map, unordered_set) • Container adapters (stack, queue, priority_queue) • Iterators and algorithms • Functors and predicates	• Use collection classes and iterators effectively to solve a problem [Usar] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usar] • Compare and contrast mechanisms to define and protect data elements within procedural, functional, and object-oriented approaches [Evaluar] • Select appropriate STL containers for specific needs [Usar].
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 8: Advanced Programming Constructs (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Object-oriented abstractions: multiple inheritance, mixins, traits, multimethods • Metaprogramming: macros, generative programming, model-based development • Operator overloading • Move semantics and rvalue references • Exception handling in C++ • Namespaces and modular programming	• Use various advanced programming constructs and idioms correctly [Usar] • Discuss how various advanced programming constructs aim to improve program structure, software quality, and programmer productivity [Evaluar] • Discuss how various advanced programming constructs interact with the definition and implementation of other language features [Familiarizarse] • Implement move semantics for performance optimization [Usar].
Readings : [**Stroustrup2013**], [**Josuttis2019**]	

Unit 9: Shell Scripting (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• System commands: – Interface with operating systems • Piping • File abstraction and operators • Programs and processes • Automating compilation and testing of C++ programs through scripts (Makefiles, build/test scripts)	• Create and execute automated scripts to manage various system tasks [Usar]
Readings : [**Robbins2005**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Applied AI
1.2 Code	:	AI161
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS111. Introduction to Programming. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Familiarity)

6. TOPICS

Unit 1: AI Fundamentals and Global Landscape (16 hours)	
Competences Expected: ABET-1,ABET-5	
Topics	**Learning Outcomes**
• What is AI? Definitions, history, and current global landscape. • Types of AI systems: chatbots, image generators, research assistants across regions. • Western AI ecosystems: OpenAI, Anthropic, Google, Microsoft. • Chinese AI ecosystems: DeepSeek, Kimi, ERNIE, Zhipu AI, Baidu. • AI capabilities and limitations: comparative analysis across platforms. • Digital literacy in the AI era: critical thinking about diverse AI outputs.	• Apply fundamental AI knowledge to identify system types and their capabilities [Usar]. • Analyze global intelligent systems and their distinctive characteristics [Evaluar]. • Critically evaluate AI outputs across different cultural contexts [Evaluar].
Readings : [**Ng2024**], [**Mollick2024**], [**UNESCO2023**], [**Tencent2024**]	

Unit 2: Effective AI Interaction and Cross-Platform Prompt Engineering (20 hours)	
Competences Expected: ABET-1,ABET-5	
Topics	**Learning Outcomes**
• Principles of effective prompt writing: clarity, context, constraints across platforms. • Prompt patterns: persona, template, chain-of-thought, few-shot for different AI systems. • Western tools: specific techniques for ChatGPT, Claude, Gemini, Copilot. • Chinese tools: specific features and best practices for DeepSeek, Kimi, ERNIE. • Cross-platform strategies: leveraging different AI strengths for complex tasks. • Iterative refinement: how to improve prompts based on outputs from diverse systems. • Practical sessions: comparative prompt writing workshops across platforms.	• Design effective prompt solutions for diverse AI systems [Evaluar]. • Develop cross-platform workflows that integrate different AI tools [Usar]. • Implement iterative refinement strategies to optimize AI outputs [Usar].
Readings : [**White2023**], [**PromptEngineering2024**], [**Microsoft2024**], [**DeepSeek2024**]	

Unit 3: AI Applications Across Disciplines and Platforms (16 hours)	
Competences Expected: ABET-1,ABET-5	
Topics	**Learning Outcomes**
• AI for research: literature review, data analysis using multiple AI assistants. • AI for writing: academic papers, reports with cross-platform verification. • AI for problem-solving: scientific calculations, engineering design with specialized tools. • AI for creativity: brainstorming, concept development across cultural contexts. • Platform-specific strengths: when to use Western vs. Chinese AI tools. • Discipline-specific workshops: tailored applications using diverse AI ecosystems. • Case studies: real-world applications in scientific research across regions.	• Apply AI fundamentals to solve discipline-specific problems [Usar]. • Design solutions that integrate multiple intelligent systems for complex tasks [Evaluar]. • Develop practical applications using Western and Chinese AI tools [Usar]. • Implement workflows that leverage specific strengths of each platform [Evaluar].
Readings : [**Mollick2023**], [**Google2024**], [**Alibaba2024**], [**OpenAI2024**]	

Unit 4: Global AI Ethics and Responsible Usage (12 hours)	
Competences Expected: ABET-1,ABET-5	
Topics	**Learning Outcomes**
• Understanding AI biases: Western and Eastern cultural perspectives. • Ethical frameworks: comparing EU AI Act, Chinese regulations, and global standards. • Academic integrity: proper citation and AI usage in coursework across platforms. • Privacy and data security: regional differences in AI data handling. • Cultural sensitivity: navigating AI outputs in global contexts. • Environmental impact: sustainability considerations of different AI systems. • Developing personal guidelines for ethical AI usage in international settings. • Case studies: ethical dilemmas in AI deployment across regions.	• Demonstrate professional responsibility in using AI technologies [Evaluar]. • Apply professional ethical principles in global AI contexts [Usar]. • Develop ethical guidelines to mitigate biases in AI systems [Evaluar]. • Evaluate ethical implications of AI deployment across different cultures [Usar].
Readings : [**UNESCO2023**], [**EU2024**], [**Bender2021**], [**ChinaAI2024**], [**Tencent2023**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Calculus I
1.2 Code	:	MA111
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA100. Basic Mathematics. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The course includes knowledge of binary relations and real functions of a real variable. Limits, Continuity, and Derivatives of real functions, applications (maxima and minima), and Differentials applied to the solution of problems involving living beings.

4. GOALS

- Understand the fundamental concepts of limits and continuity.
- Master differentiation techniques for algebraic and transcendental functions.
- Apply derivatives to solve optimization problems and analyze rates of change.
- Develop a rigorous approach to the formal definition of a derivative.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

6. TOPICS

Unit 1: Computational Arithmetic and Error Analysis (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Floating-point arithmetic and the IEEE standard • Sources of error: truncation, rounding, and discretization • Error propagation and condition number • Stability of numerical algorithms • Absolute, relative, and forward/backward error	• Define the different types of errors in numerical computation (rounding, truncation, discretization) [Familiarity] • Explain the principles of floating-point representation and arithmetic according to the IEEE standard [Familiarity] • Calculate the absolute and relative error of an approximation [Usage] • Analyze the propagation of errors in a sequence of arithmetic operations [Usage] • Distinguish between the condition of a problem and the stability of an algorithm [Assessment] • Compute the condition number of a simple problem (e.g., matrix-vector multiplication) [Usage]
Readings : [**Stewart2016**], [**Larson2018**]	

Unit 2: Roots Finding and Basic Numerical Optimization (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Stewart2016**], [**Thomas2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	General Physics
1.2 Code	:	FI101
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA111. Calculus I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The course includes knowledge of the fundamental principles and laws of Physics, Newtonian Classical Mechanics, Kinematics, Statics, Dynamics, Energy and Work, conservative and non-conservative forces, Energy and Momentum. It emphasizes acoustic phenomena, electromagnetic waves, and optics with applications in biological sciences.

4. GOALS

- Understand and apply Newton's laws in the analysis of motion and equilibrium of bodies.
- Apply the concepts of work, energy, and conservation in solving complex mechanical problems.
- Analyze the laws of thermodynamics, fluid behavior, and heat transfer phenomena.
- Study the principles of optics and wave phenomena to understand the nature of light and sound.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Particle and Rigid Body Mechanics (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Physical quantities, systems of units, and vectors. • Kinematics: Motion in one and two dimensions. • Dynamics: Newton's laws and their applications. • Statics and equilibrium of the rigid body.	• Describe the motion of bodies using kinematic and vector magnitudes [Familiarizarse]. • Apply Newton's laws to determine forces and accelerations in physical systems [Usar]. • Solve static equilibrium problems through the analysis of forces and torque [Evaluar].
Readings : [**Sears2013**], [**Tipler2010**]	

Unit 2: Work, Energy, and Momentum (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Work and kinetic energy. • Potential energy and conservation of mechanical energy. • Linear momentum and impulse. • Elastic and inelastic collisions.	• Identify forms of energy and their transformation in conservative physical processes [Familiarizarse]. • Use the work-energy theorem to analyze dynamic systems [Usar]. • Apply the conservation of momentum in solving collision problems [Evaluar].
Readings : [**Sears2013**], [**Resnick2005**]	

Unit 3: Fluid Mechanics and Thermodynamics (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Fluid statics and dynamics (Bernoulli and Archimedes). • Temperature, heat, and laws of thermodynamics. • Kinetic theory of gases. • Thermodynamic processes and heat engines.	• Understand the fundamental properties of fluids and heat transfer mechanisms [Familiarizarse]. • Calculate state variables in thermal processes using the laws of thermodynamics [Usar]. • Model the behavior of fluids and gases through the implementation of fundamental equations [Evaluar].
Readings : [**Sears2013**], [**Serway2018**]	

Unit 4: Waves and Optics (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Wave motion and acoustic phenomena. • Nature and propagation of light. • Geometric optics: Reflection and refraction. • Electromagnetic waves and interference.	• Recognize the characteristics of mechanical and electromagnetic waves in the environment [Familiarizarse]. • Analyze image formation in mirrors and lenses using ray tracing [Usar]. • Evaluate interference and diffraction phenomena based on the wave model of light [Evaluar].
Readings : [**Alonso1971**], [**Sears2009**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Objects-oriented programming II
1.2 Code	:	CS113
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	CS112. Objects-oriented programming I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This is the third course in the sequence of introductory courses in computer science. This course explores advanced C++17/C++20 features focusing on high-performance systems development. Key topics include template metaprogramming, move semantics, perfect forwarding, RAII optimization, multiple/virtual inheritance patterns, concurrency with std::thread and async/await, lock-free programming, modern C++ paradigms such as CRTP and expression templates, and interfacing C++ with other languages. The course prepares students for game engines, high-performance computing, and embedded development where C++ dominates.

4. GOALS

- Master advanced C++17/C++20 features including templates and metaprogramming
- Develop skills in concurrent and systems programming
- Understand performance optimization techniques
- Prepare for high-performance computing applications
- Master template metaprogramming and compile-time computation

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Object-Oriented Programming: Encapsulation, Subtyping, and Reflection (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Collection classes, iterators, and other common library components. • Associative containers (std::set, std::map, unordered variants) • Container adapters (stack, queue, priority_queue) • Advanced STL algorithms • Functors and predicates • Iterator categories and traits	• Use collection classes and iterators effectively to solve a problem [Usar] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Evaluar] • Compare and contrast the benefits and costs/impact of using inheritance (subclasses) and composition (specifically, how to base composition on higher order functions) [Usar] • Implement custom allocators for STL containers [Usar].
Readings : [**Stroustrup2013**], [**Josuttis2019**], [**Vandevoorde2017**]	

Unit 2: Type Systems: Polymorphism and Complementary Typing (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Generic types (parametric polymorphism): – Definition and advantages of polymorphism: parametric, subtyping, overloading, and coercion – Comparison of monomorphic and polymorphic types – Comparison with ad-hoc polymorphism (overloading) and subtype polymorphism – Generic parameters and typing – Use of generic libraries such as collections – Comparison with ad hoc polymorphism (overloading) and subtype polymorphism – Prescriptive vs descriptive polymorphism – Implementation models of polymorphic types – Subtyping • SFINAE (Substitution Failure Is Not An Error): historical foundation of constrained generic programming in C++, the basis of pre-C++20 libraries. • Concepts (C++20): declarative syntax to constrain template parameters, the modern successor to SFINAE with better error diagnostics. • if constexpr (C++17): compile-time branching, replacing much of the tag dispatch previously handled with SFINAE. • Perfect forwarding and universal references • Variadic templates • Compile-time computation with constexpr and consteval	• Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Evaluar] • Discuss the differences among generics, subtyping, and overloading [Evaluar] • Apply Concepts to constrain template parameters and improve error diagnostics [Usar]. • Distinguish when to use Concepts, if constexpr, or legacy SFINAE when maintaining existing C++ code [Evaluar]. • Implement perfect forwarding for efficient argument passing [Usar].
Readings : [**Stroustrup2013**], [**Vandevoorde2017**], [**StroustrupTour2022**]	

Unit 3: Memory Management (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Low level allocation and accessing of high-level data structures such as basic data types, n-dimensional array, vector, record, and objects • Return from procedure as automatic deallocation mechanism for local data elements in the stack • Manual memory management: allocating, deallocating, and reusing heap memory • Automated memory management: garbage collection as an automated technique using the notion of reachability • Lvalues and rvalues in C++11/14/17/20 • Move constructors and move assignment operators • Rvalue references and reference collapsing • Resource Acquisition Is Initialization (RAII) pattern • Smart pointer customization • Memory leak detection and prevention	• Explain why memory leaks and dangling pointer problems occur, and what can be done by a programmer to avoid/fix them [Usar] • Explain how programming language implementations typically organize memory into global data, text, heap, and stack sections and how features such as recursion and memory management map to this memory model [Evaluar] • Explain how a core language construct, such as data abstractions and control abstractions, is executed [Evaluar] • Implement move semantics for resource optimization [Usar]. • Design RAII classes for exception-safe resource management [Usar].
Readings : [**Meyers2014**], [**Stroustrup2013**]	

Unit 4: Parallel and Distributed Computing (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Safety and liveness: – Race conditions – Dependencies/preconditions – Fault models – Termination • Programming models: – Actor models – Procedural and reactive models – Synchronous/asynchronous programming models – Data parallelism • Communication and coordination: – Mutexes – Message-passing – Shared memory – Cobegin-coend – Monitors – Channels – Threads – Guards • std::thread and thread management • std::jthread (C++20): automatic joining (RAII) and cooperative cancellation via stop_token • Mutexes, locks, and condition variables • C++20 synchronization primitives: std::latch, std::barrier, std::counting_semaphore • Atomic operations and lock-free programming • std::async and futures • std::execution and the senders-receivers model (C++26, experimental): a preview of where standard C++ concurrency is heading, presented at a conceptual level only, not evaluated in lab given its limited availability (partial in GCC) • Thread-safe design patterns • Data races and deadlock prevention	• Implement correct concurrent programs using multiple programming models, such as shared memory, actors, futures, synchronization constructs, and data-parallelism primitives [Usar] • Use synchronization constructions such as monitor/synchronized methods in a simple program [Usar] • Use a message-passing model to analyze a communication protocol [Evaluar] • Implement lock-free data structures using atomics [Usar]. • Differentiate std::thread from std::jthread and justify when to use each [Usar]. • Analyze thread-safety of concurrent algorithms [Evaluar].
Readings : [**Williams2019**], [**Stroustrup2013**]	

Unit 5: Advanced Programming Constructs (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Object-oriented abstractions: multiple inheritance, mixins, traits, multimethods • Metaprogramming: macros, generative programming, model-based development • Multiple inheritance and virtual inheritance • CRTP (Curiously Recurring Template Pattern) • deducing this (C++23): simplifies CRTP by removing the need for static_cast<Derived&>(*this) for static polymorphism • Expression templates • Type traits and compile-time introspection • Policy-based design • Interface design for libraries	• Use various advanced programming constructs and idioms correctly [Usar] • Discuss how various advanced programming constructs aim to improve program structure, software quality, and programmer productivity [Evaluar] • Discuss how various advanced programming constructs interact with the definition and implementation of other language features [Evaluar] • Implement CRTP for static polymorphism [Usar]. • Design extensible library interfaces using policy-based design [Usar].
Readings : [**Stroustrup2013**], [**Gregor2017**]	

Unit 6: Systems Execution and Memory Model (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Run-time representation of data abstractions such as variables, arrays, vectors, records, pointer-based data elements such as linked-lists and trees, and objects • Data structures for translation, execution, translation, and code mobility such as stack, heap, aliasing (sharing using pointers), indexed sequence and string • Implementation of the dynamic dispatch introduced in OOP I: virtual function table (vtable), vtable pointer (vptr), and its memory/performance cost versus static dispatch • Interfacing C++ with other languages (FFI) • System calls and OS interaction • Memory-mapped I/O • Performance profiling and benchmarking • Optimization techniques for HPC	• Explain how a core language construct, such as data abstractions and control abstractions, is executed [Usar] • Explain how programming language implementations typically organize memory into global data, text, heap, and stack sections and how features such as recursion and memory management map to this memory model [Evaluar] • Explain why memory leaks and dangling pointer problems occur, and what can be done by a programmer to avoid/fix them [Usar] • Profile and optimize C++ programs for performance [Usar]. • Explain how the compiler implements runtime polymorphism via vtables [Evaluar]. • Interface C++ code with C libraries [Usar].
Readings : [**Stroustrup2013**], [**Williams2019**]	

Unit 7: Advanced Programming Constructs (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Object-oriented abstractions: multiple inheritance, mixins, traits, multimethods • Creational, structural, and behavioral patterns • Thread-safe singleton patterns • Modern C++ design idioms • Dependency injection and inversion of control	• Use various advanced programming constructs and idioms correctly [Usar] • Discuss how various advanced programming constructs aim to improve program structure, software quality, and programmer productivity [Evaluar] • Discuss how various advanced programming constructs interact with the definition and implementation of other language features [Familiarizarse] • Implement thread-safe design patterns [Usar]. • Evaluate architectural trade-offs in C++ systems [Evaluar].
Readings : [**Gamma1994**], [**Stroustrup2013**]	

Unit 8: Advanced Programming Constructs (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Control abstractions: exception handling, continuations, monads. • Compiler optimizations and flags • Benchmarking methodologies • Cache-friendly data structures • Branch prediction and alignment • Reducing compilation time • Continuous integration for performance testing	• Use various advanced programming constructs and idioms correctly [Usar] • Discuss how various advanced programming constructs aim to improve program structure, software quality, and programmer productivity [Evaluar] • Discuss how various advanced programming constructs interact with the definition and implementation of other language features [Familiarizarse] • Apply optimization techniques to improve program performance [Usar]. • Interpret benchmark results to guide optimization decisions [Usar].
Readings : [**Meyers2014**], [**Stroustrup2013**]	

Unit 9: Version Control and CI/CD (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Software process automation: – Build systems - the value of fast, hermetic, reproducible builds, compare/contrast approaches to building a project. – Continuous Integration (CI) - the use of automation and automated tests to do preliminary validation that the current head/trunk revision builds and passes (basic) tests. – Continuous Deployment (CD) - the use of automation to automatically release every change that passes the automated tests to the production environment, ensuring frequent and reliable deliveries. – Dependency management - updating external/upstream dependencies, package management, SemVer. • Build systems for C++: CMake, the de facto industry standard • Modern alternatives: xmake, Meson, and Bazel – comparing declarative approaches versus Lua/Python-based ones • Organizing multi-file projects: translation units, headers, static and dynamic libraries • Dependency management in C++ (vcpkg, Conan)	• Understand the use of CI/CD systems as a ground-truth for the state of the team's shared code (build and test success) [Familiarizarse] • Configure a multi-file C++ project using a declarative build system [Usar]. • Compare CMake with at least one modern alternative (xmake, Meson, or Bazel) in terms of build speed and reproducibility [Evaluar].
Readings : [**Stroustrup2013**], [**StroustrupTour2022**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Discrete Structures I
1.2 Code	:	CS1D1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA100. Basic Mathematics. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Discrete structures are the backbone of computer science. This first course focuses on the fundamental logic and formal reasoning necessary to evaluate the correctness of computational processes. Students will master the formal language of mathematics to avoid ambiguity and will learn various proof techniques essential for the rigorous analysis of algorithms and software specifications.

4. GOALS

- Understand the principles of mathematical logic and propositional calculus.
- Apply predicate logic to model complex constraints in computing.
- Master multiple proof techniques, including direct, contradiction, and induction.
- Develop the ability to formally argue the correctness of mathematical statements.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Familiarity)

6. TOPICS

Unit 1: Discrete Mathematics (24 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Logic: propositional and predicate logic, quantifiers, proofs • Sets: operations, cardinality, power sets, Cartesian products • Relations: properties, equivalence relations, partial orders • Functions: injective, surjective, bijective, composition, inverse • Advanced logic: proof systems, completeness, decidability • Advanced set theory: infinite sets, Cantor's diagonal argument	• Explain propositional and predicate logic [Explain] • Construct simple logical proofs [Apply] • Define basic set operations and relations [Explain] • Define properties of relations [Explain] • Define types of functions [Explain]
Readings : [**Rosen2019**], [**Lehman2018**]	

Unit 2: Discrete Mathematics (24 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Logic: propositional and predicate logic, quantifiers, proofs • Advanced relations: closures, lattices • Advanced functions: recursive functions, computability	• Construct simple logical proofs [Apply] • Apply set operations to solve problems [Apply] • Determine if a relation is an equivalence relation [Apply] • Compose functions and find inverses [Apply]
Readings : [**Rosen2019**], [**Velleman2019**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Platform Based Development
1.2 Code	:	CS2B1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS112. Objects-oriented programming I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Platform-based development focuses on the design and implementation of applications within specific computing environments. This course utilizes the Software Process and Design (SPD) framework to explore the unique constraints and capabilities of Web, Mobile, and Interactive platforms. Students will learn to apply shared architectural concerns and platform-specific APIs to build robust, scalable, and user-centric software solutions.

4. GOALS

- Understand the shared aspects and unique constraints of modern platforms.
- Develop professional applications for Web and Mobile environments.
- Apply interactive computing principles to enhance user experience.
- Design software architectures that leverage platform-specific services.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

6. TOPICS

Unit 1: Common Aspects: Platforms, APIs and Constraints (5 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Overview of development platforms (i.e., web, mobile, game, robotics, embedded, and interactive). – Input/sensors/control devices/haptic devices – Resource constraints ∗ Computational ∗ Data storage ∗ Memory ∗ Communication – Requirements - security, uptime availability, fault tolerance Reliability Concepts, Reliability Engineering, Security Design and Controls Engineering, Threat Analysis and Security Engineering, Trusted Computing and Privacy Engineering – Output/actuators/haptic devices • Programming via platform-specific Application Programming Interface (API) vs traditional application construction • Programming under platform constraints and requirements (e.g., available development tools, development, security considerations) AI and Hardware Security, Access Control and Applications, Security Mindset and Principles	• List the constraints of mobile programming [List]
Readings : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unit 2: Common Aspects: Platform Languages and Web Programming Patterns (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Overview of platform Languages (e.g., Python, Swift, Lua, Kotlin) • Techniques for learning and mastering a platform-specific programming language	• List the characteristics of scripting languages [List] • Describe the three-tier model of web programming [Describe] • Describe how the state is maintained in web programming [Describe]
Readings : [**Fielding:2000:Rest**], [**Gasca2021**]	

Unit 3: Frontend, Frameworks and Web Standards (12 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Web programming languages (e.g., HTML5, JavaScript, Typescript, CSS) • Web platforms, frameworks, or meta-frameworks – Cloud services – API, Web Components • Software as a Service (SaaS) • Web standards such as document object model, accessibility Accessibility and Inclusive Design • Security and Privacy Considerations Computer Crimes and Attacks, Security Laws, Policies and Responsibility	• Compare and contrast web programming with general-purpose programming [Compare] • Describe the differences between Software-as-a-Service (SaaS) and traditional software products [Describe] • Discuss how web standards impact software development [Debate] • Evaluate an existing web application against current web standards [Evaluate]
Readings : [**Flanagan2020**], [**Duckett2014**], [**Fielding:2000:Rest**]	

Unit 4: Mobile Platforms: Development Fundamentals (6 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Development with – Mobile programming languages – Mobile programming environments • Mobile platform constraints – User interface design Understanding the User: Individual goals and interactions with others – Security • Access – Accessing data through APIs Query Construction – Designing API endpoints for mobile apps: pitfalls and design considerations – Network and web interfaces Data Modeling, Fundamentals of Networks and Communications	• Compare mobile programming with general-purpose programming [Compare] • Evaluate the pros and cons of native and cross-platform mobile application development [Evaluate]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 5: Mobile Platforms: Architecture and Constraints (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Development – Native versus cross-platform development – Software design/architecture patterns for mobile applications API Design and Modeling, Principles and Software Architecture, Software Design Quality and Evaluation • Mobile platform constraints – Responsive user interface design Accessibility and Inclusive Design – Heterogeneity and mobility of devices – Differences in user experiences (e.g., between mobile and web-based applications) – Power and performance tradeoff • Specification and testing Software Development Practices, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices	• Develop a location-aware mobile application with data API integration [Create] • Build a sensor-driven mobile application capable of logging data on a remote server [Create] • Create a communication app incorporating telephony and instant messaging [Create]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 6: Mobile Platforms: Capabilities and Asynchronous Computing (3 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Mobile computing affordances – Location-aware applications – Sensor-driven computing (e.g., gyroscope, accelerometer, health data from a watch) – Telephony and instant messaging – Augmented reality Immersion • Asynchronous computing Algorithms – Difference from traditional synchronous programming – Handling success via callbacks – Handling errors asynchronously – Testing asynchronous code and typical problems in testing	• Develop a location-aware mobile application with data API integration [Create] • Build a sensor-driven mobile application capable of logging data on a remote server [Create] • Create a communication app incorporating telephony and instant messaging [Create]
Readings : [**Deitel2017**], [**Hellman2013**]	

Unit 7: SPD-SEP/Interactive (10 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Ethical guidelines when using AI models to assist in journalism and content creation • Accountability for AI-generated outputs • Behavior among prompt programmers and AI developers • Trust with the public when using AI models	• Indicate a framework for accountability in AI model deployment, including clear documentation and attribution [Evaluate] • Discuss ethical codes of conduct and professional standards relevant to prompt programming and AI development [Debate] • Create communication plans and materials to educate the public about AI capabilities, limitations, and ethical safeguards [Create]
Readings : [**Duckett2014**], [**Deitel2017**]	

Unit 8: Interactive Data and Analytics Platforms (6 hours)	
Competences Expected: ABET-2,ABET-4	
Topics	**Learning Outcomes**
• Computational Notebook Platforms (e.g., Jupyter notebooks; Google Colab; R; SPSS; Observable) • Cloud SQL and Data Analysis Platforms (e.g., BigQuery) Query Construction – Apache Spark – Data Visualizations Visualization • Interactive Presentations Backed by Data • Design Tools Requiring Low-Latency Feedback Loops – Rendering tools – Graphic design tools	• Analyze large datasets interactively [Analyze] • Create a backing track for a musical performance, such as live coding [Create] • Create compelling computational notebooks that construct a narrative for a given journalistic goal/story [Create] • Implement interactive code that uses a dataset and generates exploratory graphics [Implement] • Contextualize the attributes of different data analysis styles, such as interactive vs engineered pipeline [Contextualize] • Write a program using a notebook computing platform (e.g., searching, sorting, or graph manipulation) [Write]
Readings : [**Flanagan2020**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Calculus II
1.2 Code	:	MA112
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA111. Calculus I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The course includes knowledge of Integral Calculus: the Indefinite Integral (antiderivative). Integration Techniques. Definite Integral. Integration of hyperbolic and inverse functions. Applications of the Definite Integral (Areas of plane regions, Volumes, Length, Surface Area). Improper Integral. Applications in problems involving living beings.

4. GOALS

- Understand the definite integral as a limit of Riemann sums.
- Master various integration techniques (substitution, parts, partial fractions).
- Apply the Fundamental Theorem of Calculus to solve complex problems.
- Use integrals to calculate areas between curves and volumes of solids of revolution.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Numerical Differentiation and Integration (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Finite difference approximations (forward, backward, central) • Richardson extrapolation • Newton-Cotes formulas (Trapezoidal rule, Simpson's rule) • Gaussian quadrature • Error analysis for numerical integration	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Stewart2016**], [**Thomas2014**]	

Unit 2: Interpolation and Function Approximation (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Stewart2016**], [**Thomas2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Linear Algebra I
1.2 Code	:	MA121
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA100. Basic Mathematics. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Numerical Linear Algebra (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• LU decomposition and its applications • QR decomposition and least squares problems • Singular Value Decomposition (SVD) and its applications • Condition number and numerical stability of linear systems • Iterative methods for linear systems (Jacobi, Gauss-Seidel, Conjugate Gradient)	• Implement LU and QR decomposition algorithms for solving linear systems [Usage] • Explain the geometric and algebraic meaning of the Singular Value Decomposition [Familiarity] • Use SVD for applications such as matrix approximation and principal component analysis [Usage] • Analyze the stability of a linear system using its condition number [Assessment] • Select and implement an appropriate iterative method for solving large, sparse linear systems [Assessment]
Readings : [**Lay2016**], [**Strang2016**]	

Unit 2: Computational Arithmetic and Error Analysis (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Floating-point arithmetic and the IEEE standard • Sources of error: truncation, rounding, and discretization • Error propagation and condition number • Stability of numerical algorithms • Absolute, relative, and forward/backward error	• Define the different types of errors in numerical computation (rounding, truncation, discretization) [Familiarity] • Explain the principles of floating-point representation and arithmetic according to the IEEE standard [Familiarity] • Calculate the absolute and relative error of an approximation [Usage] • Analyze the propagation of errors in a sequence of arithmetic operations [Usage] • Distinguish between the condition of a problem and the stability of an algorithm [Assessment] • Compute the condition number of a simple problem (e.g., matrix-vector multiplication) [Usage]
Readings : [**Lay2016**], [**Strang2016**]	

Unit 3: Roots Finding and Basic Numerical Optimization (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Lay2016**], [**Strang2016**]	

Unit 4: Interpolation and Function Approximation (25 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Lay2016**], [**Anton2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Technical and professional English I
1.2 Code	:	ID101
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	None
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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 pratical 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.

4. GOALS

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

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: Hello everybody! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 2: Meeting people! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 3: The world of work (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 4: Take it easy! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 5: Where do you live? (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 6: Can you speak English? (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 7: Then and now! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Discrete Structures II
1.2 Code	:	CS1D2
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	CS1D1. Discrete Structures I. (3rd Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Continuing from Discrete Structures I, this course explores more complex mathematical structures such as sets, relations, and functions, which are the basis for data modeling in databases and programming languages. Additionally, students will study counting techniques and recurrence relations, providing the analytical tools required to estimate the computational complexity of algorithms.

4. GOALS

- Perform complex operations on sets, functions, and relations.
- Model database relations and function properties formally.
- Apply counting techniques to analyze search spaces and probability.
- Solve recurrence relations to analyze the efficiency of recursive algorithms.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Familiarity)

6. TOPICS

Unit 1: Discrete Mathematics (24 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Sets: operations, cardinality, power sets, Cartesian products • Relations: properties, equivalence relations, partial orders • Functions: injective, surjective, bijective, composition, inverse	• Define basic set operations and relations [Evaluar] • Apply set operations to solve problems [Usar] • Define properties of relations [Evaluar] • Determine if a relation is an equivalence relation [Usar] • Define types of functions [Evaluar] • Compose functions and find inverses [Usar]
Readings : [**Rosen2019**], [**Scheinerman2012**]	

Unit 2: Discrete Mathematics (24 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Counting: permutations, combinations, binomial theorem • Advanced counting: generating functions, inclusion-exclusion • Recurrence relations: solving simple recurrences • Advanced recurrence: generating functions, asymptotic behavior	• Explain permutations and combinations [Evaluar] • Apply counting techniques to solve problems [Usar] • Define recurrence relations [Evaluar] • Solve simple recurrence relations [Usar] • Develop generating functions for counting problems [Usar] • Apply generating functions to solve complex recurrence relations [Evaluar]
Readings : [**Rosen2019**], [**Grimaldi2003**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Algorithms and Data Structures
1.2 Code	:	CS210
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS113. Objects-oriented programming II. (3^{rd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This course provides a comprehensive study of fundamental data structures and algorithms with emphasis on implementation using advanced C++ features. Building on prior C++ knowledge, students will implement generic data structures using templates and traits, ensuring type flexibility and performance. The course covers linear structures (vectors, various linked lists), tree structures (binary, AVL, B-trees), heaps, hash tables, tries, and concurrent data structures. Special focus is placed on modern C++ techniques including variadic templates for generic operations and concurrent programming for thread-safe data structures.

4. GOALS

- Implement generic data structures using templates and traits.
- Master advanced tree structures and balanced trees.
- Understand and implement concurrent data structures.
- Apply variadic templates for generic operations on containers.
- Analyze algorithmic complexity of all implemented structures.
- Design efficient data structures for specific problem constraints.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Advanced Foundation for Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Template metaprogramming review and advanced traits. • Type traits and SFINAE for conditional compilation. • Variadic templates and parameter packs. • Perfect forwarding and universal references. • Lambda expressions with generic captures. • Policy-based design and CRTP (Curiously Recurring Template Pattern). • Concept-based constraints (C++20 concepts if applicable).	• Design type traits for customizing data structure behavior [Usar]. • Implement variadic templates for generic container operations [Usar]. • Apply policy-based design to create configurable data structures [Evaluar]. • Use perfect forwarding to implement efficient constructors [Usar].
Readings : [**Stroustrup2013**], [**Vandevoorde2017**]	

Unit 2: Foundational Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Abstract Data Type (ADT) and operations on an ADT: – Dictionary operations (insert, delete, find) • Arrays: – Numeric vs non-numeric, character strings – Single (vector) vs multidimensional (matrix) • Records/Structs/Tuples and Objects • Linked lists (for historical reasons): – Single vs Double and Linear vs Circular • Stacks • Queues and deques: – Heap-based priority queue • Hash tables/maps: – Collision resolution and complexity (e.g., probing, chaining, rehash) • Trees: – Binary, n-ary, and search trees – Balanced (e.g., AVL, Red-Black, Heap) • Sets	• For each ADT/Data-Structure in this unit: – Explain its definition, properties, representation(s), and associated ADT operations. – Explain step-by-step how the ADT operations associated with the data structure transform it. [Usar] • For each algorithm in this unit explain step-by-step how the algorithm operates [Usar] • Implement vector with customizable allocation policy using templates [Usar]. • Design generic linked list nodes with trait-based memory management [Usar]. • Compare time/space complexity of different linear structures [Evaluar]. • Apply variadic templates to initialize containers with multiple elements [Usar].
Readings : [**Cormen2009**], [**Stroustrup2013**]	

Unit 3: Foundational Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Trees: – Binary, n-ary, and search trees – Balanced (e.g., AVL, Red-Black, Heap) • Binary tree template implementation with iterator support. • Binary search tree with trait-based comparison operators. • Tree traversal algorithms (iterative and recursive). • Expression trees and their applications. • Template-based node structures with configurable payload.	• Given requirements for a problem, develop multiple solutions using various data structures and algorithms. Subsequently, evaluate the suitability, strengths, and weaknesses selecting an approach that best satisfies the requirements [Evaluar] • Explain factors beyond computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarizarse] • Implement generic binary tree with configurable traversal order [Usar]. • Design expression tree evaluator using visitor pattern [Usar]. • Analyze tree operation complexities for different balancing strategies [Evaluar].
Readings : [**Cormen2009**], [**Knuth1997**]	

Unit 4: Balanced and Multi-way Trees (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• AVL tree implementation with template-based balancing trait. • Red-black tree properties and operations. • B-Tree template implementation for disk-based storage. • 2-3 Tree and 2-3-4 Tree structures. • Digital trie (prefix tree) implementation. • Suffix trees and their applications in string processing. • Node splitting/merging strategies for B-trees.	• Implement self-balancing trees using policy-based design [Usar]. • Design B-tree for database index applications [Usar]. • Evaluate trade-offs between different balanced tree structures [Evaluar]. • Implement trie for efficient string prefix matching [Usar].
Readings : [**Cormen2009**], [**Knuth1997**], [**Sedgewick2011**]	

Unit 5: Foundational Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Trees: – Binary, n-ary, and search trees – Balanced (e.g., AVL, Red-Black, Heap) • Binary heap implementation using array representation. • Min-heap and max-heap variants. • Heapify algorithms and complexity analysis. • Priority queue ADT implementation. • Binomial heap and Fibonacci heap overview. • Heap applications: Dijkstra's algorithm, heap sort.	• Explain the heap property and the use of heaps as an implementation of a priority queue [Familiarizarse] • Implement generic heap with configurable comparison trait [Usar]. • Design priority queue supporting multiple priority update strategies [Usar]. • Analyze heap operations complexity for different heap types [Evaluar]. • Apply heap in graph algorithm implementations [Usar].
Readings : [**Cormen2009**], [**Sedgewick2011**]	

Unit 6: Foundational Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Hash tables/maps: – Collision resolution and complexity (e.g., probing, chaining, rehash) • Hash function design and properties. • Separate chaining implementation with linked lists. • Open addressing: linear probing, quadratic probing, double hashing. • Perfect hashing and minimal perfect hashing. • Cuckoo hashing and hopscotch hashing. • Hash table resizing strategies and load factor management.	• Explain how collision avoidance and collision resolution is handled in hash tables [Familiarizarse] • Design generic hash table with configurable collision resolution [Usar]. • Implement custom hash functions for user-defined types [Usar]. • Evaluate performance of different hashing schemes [Evaluar]. • Apply hash tables in compiler symbol table implementation [Usar].
Readings : [**Cormen2009**], [**Knuth1997**]	

Unit 7: Concurrent Data Structures (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Thread-safe data structure design principles. • Lock-based concurrent linked lists. • Lock-free and wait-free algorithms. • Concurrent hash tables with fine-grained locking. • Atomic operations and memory ordering. • Concurrent queues (bounded and unbounded). • Concurrent stacks and elimination techniques.	• Design thread-safe data structures using mutex and lock guards [Usar]. • Implement lock-free linked list using atomic operations [Usar]. • Analyze performance of concurrent vs sequential data structures [Evaluar]. • Apply concurrent queues in producer-consumer patterns [Usar].
Readings : [**Herlihy2012**], [**Williams2019**]	

Unit 8: Specialized Structures and Applications (8 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Persistent data structures. • Disjoint-set union (union-find) with path compression. • Bloom filters and their probabilistic properties. • Skip lists for probabilistic balanced search. • Segment trees and Fenwick trees for range queries. • Spatial data structures: k-d trees, quadtrees. • Cache-oblivious data structures.	• Implement persistent data structures with structural sharing [Usar]. • Design union-find with both union by rank and path compression [Usar]. • Apply specialized structures to solve domain-specific problems [Evaluar]. • Evaluate space-time tradeoffs in probabilistic data structures [Evaluar].
Readings : [**Okasaki1999**], [**Sedgewick2011**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Data Management
1.2 Code	:	CS271
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	• CS112. Objects-oriented programming I. (2nd Sem) • CS1D1. Discrete Structures I. (3rd Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This course introduces fundamental concepts and techniques in data management, focusing on relational database systems. Students will learn data modeling using entity-relationship diagrams, relational database design principles, normalization theory, and Structured Query Language (SQL). The course covers core database system components, transaction management, and basic query processing. Emphasis is placed on practical skills for designing, implementing, and querying databases, as well as understanding the role of data in organizations and the associated ethical considerations in data management.

4. GOALS

- Design and implement normalized relational databases.
- Formulate complex queries using SQL for data retrieval and manipulation.
- Apply data modeling techniques using entity-relationship diagrams.
- Understand database system architecture and transaction management.
- Recognize ethical and security considerations in data management.
- Utilize basic database administration tools and techniques.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Core Database System Concepts (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Purpose and advantages of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture, data independence, and data abstraction • Transaction management • Normalization • Approaches for managing large volumes of data (e.g., NoSQL database systems, use of MapReduce) • How to support CRUD-only applications • Distributed databases/cloud-based systems • Structured, semi-structured, and unstructured data • Use of a declarative query language • Systems supporting structured and/or stream content	• Identify at least four advantages that using a database system provides [Familiarizarse] • Enumerate the components of a (relational) database system [Familiarizarse] • Explain the advantages of database systems over file systems [Usar]. • Describe the three-schema architecture and data independence [Familiarizarse]. • Identify different types of database users and their roles [Familiarizarse].
Readings : [**Ramakrishnan2002**], [**Silberschatz2019**]	

Unit 2: Data Modeling (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Conceptual models (e.g., entity-relationship, UML diagrams) • Entity sets, relationship sets, and attributes • Cardinality constraints: one-to-one, one-to-many, many-to-many • Weak entity sets and identifying relationships • Extended E-R features: specialization, generalization, aggregation • UML class diagrams for data modeling	• Describe the components of the E-R (or some other non-relational) data model [Familiarizarse] • Model a given environment using a conceptual data model [Usar] • Design E-R diagrams for real-world scenarios [Usar]. • Convert E-R diagrams to relational schemas [Usar]. • Apply extended E-R features for complex modeling [Evaluar].
Readings : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unit 3: Relational Databases (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Entity and referential integrity: Candidate key, superkeys • Relational database design • Relational algebra operations: select, project, join, union, etc. • Integrity constraints: domain, key, entity, referential • Relational calculus: tuple and domain calculus • Mapping E-R diagrams to relational schemas	• Describe the defining characteristics behind the relational data model [Familiarizarse] • Comment on the difference between a foreign key and a superkey [Usar] • Express queries using relational algebra [Usar]. • Design relational schemas with appropriate constraints [Usar]. • Map complex E-R diagrams to normalized relations [Evaluar].
Readings : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unit 4: Query Construction (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• SQL Query Formation: – Interactive SQL execution – Programmatic execution of an SQL query • Data definition language: CREATE, ALTER, DROP • Data types and domain constraints reation and management • View definition and materialized views • Database schema design and implementation	• Compose SQL queries that incorporate select, project, join, union, intersection, set difference, and set division [Usar] • Define, in SQL, a relation schema, including all integrity constraints and delete/update triggers [Usar] • Create and modify database schemas using SQL DDL [Usar]. • Implement integrity constraints at schema level [Usar]. • Design and create appropriate indexes for performance [Evaluar].
Readings : [**Ramakrishnan2002**], [**Beaulieu2009**]	

Unit 5: Query Construction (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Complex SQL queries: nested subqueries, correlated subqueries • Aggregation functions and GROUP BY operations • Set operations: UNION, INTERSECT, EXCEPT • JOIN operations: inner, outer, natural, cross joins • Data manipulation: INSERT, UPDATE, DELETE • Transaction control: COMMIT, ROLLBACK, SAVEPOINT	• Determine when a nested SQL query is correlated or not [Usar] • Compose an SQL query to update a tuple in a relation [Usar] • Formulate complex queries with multiple joins and subqueries [Usar]. • Apply aggregation for data analysis and reporting [Usar]. • Implement data modification operations with transaction control [Evaluar].
Readings : [**Ramakrishnan2002**], [**Beaulieu2009**]	

Unit 6: Relational Databases (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Introduction to Functional dependency theory • Normalization Theory: – Decomposition of a schema; lossless-join, and dependency-preservation properties of a decomposition – Normal forms (BCNF) – Denormalization (for efficiency) • Functional dependencies and inference rules • Normal forms: 1NF, 2NF, 3NF, BCNF • Decomposition algorithms: lossless-join and dependency-preserving • Practical considerations in normalization	• Describe how functional dependency theory generalizes the notion of key [Familiarizarse] • Detect which normal form a given decomposition yields [Usar] • Identify functional dependencies in relational schemas [Usar]. • Normalize relations to BCNF [Evaluar]. • Evaluate decompositions for lossless-join and dependency preservation [Evaluar].
Readings : [**Ramakrishnan2002**], [**Elmasri2015**]	

Unit 7: DBMS Internals (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Transaction Management: – Isolation Levels – ACID – Serializability – Distributed Transactions • Concurrency Control: – 2-Phase Locking – Deadlocks handling strategies – Quorum-based consistency models • ACID properties: Atomicity, Consistency, Isolation, Durability • Serializability and conflict serializability • Lock-based concurrency control: two-phase locking • Deadlock detection and prevention • Isolation levels and their trade-offs	• Describe the four properties for a correct transaction manager [Familiarizarse] • Outline the principle of serializability [Familiarizarse] • Explain ACID properties and their importance [Usar]. • Analyze transaction schedules for serializability [Evaluar]. • Apply locking protocols for concurrency control [Usar].
Readings : [**Ramakrishnan2002**], [**Bernstein1987**]	

Unit 8: Data Security and Privacy (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Differences between data security and data privacy • Personally identifying information (PII) and its protection • Ethical considerations in ensuring the security and privacy of data • Database security: authentication, authorization, encryption • SQL injection attacks and prevention • Introduction to NoSQL databases and big data • Data governance and compliance	• Describe the differences in the goals for data security and data privacy [Familiarizarse] • Identify and mitigate risks associated with different approaches to protecting data [Usar] • Implement basic database security measures [Usar]. • Prevent SQL injection vulnerabilities [Evaluar]. • Discuss ethical considerations in data collection and use [Evaluar].
Readings : [**Ramakrishnan2002**], [**Elmasri2015**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Linear Algebra II
1.2 Code	:	MA122
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA121. Linear Algebra I. (3rd Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Numerical Linear Algebra (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• LU decomposition and its applications • QR decomposition and least squares problems • Singular Value Decomposition (SVD) and its applications • Condition number and numerical stability of linear systems • Iterative methods for linear systems (Jacobi, Gauss-Seidel, Conjugate Gradient)	• Implement LU and QR decomposition algorithms for solving linear systems [Usage] • Explain the geometric and algebraic meaning of the Singular Value Decomposition [Familiarity] • Use SVD for applications such as matrix approximation and principal component analysis [Usage] • Analyze the stability of a linear system using its condition number [Assessment] • Select and implement an appropriate iterative method for solving large, sparse linear systems [Assessment]
Readings : [**Strang2016**], [**Lay2016**]	

Unit 2: Computational Arithmetic and Error Analysis (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Floating-point arithmetic and the IEEE standard • Sources of error: truncation, rounding, and discretization • Error propagation and condition number • Stability of numerical algorithms • Absolute, relative, and forward/backward error	• Define the different types of errors in numerical computation (rounding, truncation, discretization) [Familiarity] • Explain the principles of floating-point representation and arithmetic according to the IEEE standard [Familiarity] • Calculate the absolute and relative error of an approximation [Usage] • Analyze the propagation of errors in a sequence of arithmetic operations [Usage] • Distinguish between the condition of a problem and the stability of an algorithm [Assessment] • Compute the condition number of a simple problem (e.g., matrix-vector multiplication) [Usage]
Readings : [**Strang2016**], [**Lay2016**]	

Unit 3: Interpolation and Function Approximation (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Strang2016**], [**Lay2016**]	

Unit 4: Numerical Methods for Ordinary Differential Equations (ODEs) (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Strang2016**], [**Trefethen1997**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Calculus III
1.2 Code	:	MA210
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA112. Calculus II. (3^{rd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The nature of the course is theoretical-practical; its purpose is to develop a conceptual and methodological framework in students aimed at providing a solid mathematical foundation for understanding all natural phenomena. It includes the following thematic units: Differential calculus in several variables. Integral calculus in several variables. Vector calculus.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Numerical Differentiation and Integration (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Finite difference approximations (forward, backward, central) • Richardson extrapolation • Newton-Cotes formulas (Trapezoidal rule, Simpson's rule) • Gaussian quadrature • Error analysis for numerical integration	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Stewart2016**], [**Thomas2014**]	

Unit 2: Numerical Methods for Ordinary Differential Equations (ODEs) (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Euler's method and its variants (forward, backward) • Runge-Kutta methods • Multistep methods (Adams-Bashforth, Adams-Moulton) • Stability and convergence analysis for ODE solvers • Systems of ODEs and higher-order ODEs	• Implement basic ODE solvers (Euler, Runge-Kutta) [Usage] • Analyze the stability and convergence properties of ODE numerical methods [Assessment] • Select an appropriate numerical method for a given ODE based on stability and accuracy requirements [Assessment] • Solve systems of ODEs and convert higher-order ODEs to first-order systems [Usage]
Readings : [**Stewart2016**], [**Thomas2014**]	

Unit 3: Interpolation and Function Approximation (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Stewart2016**], [**Thomas2014**]	

Unit 4: Roots Finding and Basic Numerical Optimization (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Stewart2016**], [**Thomas2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Technical and professional English II
1.2 Code	:	ID102
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	ID101. Technical and professional English I. (3^{rd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 6 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. GOALS

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

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: How long ago? (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 2: Food you like! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 3: The world of work (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 4: Looking good! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 5: Life is an adventure! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 6: You‘re pretty smart! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

Unit 7: Have you ever? (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars021S**], [**Cambridge06**], [**MacGrew99**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Computer graphics
1.2 Code	:	CS251
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	CS210. Algorithms and Data Structures. (4^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Computer Graphics is a fundamental area of Computer Science that focuses on the mathematical and computational techniques for generating and manipulating images. This course provides students with the foundations of 2D and 3D modeling, rendering pipelines, and the physical principles of light and color, enabling them to develop high-performance visual applications, simulations, and interactive systems.

4. GOALS

- Understand the graphics pipeline and coordinate system transformations.
- Implement rendering algorithms for realistic and real-time visualization.
- Apply mathematical modeling to create and manipulate 3D objects.

5. COMPETENCES

No items for this table

6. TOPICS

Unit 1: Fundamental Concepts of Graphics and Interactive Techniques (16 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Overview of computer graphics pipeline and interactive systems • Basic rendering concepts: rasterization, ray tracing, shading models • Color theory and color spaces • Coordinate systems and transformations • Basic interaction techniques: input devices, event handling • Advanced rendering concepts: global illumination, physically based rendering • Graphics hardware and acceleration techniques • Human perception and visual cognition	• Explain the basic stages of the computer graphics pipeline [Explain] • Describe basic rendering techniques such as rasterization and ray tracing [Describe] • Apply color theory to design visually effective graphics [Apply] • Implement coordinate transformations for 2D and 3D graphics [Implement] • Design basic interactive applications using common input devices and event handling [Design] • Compare advanced rendering techniques such as global illumination and physically based rendering [Compare] • Analyze the role of graphics hardware in accelerating rendering tasks [Analyze] • Explain how human perception influences graphics and interaction design [Explain]
Readings : [**Marschner2021**], [**Hearn2014**]	

Unit 2: Applied Rendering and Techniques (16 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Rasterization algorithms: scan conversion, antialiasing • Ray tracing: acceleration structures, sampling, shading • Texture mapping and filtering • Shading models: Lambertian, Phong, BRDF • Advanced techniques: shadow mapping, environment mapping, deferred rendering	• Implement rasterization algorithms for primitive rendering [Implement] • Implement basic ray tracing with acceleration structures [Implement] • Apply texture mapping and filtering techniques to enhance visual realism [Apply] • Compare different shading models and their applications [Compare] • Implement advanced rendering techniques such as shadow mapping or deferred rendering [Implement]
Readings : [**Marschner2021**], [**Hearn2014**], [**Angel2014**]	

Unit 3: Geometric Modeling (16 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Geometric representations: polygons, curves, surfaces, implicit forms • Modeling techniques: splines, subdivision surfaces, constructive solid geometry • Procedural modeling and generation • Geometric algorithms: intersection, distance, decomposition	• Describe different geometric representations and their trade-offs [Describe] • Implement modeling techniques such as spline curves or subdivision surfaces [Implement] • Create procedural models using algorithms or rules [Create] • Apply geometric algorithms to solve problems like intersection testing or mesh processing [Apply]
Readings : [**Marschner2021**], [**Angel2014**]	

Unit 4: Visualization (16 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Principles of data visualization: perception, cognition, and effective encoding • Types of visualizations: scientific, information, geographical • Visualization techniques for multivariate, temporal, and hierarchical data • Interactive visualization and exploration • Evaluation of visualization effectiveness	• Apply principles of perception and cognition to design effective visualizations [Apply] • Select appropriate visualization techniques for different data types and tasks [Evaluate] • Implement interactive visualizations for data exploration [Implement] • Evaluate the effectiveness of a visualization using empirical methods [Evaluate]
Readings : [**Marschner2021**]	

Unit 5: Shading and Advanced Rendering (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Advanced shading: physically based shading, subsurface scattering • Global illumination: radiosity, photon mapping, path tracing • Real-time rendering techniques: level of detail, culling, occlusion • GPU programming and shaders	• Implement advanced shading models for realistic materials [Implement] • Compare global illumination algorithms and their trade-offs [Compare] • Optimize real-time rendering using techniques like level of detail and culling [Create] • Develop GPU shaders for custom rendering effects [Create]
Readings : [**Marschner2021**], [**Angel2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Calculus IV
1.2 Code	:	MA211
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	• MA210. Calculus III. (4^{th} Sem) • MA121. Linear Algebra I. (3^{rd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The nature of the course is theoretical-practical; its purpose is to develop a conceptual and methodological framework aimed at providing a solid mathematical foundation for understanding all natural phenomena. It includes the following thematic units: Sequences in R^n, C, (R^m, R). Directional derivatives and implicit differentiation. Isometries of R^n, conics, and quadric surfaces. Vector fields.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Numerical Differentiation and Integration (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Finite difference approximations (forward, backward, central) • Richardson extrapolation • Newton-Cotes formulas (Trapezoidal rule, Simpson's rule) • Gaussian quadrature • Error analysis for numerical integration	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Marsden2003**], [**Stewart2016**]	

Unit 2: Numerical Methods for Ordinary Differential Equations (ODEs) (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Euler's method and its variants (forward, backward) • Runge-Kutta methods • Multistep methods (Adams-Bashforth, Adams-Moulton) • Stability and convergence analysis for ODE solvers • Systems of ODEs and higher-order ODEs	• Implement basic ODE solvers (Euler, Runge-Kutta) [Usage] • Analyze the stability and convergence properties of ODE numerical methods [Assessment] • Select an appropriate numerical method for a given ODE based on stability and accuracy requirements [Assessment] • Solve systems of ODEs and convert higher-order ODEs to first-order systems [Usage]
Readings : [**Marsden2003**], [**Stewart2016**]	

Unit 3: Interpolation and Function Approximation (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Marsden2003**], [**Stewart2016**]	

Unit 4: Roots Finding and Basic Numerical Optimization (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Marsden2003**], [**Stewart2016**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Numerical Analysis I
1.2 Code	:	MA221
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	• MA112. Calculus II. (3^{rd} Sem) • MA122. Linear Algebra II. (4^{th} Sem) • AI161. Applied AI. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Computational Arithmetic and Error Analysis (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Floating-point arithmetic and the IEEE standard • Sources of error: truncation, rounding, and discretization • Error propagation and condition number • Stability of numerical algorithms • Absolute, relative, and forward/backward error	• Define the different types of errors in numerical computation (rounding, truncation, discretization) [Familiarity] • Explain the principles of floating-point representation and arithmetic according to the IEEE standard [Familiarity] • Calculate the absolute and relative error of an approximation [Usage] • Analyze the propagation of errors in a sequence of arithmetic operations [Usage] • Distinguish between the condition of a problem and the stability of an algorithm [Assessment] • Compute the condition number of a simple problem (e.g., matrix-vector multiplication) [Usage]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 2: Numerical Linear Algebra (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• LU decomposition and its applications • QR decomposition and least squares problems • Singular Value Decomposition (SVD) and its applications • Condition number and numerical stability of linear systems • Iterative methods for linear systems (Jacobi, Gauss-Seidel, Conjugate Gradient)	• Implement LU and QR decomposition algorithms for solving linear systems [Usage] • Explain the geometric and algebraic meaning of the Singular Value Decomposition [Familiarity] • Use SVD for applications such as matrix approximation and principal component analysis [Usage] • Analyze the stability of a linear system using its condition number [Assessment] • Select and implement an appropriate iterative method for solving large, sparse linear systems [Assessment]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 3: Roots Finding and Basic Numerical Optimization (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 4: Interpolation and Function Approximation (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Burden2016**], [**Chapra2015**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Real Analysis I
1.2 Code	:	MA222
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• MA112. Calculus II. (3rd Sem) • ID101. Technical and professional English I. (3rd Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The subject belongs to the area of Specific Studies; the nature of the course is theoretical-practical, and its purpose is oriented toward providing a solid mathematical foundation for understanding all natural phenomena. It includes the following thematic units: Countability. Sequences of real numbers. Topology of R. Limit and continuity of real functions of a real variable. Derivation. Riemann-Stieltjes integral.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Computational Arithmetic and Error Analysis (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Floating-point arithmetic and the IEEE standard • Sources of error: truncation, rounding, and discretization • Error propagation and condition number • Stability of numerical algorithms • Absolute, relative, and forward/backward error	• Define the different types of errors in numerical computation (rounding, truncation, discretization) [Familiarity] • Explain the principles of floating-point representation and arithmetic according to the IEEE standard [Familiarity] • Calculate the absolute and relative error of an approximation [Usage] • Analyze the propagation of errors in a sequence of arithmetic operations [Usage] • Distinguish between the condition of a problem and the stability of an algorithm [Assessment] • Compute the condition number of a simple problem (e.g., matrix-vector multiplication) [Usage]
Readings : [**Rudin1976**], [**Bartle2011**]	

Unit 2: Roots Finding and Basic Numerical Optimization (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bisection method and fixed-point iteration • Newton's method and secant method for root finding • Convergence analysis and order of convergence • Gradient descent for univariate optimization • Rate of convergence and stopping criteria	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Rudin1976**], [**Bartle2011**]	

Unit 3: Interpolation and Function Approximation (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Rudin1976**], [**Bartle2011**]	

Unit 4: Numerical Differentiation and Integration (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Finite difference approximations (forward, backward, central) • Richardson extrapolation • Newton-Cotes formulas (Trapezoidal rule, Simpson's rule) • Gaussian quadrature • Error analysis for numerical integration	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Rudin1976**], [**Bartle2011**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Introduction to Statistics
1.2 Code	:	MA251
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	AI161. Applied AI. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Write justification for this course here ...

4. GOALS

- Write your first goal here..
- Write your second goal here..

5. COMPETENCES

No items for this table

6. TOPICS

Unit 1: Unit title (2 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Topic1 • Topic2	• LearningOutcome1 [Familiarizarse]. • LearningOutcome2 [Usar]. • LearningOutcome3 [Evaluar].
Readings : [**Bibitem1**], [**Bibitem2**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Numbers theory
1.2 Code	:	MA281
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	2
1.5 Prerrequisites	:	MA100. Basic Mathematics. (1^{st} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Number Theory is a fundamental branch of pure mathematics that studies the properties of integers, serving as the foundation for numerous applications in computational mathematics, cryptography, and information theory. This course equips students with the theoretical and algorithmic tools to analyze problems of divisibility, primality, congruences, and Diophantine equations, developing rigorous logical-abstract thinking. The computational focus allows for the implementation and validation of theoretical concepts, laying the groundwork for areas such as public-key cryptography and factorization algorithms.

4. GOALS

- Understand and apply the fundamental principles of divisibility, the Euclidean algorithm, and the fundamental theorem of arithmetic.
- Analyze and solve problems involving linear congruences and systems of congruences using the Chinese Remainder Theorem.
- Study basic arithmetic functions and their applications in the distribution of prime numbers.
- Introduce the concepts of quadratic residues and the law of quadratic reciprocity, understanding their theoretical importance.
- Implement basic numerical algorithms (Euclid's algorithm, primality testing, modular exponentiation) to validate theoretical concepts.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Divisibility and Congruences (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Axioms of the integers: Well-Ordering Principle and Mathematical Induction. • Divisibility: Division Algorithm, Greatest Common Divisor (Extended Euclidean Algorithm) and Bézout's Lemma. • Prime and composite numbers. Fundamental Theorem of Arithmetic. Infinitude of primes. • Congruences: definition, algebraic properties, and residue classes modulo n. • Computational applications: implementation of the Euclidean algorithm and basic primality tests.	• Understand the fundamental axioms of the integers and the proof technique of induction [Familiarizarse]. • Use the extended Euclidean algorithm to compute the greatest common divisor and the Bézout identity [Usar]. • Prove basic properties of prime numbers and congruences, applying them to elementary problems [Evaluar].
Readings : [**Rosen2011**], [**Apostol1976**]	

Unit 2: Diophantine Equations and the Chinese Remainder Theorem (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Linear Diophantine equations: conditions for solvability and general solution method. • Chinese Remainder Theorem for systems of linear congruences with coprime moduli. • Modular arithmetic: the ring $\mathbb{Z}_n$, units, and zero divisors. • Fermat's and Euler's Theorems. Euler's totient function ϕ and its properties. • Applications in basic cryptography: Caesar cipher and foundations of RSA.	• Identify the conditions for the existence of solutions to linear Diophantine equations [Familiarizarse]. • Apply the Chinese Remainder Theorem to solve systems of congruences and calendar problems [Usar]. • Use Fermat's and Euler's theorems to simplify modular exponentiation calculations and analyze basic algebraic structures [Evaluar].
Readings : [**Rosen2011**], [**Burton2011**]	

Unit 3: Arithmetic Functions and Multiplicative Number Theory (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Arithmetic functions: Definition and examples (tau, sigma, phi, mu). • Dirichlet convolution and the Möbius inversion formula. • Perfect, amicable, and abundant numbers. Open conjectures. • Orders and primitive roots modulo a prime. • Discrete logarithms: definition and applications in key exchange protocols (Diffie-Hellman).	• Recognize the properties of the most important arithmetic functions [Familiarizarse]. • Apply Möbius inversion to solve counting problems in number theory [Usar]. • Compute orders and primitive roots, relating them to the cyclic structure of $\mathbb{Z}_p^*$ [Evaluar].
Readings : [**Apostol1976**], [**Nathanson2000**]	

Unit 4: Quadratic Residues and Computational Applications (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Quadratic residues and non-residues. Legendre symbol and its properties. • Law of Quadratic Reciprocity. Proof and applications in computing the Legendre symbol. • Jacobi symbol and its use in probabilistic primality tests. • Introduction to primality testing: Miller-Rabin test. • Integer factorization: Pollard's $p-1$ method and notions of the RSA algorithm.	• Determine whether a number is a quadratic residue modulo a prime using the Legendre symbol [Familiarizarse]. • Apply the Law of Quadratic Reciprocity to compute Legendre and Jacobi symbols efficiently [Usar]. • Implement basic primality testing and factorization algorithms, analyzing their complexity and limitations [Evaluar].
Readings : [**Rosen2011**], [**Hoffstein2008**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Technical and professional English III
1.2 Code	:	ID201
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	ID102. Technical and professional English II. (4^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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

4. GOALS

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

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: Getting to know you! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 2: The way we live! (0 hours)	
Competences Expected: ABET-4,ABET-6	
Topics	**Learning Outcomes**
• Simple present tense. • Present Continuous Time. • Collocations. • Vocabulary of the countries of the world. • Expressions of anger. • Connectors.	• Ath 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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 3: It all went wrong! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 4: Let's go shopping! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 5: What do you want to do? (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 6: The best in the world! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 7: Fame! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Ordinary Differential Equations
1.2 Code	:	MA212
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	6
1.5 Prerrequisites	:	• CS112. Objects-oriented programming I. (2^{nd} Sem) • MA121. Linear Algebra I. (3^{rd} Sem) • MA222. Real Analysis I. (5^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	4 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Ordinary Differential Equations course provides the mathematical tools to analyze systems that evolve with respect to a single independent variable. It focuses on the modeling of dynamic systems through first-order equations, higher-order equations, and systems of equations, using both analytical and numerical methods. It is fundamental for the understanding of mechanics, electrical circuits, and population models.

4. GOALS

- Solve first-order and higher-order ordinary differential equations using analytical methods.
- Apply the Laplace transform to solve initial value problems involving discontinuous functions.
- Analyze the stability of linear and nonlinear systems of differential equations.
- Implement numerical methods for approximating solutions in complex models.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: First-Order Ordinary Differential Equations (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Modeling with first-order equations: growth, mixtures, and cooling	• Classify first-order ODEs and select an appropriate solution method (separation of variables, linear equation, exact equation, or integrating factor) [Familiarity] • Formulate and solve first-order models for growth and decay, mixing problems, Newton's law of cooling, and Torricelli's law [Usage] • Analyze autonomous first-order ODEs qualitatively using direction fields, and identify equilibrium points and their stability [Assessment]
Readings : [**Boyce2017**], [**Zill2018**]	

Unit 2: Higher-Order Differential Equations and Systems (40 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Definition and properties of the Laplace Transform; inverse transforms and translation theorems • Derivatives of a transform and convolution integrals	• State the Picard-Lindelöf existence and uniqueness theorem and interpret its implications for the continuation of solutions of a higher-order ODE [Familiarity] • Solve linear systems of ODEs using eigenvalue methods and sketch their phase portraits [Usage] • Apply the Laplace transform to solve ODEs with piecewise and impulsive forcing functions [Assessment]
Readings : [**Boyce2017**], [**Zill2018**]	

Unit 3: Numerical Methods for Differential Equations (36 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Phase portraits of 2D systems: equilibria, stability, and limit cycles • Stability of critical points and nonlinear systems	• Compare explicit and implicit ODE solvers and explain their relative stability for stiff problems [Familiarity] • Implement a Runge-Kutta method with adaptive step control and apply it to a system of ODEs [Usage] • Analyze the stability and convergence of a finite difference scheme for a parabolic PDE using the Lax equivalence theorem [Assessment]
Readings : [**Boyce2017**], [**Zill2018**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Numerical Analysis II
1.2 Code	:	MA325
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA221. Numerical Analysis I. (5^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Numerical Methods for Ordinary Differential Equations (ODEs) (20 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Euler's method and its variants (forward, backward) • Runge-Kutta methods • Multistep methods (Adams-Bashforth, Adams-Moulton) • Stability and convergence analysis for ODE solvers • Systems of ODEs and higher-order ODEs	• Implement basic ODE solvers (Euler, Runge-Kutta) [Usage] • Analyze the stability and convergence properties of ODE numerical methods [Assessment] • Select an appropriate numerical method for a given ODE based on stability and accuracy requirements [Assessment] • Solve systems of ODEs and convert higher-order ODEs to first-order systems [Usage]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 2: Numerical Differentiation and Integration (20 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Finite difference approximations (forward, backward, central) • Richardson extrapolation • Newton-Cotes formulas (Trapezoidal rule, Simpson's rule) • Gaussian quadrature • Error analysis for numerical integration	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 3: Numerical Linear Algebra (20 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• LU decomposition and its applications • QR decomposition and least squares problems • Singular Value Decomposition (SVD) and its applications • Condition number and numerical stability of linear systems • Iterative methods for linear systems (Jacobi, Gauss-Seidel, Conjugate Gradient)	• Implement LU and QR decomposition algorithms for solving linear systems [Usage] • Explain the geometric and algebraic meaning of the Singular Value Decomposition [Familiarity] • Use SVD for applications such as matrix approximation and principal component analysis [Usage] • Analyze the stability of a linear system using its condition number [Assessment] • Select and implement an appropriate iterative method for solving large, sparse linear systems [Assessment]
Readings : [**Burden2016**], [**Chapra2015**]	

Unit 4: Interpolation and Function Approximation (20 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Polynomial interpolation (Lagrange, Newton forms) • Spline interpolation (cubic splines) • Least squares approximation • Runge's phenomenon and Chebyshev nodes • Error analysis for interpolation	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Burden2016**], [**Chapra2015**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Statistics and Probabilities
1.2 Code	:	MA351
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	• MA251. Introduction to Statistics. (5^{th} Sem) • CS112. Objects-oriented programming I. (2^{nd} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Probability and statistics provide the mathematical foundation for reasoning under uncertainty, analyzing data, and building predictive models – skills that are essential across computing, artificial intelligence, computational mathematics, and engineering alike. This course develops a rigorous understanding of probability theory and its use in describing random phenomena, together with the fundamentals of descriptive statistics, statistical inference (estimation, hypothesis testing, and Bayesian reasoning), and regression modeling, so that students can summarize and analyze data, quantify uncertainty, and draw sound quantitative conclusions in their own field of application.

4. GOALS

- Understand the fundamentals of probability theory and descriptive statistics.
- Apply probability distributions and statistical inference methods to analyze data and quantify uncertainty.
- Develop skills to model random phenomena and build predictive statistical models such as linear regression.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

6. TOPICS

Unit 1: Numerical Summaries: Location, Spread, and Shape (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Measures of location: mean, median, mode, trimmed mean, and weighted mean • Measures of spread: variance, standard deviation, IQR, MAD, and range • Measures of shape: skewness and kurtosis; quantiles and percentiles • Robust summary statistics: resistance to outliers and heavy-tailed distributions	• Calculate and interpret measures of location, spread, and shape for a univariate dataset [Usage] • Select appropriate summary statistics based on the distributional properties of the data [Usage] • Compare classical and robust summary statistics and justify the choice for a given dataset [Assessment]
Readings : [**Devore2016**]	

Unit 2: Sample Spaces and Events (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Random experiments, outcomes, and sample spaces: finite, countable, and uncountable • Set-theoretic operations on events: union, intersection, complement, and De Morgan's laws • Sigma-algebras and measurable spaces as the foundation for rigorous probability • Borel sets and the Borel sigma-algebra on the real line	• Define sample spaces, events, and sigma-algebras for a given random experiment [Familiarity] • Apply set-theoretic operations to compute probabilities of compound events [Usage] • Construct appropriate sigma-algebras for measurable spaces arising in statistical problems [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 3: Probability Axioms and Properties (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Kolmogorov's three axioms of probability and their consequences • Derived properties: complement rule, addition rule, monotonicity, and continuity of probability • Classical, frequentist, and subjective (Bayesian) interpretations of probability • Counting methods: permutations, combinations, and the multiplication principle	• State Kolmogorov's axioms and derive fundamental probability properties [Familiarity] • Calculate probabilities of events using the addition rule and complement rule [Usage] • Apply counting methods to compute probabilities in finite sample spaces [Usage]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 4: Conditional Probability and Independence (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Definition and interpretation of conditional probability P(A|B) and its properties • Bayes' theorem and the law of total probability: derivation and applications • Independence of events: pairwise and mutual independence • Conditional independence and its role in probabilistic graphical models	• Explain conditional probability and independence and their relationship [Familiarity] • Apply Bayes' theorem and the law of total probability to solve probability problems [Usage] • Analyze whether events are independent or conditionally independent in a given probability model [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 5: Discrete Random Variables and Distributions (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Probability mass functions (PMF) and cumulative distribution functions (CDF) for discrete random variables • Key discrete distributions: Bernoulli, Binomial, Geometric, Negative Binomial, Poisson, Hypergeometric • Properties of discrete distributions: mean, variance, skewness, and kurtosis • Poisson approximation to the Binomial and limiting distribution arguments	• Identify and describe the key discrete probability distributions and their parameters [Familiarity] • Calculate probabilities, means, and variances using PMFs of standard discrete distributions [Usage] • Select an appropriate discrete distribution to model a given random phenomenon and justify the choice [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 6: Continuous Random Variables and Distributions (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Probability density functions (PDF) and CDFs for continuous random variables • Key continuous distributions: Uniform, Normal, Exponential, Gamma, Beta, Chi-squared, t, F • The Normal distribution: properties, standard normal, and its central role in statistical inference • The exponential family of distributions: canonical form and sufficient statistics	• Describe the properties and parameters of key continuous probability distributions [Familiarity] • Calculate probabilities and quantiles for standard continuous distributions [Usage] • Identify distributions belonging to the exponential family and derive their sufficient statistics [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 7: Joint, Marginal, and Conditional Distributions (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Joint PMFs and PDFs for two or more random variables • Marginal and conditional distributions derived from joint distributions • Covariance, correlation, and their properties; independence implies zero covariance (converse not always true) • The bivariate normal distribution and its conditional distributions	• Derive marginal and conditional distributions from a given joint distribution [Usage] • Calculate covariance and correlation between two random variables [Usage] • Analyze the conditional structure of the bivariate normal distribution [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 8: Limit Theorems: LLN and CLT (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Weak law of large numbers: convergence in probability and its implications for estimation • Central limit theorem: statement, conditions, and role in justifying normal approximations • Applications of the CLT: normal approximation to the binomial and sampling distributions • Strong law of large numbers: almost sure convergence • The delta method: asymptotic distribution of smooth functions of estimators	• State the LLN and CLT and explain the conditions under which they hold [Familiarity] • Apply the CLT to derive normal approximations and sampling distributions [Usage] • Apply the delta method to derive asymptotic distributions of nonlinear estimators [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 9: Point Estimation (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Desirable properties: unbiasedness, consistency, efficiency, and sufficiency • Method of moments estimation: derivation and asymptotic properties • Maximum likelihood estimation: derivation, properties, and invariance principle • Cramer-Rao lower bound and the concept of Fisher information	• Describe the key properties of estimators and explain the trade-offs between them [Familiarity] • Derive method of moments and MLE estimators for standard distributions [Usage] • Evaluate the efficiency of an estimator using the Cramer-Rao lower bound [Assessment]
Readings : [**Devore2016**]	

Unit 10: Interval Estimation and Confidence Intervals (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Constructing confidence intervals using pivotal quantities and exact distributions • Correct interpretation of confidence intervals: coverage probability and common misconceptions • Large-sample confidence intervals based on asymptotic normality • Duality between confidence intervals and hypothesis tests	• Construct confidence intervals for means, proportions, and variances using exact and large-sample methods [Usage] • Interpret confidence intervals correctly and identify common misinterpretations [Usage] • Explain the duality between confidence intervals and hypothesis tests and use it to draw inferences [Assessment]
Readings : [**Devore2016**]	

Unit 11: Hypothesis Testing Foundations (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Neyman-Pearson framework: null/alternative hypotheses, rejection regions, Type I and II errors • p-values: definition, correct interpretation, and common misuse; power and sample size • Common parametric tests: z-test, t-test, F-test, chi-squared test • Neyman-Pearson lemma and uniformly most powerful tests	• Explain the logic of hypothesis testing, Type I and II errors, and the meaning of a p-value [Familiarity] • Apply standard parametric tests and interpret the results in context [Usage] • Derive the most powerful test for a simple hypothesis using the Neyman-Pearson lemma [Assessment]
Readings : [**Devore2016**]	

Unit 12: Bayesian Inference (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Bayesian framework: prior, likelihood, posterior, and the role of Bayes' theorem • Prior distributions: informative, non-informative, and conjugate priors • Posterior inference: point estimates, credible intervals, and posterior predictive distributions • Bayesian model comparison: Bayes factors and posterior model probabilities	• Explain the Bayesian approach to inference and contrast it with the frequentist approach [Familiarity] • Derive the posterior distribution for standard conjugate models [Usage] • Apply Bayes factors to compare statistical models and interpret the evidence [Assessment]
Readings : [**Ross2014**], [**Devore2016**]	

Unit 13: Simple Linear Regression (4 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Ordinary least squares estimation: derivation of coefficients and their properties • Inference for regression coefficients: t-tests, confidence intervals, and p-values • Coefficient of determination R-squared and analysis of variance decomposition • Gauss-Markov theorem: BLUE property of OLS under classical assumptions	• Fit a simple linear regression model and interpret the estimated coefficients [Usage] • Conduct hypothesis tests and construct confidence intervals for regression parameters [Usage] • Verify the Gauss-Markov assumptions and assess their consequences when violated [Assessment]
Readings : [**Devore2016**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Functional analysis I
1.2 Code	:	MA371
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• MA222. Real Analysis I. (5^{th} Sem) • ID201. Technical and professional English III. (5^{th} Sem) • MA122. Linear Algebra II. (4^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Functional analysis is a central pillar of modern mathematics, providing the framework to solve problems in differential equations, quantum mechanics, and numerical analysis. This course bridges the gap between linear algebra and infinite-dimensional spaces. By introducing Measure Theory as a rigorous foundation for integration, the course enables the study of Banach and Hilbert spaces, linear operators, and the fundamental theorems that govern functional systems.

4. GOALS

- Establish a rigorous understanding of Lebesgue measure and integration theory.
- Master the topological and algebraic structures of Banach and Hilbert spaces.
- Apply the fundamental theorems of functional analysis (Hahn-Banach, Open Mapping, Closed Graph).
- Analyze the properties of linear operators and their spectral theory in infinite dimensions.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Measure and Integration (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• σ-algebras and measurable spaces • Lebesgue measure on $\mathbb{R}^n$ • Measurable functions and Lebesgue integration • Convergence theorems (Monotone, Dominated, Fatou's Lemma) • L^p spaces: completeness and duality	• Construct the Lebesgue measure and identify measurable sets [Assessment] • Evaluate integrals using convergence theorems [Usage] • Prove the completeness of L^p spaces [Familiarity]
Readings : [**Folland1999**], [**Royden2010**]	

Unit 2: Normed and Banach Spaces (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Normed vector spaces and completeness • Schauder bases and infinite dimensionality • Linear operators and functionals • Dual spaces and the Hahn-Banach theorem • Category theorems: Baire, Open Mapping, and Closed Graph	• Verify the completeness of various functional spaces [Assessment] • Apply the Hahn-Banach theorem to extend linear functionals [Usage] • Utilize the Open Mapping theorem to determine operator boundedness [Assessment]
Readings : [**Kreyszig1989**], [**Brezis2011**]	

Unit 3: Hilbert Spaces (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Inner product spaces and Hilbert structure • Orthogonality, projection theorem, and Riesz representation • Orthonormal sets and Fourier series in Hilbert spaces • Self-adjoint, unitary, and normal operators	• Construct orthogonal projections in Hilbert spaces [Usage] • Represent linear functionals using the Riesz theorem [Assessment] • Analyze the geometry of infinite-dimensional inner product spaces [Familiarity]
Readings : [**Kreyszig1989**], [**Conway1990**]	

Unit 4: Operator Theory and Spectral Theory (16 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Compact operators on Banach and Hilbert spaces • The Fredholm alternative • Spectral theory of bounded linear operators • Spectral theorem for compact self-adjoint operators	• Identify compact operators and their properties [Assessment] • Determine the spectrum of linear operators [Usage] • Apply the spectral theorem to decompose operators [Assessment]
Readings : [**Brezis2011**], [**Yosida1980**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Technical and professional English IV
1.2 Code	:	ID202
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	ID201. Technical and professional English III. (5^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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

4. 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.

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: Do and don't! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 2: Going places! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 3: Scared to death! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 4: Things that changed the world! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 5: Dreams and reality! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 6: Making a living! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

Unit 7: All you need is love! (0 hours)	
Competences Expected: ABET-4,ABET-6	
Topics	**Learning Outcomes**
• Past Perfect and Past Simple Â Â Â Â Â Â • Report Expressions Â Â Â Â Â Â • Expressions of words in different contexts Â Â Â Â Â Â • Short and formal farewells Â Â Â Â Â Â • 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 : [**Soars022S**], [**Cambridge06**], [**MacGrew99**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Artificial Intelligence
1.2 Code	:	CS261
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	MA251. Introduction to Statistics. (5^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Artificial Intelligence (AI) is a fundamental field of computer science that seeks to create systems capable of performing tasks that typically require human intelligence. This course provides an introduction to the techniques used to build such systems, including search algorithms, probabilistic reasoning, and machine learning. Students will learn to model complex problems and implement intelligent agents, while considering the social and ethical implications of these technologies.

4. GOALS

- Understand the fundamental concepts and history of Artificial Intelligence.
- Implement and evaluate search algorithms for problem-solving.
- Master the basics of machine learning, including supervised and unsupervised learning.
- Apply probabilistic reasoning to handle uncertainty in intelligent systems.
- Analyze the ethical challenges and social impact of AI.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Introduction to AI: Concepts and Intelligent Behavior (4 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Overview of AI problems, Examples of successful recent AI applications • Definitions of agents with examples (e.g., reactive, deliberative) • What is intelligent behavior?: – The Turing test and its flaws – Multimodal input and output – Simulation of intelligent behavior – Rational versus non-rational reasoning • Overview of AI Applications, growth, and impact (economic, societal, ethics) • Philosophical issues • History of AI	• Describe the Turing test and the "Chinese Room" thought experiment [Explain] • Differentiate between optimal reasoning/behavior and human-like reasoning/behavior [Evaluate] • Differentiate the terms: AI, machine learning, and deep learning [Explain]
Readings : [**Russell2020**], [**Luger2008**]	

Unit 2: Problem Characteristics and Agent Nature (4 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Problem characteristics: – Fully versus partially observable – Single versus multi-agent – Deterministic versus stochastic – Static versus dynamic – Discrete versus continuous • Nature of agents: – Autonomous, semi-autonomous, mixed-initiative autonomy – Reflexive, goal-based, and utility-based – Decision making under uncertainty and with incomplete information – The importance of perception and environmental interactions – Learning-based agents – Embodied agents: ∗ sensors, dynamics, effectors • Practice identifying problem characteristics in example environments • Additional depth on nature of agents with examples • Additional depth on AI Applications, Growth, and Impact (economic, societal, ethics, security)	• Enumerate the characteristics of a specific problem [Apply]
Readings : [**Russell2020**], [**Luger2008**]	

Unit 3: State and Uninformed Search (5 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• State space representation of a problem: – Specifying states, goals, and operators – Factoring states into representations (hypothesis spaces) – Problem solving by graph search: ∗ e.g., Graphs as a space, and tree traversals as exploration of that space ∗ Dynamic construction of the graph (not given upfront) • Uninformed graph search for problem solving: – Breadth-first search – Depth-first search: ∗ With iterative deepening – Uniform cost search • Space and time complexities of graph search algorithms • Bidirectional search • Understanding the search space: – Constructing search trees – Dynamic search spaces – Combinatorial explosion of search space – Search space topology (e.g., ridges, saddle points, local minima)	• Design the state space representation for a puzzle (e.g., N-queens or 3-jug problem) [Apply] • Select and implement an appropriate uninformed search algorithm for a problem (e.g., tic-tac-toe), and characterize its time and space complexities [Apply]
Readings : [**Russell2020**], [**Luger2008**]	

Unit 4: Heuristic Search and CSP (6 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Heuristic graph search for problem solving: – Heuristic construction and admissibility – Hill-climbing – Local minima and the search landscape: ∗ Local vs global solutions – Greedy best-first search – A* search • Beam search • Implementation of A* search • Constraint satisfaction • Variations on A* (IDA*, SMA*, RBFS)	• Select and implement an appropriate informed search algorithm for a problem after designing a helpful heuristic function (e.g., a robot navigating a 2D gridworld) [Apply] • Evaluate whether a heuristic for a given problem is admissible/can guarantee an optimal solution [Evaluate] • Design and implement A*/beam search to solve a problem, and compare it against other search algorithms in terms of the solution cost, number of nodes expanded, etc [Apply] • Compare and contrast various heuristic searches vis-a-vis applicability to a given problem [Evaluate] • Model a logic or Sudoku puzzle as a constraint satisfaction problem, solve it with backtrack search, and determine how much arc consistency can reduce the search space [Apply]
Readings : [**Russell2020**], [**Luger2008**]	

Unit 5: Adversarial Search and Stochastic Optimization (5 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Two-player adversarial games: – Minimax search – Alpha-beta pruning: ∗ Ply cutoff • Local search • Tabu search • Two-player adversarial games: – The horizon effect – Opening playbooks/endgame solutions – What it means to ”solve” a game (e.g., checkers) • Implementation of minimax search, beam search • Expectimax search (MDP-solving) and chance nodes • Stochastic search: – Simulated annealing – Genetic algorithms – Monte-Carlo tree search	• Apply minimax search in a two-player adversarial game (e.g., connect four), using heuristic evaluation at a particular depth to compute the scores to back up [Apply] • Apply minimax search with alpha-beta pruning to prune search space in a two-player adversarial game (e.g., connect four) [Apply] • Compare and contrast genetic algorithms with classic search techniques, explaining when it is most appropriate to use a genetic algorithm to learn a model versus other forms of optimization (e.g., gradient descent) [Evaluate] • Design and implement a genetic algorithm solution to a problem [Apply] • Design and implement a simulated annealing schedule to avoid local minima in a problem [Apply]
Readings : [**Russell2020**], [**Luger2008**]	

Unit 6: Machine Learning Fundamentals (7 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Definition and examples of a broad variety of machine learning tasks: – Supervised learning: ∗ Classification ∗ Regression – Reinforcement learning – Unsupervised learning: ∗ Clustering • Fundamental ideas: – No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences. – Sources of error and undecidability in machine learning • A simple statistical-based supervised learning such as linear regression or decision trees: – Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly • The overfitting problem/controlling solution complexity (regularization, pruning - intuition only): – The bias (underfitting) - variance (overfitting) tradeoff	• Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain [Explain] • Differentiate the terms of AI, machine learning, and deep learning [Evaluate] • Frame an application as a classification problem, including the available input features and output to be predicted (e.g., identifying alphabetic characters from pixel grid input) [Apply] • Identify overfitting in the context of a problem and learning curves and describe solutions to overfitting [Evaluate] • Explain how machine learning works as an optimization/search process [Explain]
Readings : [**Russell2020**], [**Alpaydin2020**]	

Unit 7: Machine Learning Data, Representations and Evaluation (7 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Working with Data: – Data preprocessing: ∗ Importance and pitfalls of preprocessing choices – Handling missing values (imputing, flag-as-missing): ∗ Implications of imputing vs flag-as-missing – Encoding categorical variables, encoding real-valued data – Normalization/standardization – Emphasis on real data, not textbook examples • Representations: – Hypothesis spaces and complexity – Simple basis feature expansion, such as squaring univariate features – Learned feature representations • Machine learning evaluation: – Separation of train, validation, and test sets – Performance metrics for classifiers – Estimation of test performance on held-out data – Tuning the parameters of a machine learning model with a validation set – Importance of understanding what a model is doing, where its pitfalls/shortcomings are, and the implications of its decisions	• Apply two or more simple statistical learning algorithms to a classification task and measure the classifiers' accuracy [Apply] • Implement a statistical learning algorithm and the corresponding optimization process to train the classifier and obtain a prediction on new data [Apply] • Explain proper ML evaluation procedures, including the differences between training and testing performance, and what can go wrong with the evaluation process leading to inaccurate reporting of ML performance [Explain] • Compare two machine learning algorithms on a dataset, implementing the data preprocessing and evaluation methodology (e.g., metrics and handling of train/test splits) from scratch [Apply]
Readings : [**Russell2020**], [**Alpaydin2020**]	

Unit 8: Neural Networks and Ethics (6 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Basic neural networks: – Fundamentals of understanding how neural networks work and their training process, without details of the calculations – Basic introduction to generative neural networks (e.g., large language models) • Ethics for Machine Learning: – Focus on real data, real scenarios, and case studies – Dataset/algorithmic/evaluation bias and unintended consequences	• Describe the neural network training process and resulting learned representations [Explain]
Readings : [**Russell2020**], [**Alpaydin2020**]	

Unit 9: Machine Learning Formulation and Evaluation (3 hours)	
Competences Expected: ABET-2	
Topics	**Learning Outcomes**
• Formulation of simple machine learning as an optimization problem, such as least squares linear regression or logistic regression: – Objective function – Gradient descent – Regularization to avoid overfitting (mathematical formulation) • Ensembles of models: – Simple weighted majority combination • Performance evaluation: – Other metrics for classification (e.g., error, precision, recall) – Performance metrics for regressors – Confusion matrix – Cross-validation: ∗ Parameter tuning (grid/random search, via cross-validation)	• Compare and contrast several learning techniques (e.g., decision trees, logistic regression, naive Bayes, neural networks, and belief networks), providing examples of when each strategy is superior [Compare] • Evaluate the performance of a simple learning system on a real-world dataset [Evaluate] • Characterize the state of the art in learning theory, including its achievements and shortcomings [Explain] • Explain the problem of overfitting, along with techniques for detecting and managing the problem [Explain] • Explain the triple tradeoff among the size of a hypothesis space, the size of the training set, and performance accuracy [Explain]
Readings : [**Goodfellow2016**]	

Unit 10: Deep Learning, RL and Ethics (3 hours)	
Competences Expected: ABET-2	
Topics	**Learning Outcomes**
• Deep learning: – Deep feed-forward networks (intuition only, no mathematics) – Convolutional neural networks (intuition only, no mathematics) – Visualization of learned feature representations from deep nets – Other architectures (generative NN, recurrent NN, transformers, etc.) • Overview of reinforcement learning methods • Two or more applications of machine learning algorithms: – E.g., medicine and health, economics, vision, natural language, robotics, game play • Ethics for Machine Learning: – Continued focus on real data, real scenarios, and case studies – Privacy – Fairness – Intellectual property – Explainability	• Visualize the training progress of a neural network through learning curves in a well-established toolkit (e.g., TensorBoard) and visualize the learned features of the network [Apply] • Given a real-world application of machine learning, describe ethical issues regarding the choices of data, preprocessing steps, algorithm selection, and visualization/presentation of results [Apply]
Readings : [**Goodfellow2016**]	

Unit 11: Knowledge Representation and Probabilistic Reasoning Foundations (6 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Types of representations: – Symbolic, logical: ∗ Creating a representation from a natural language problem statement – Learned subsymbolic representations – Graphical models (e.g., naive Bayes, Bayesian network) • Review of probabilistic reasoning, Bayes theorem • Bayesian reasoning: – Bayesian inference • Random variables and probability distributions: – Axioms of probability – Probabilistic inference – Bayes' Rule (derivation) – Bayesian inference (more complex examples)	• Given a natural language problem statement, encode it as a symbolic or logical representation [Apply] • Compute a probabilistic inference in a real-world problem using Bayes' theorem to determine the probability of a hypothesis given evidence [Apply] • Apply Bayes' rule to determine the probability of a hypothesis given evidence [Apply] • Compute the probability of outcomes and test whether outcomes are independent [Apply]
Readings : [**Russell2020**]	

Unit 12: Independence, Markov Models and Decision Making (6 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Independence • Conditional Independence • Markov chains and Markov models • Utility and decision making	• Explain how we can make decisions under uncertainty, using concepts such as Bayes theorem and utility [Explain]
Readings : [**Russell2020**]	

Unit 13: AI Applications and Societal Impact (4 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• At least one application of AI to a specific problem and field, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose at least one for the CS Core): – Formulating and evaluating a specific application as an AI problem: ∗ How to deal with underspecified or ill-posed problems – Data availability/scarcity and cleanliness: ∗ Basic data cleaning and preprocessing ∗ Data set bias – Algorithmic bias – Evaluation bias – Assessment of societal implications of the application • Deployed deep generative models: – High-level overview of deep image generative models (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion, etc.), their uses, and their shortcomings/pitfalls. – High-level overview of large language models (e.g., as of 2023, ChatGPT, Bard, etc.), their uses, and their shortcomings/pitfalls. • Overview of societal impact of AI: – Ethics – Fairness – Trust/explainability – Privacy and usage of training data – Human autonomy and oversight/regulations/legal requirements – Sustainability	• Given a real-world application domain and problem, formulate an AI solution to it, identifying proper data/input, preprocessing, representations, AI techniques, and evaluation metrics/methodology [Apply]
Readings : [**Russell2020**], [**Alpaydin2020**]	

Unit 14: Additional AI Applications and Societal Impact (4 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• One or more additional applications of AI to a broad set of problems and diverse fields, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose a different area from that chosen for the CS Core): – Formulating and evaluating a specific application as an AI problem: ∗ How to deal with underspecified or ill-posed problems – Data availability/scarcity and cleanliness: ∗ Basic data cleaning and preprocessing ∗ Data set bias – Algorithmic bias – Evaluation bias – Assessment of societal implications of the application • Additional depth on deployed deep generative models: – Introduction to how deep image generative models work, (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion) including discussion of attention – Introduction to how large language models work, (e.g., as of 2023, ChatGPT, Bard) including discussion of attention – Idea of foundational models, how to use them, and the benefits/issues with training them from big data • Analysis and discussion of the societal impact of AI: – Ethics – Fairness – Trust/explainability – Privacy and usage of training data – Human autonomy and oversight/regulations/legal requirements – Sustainability	• Analyze the societal impact of one or more specific real-world AI applications, identifying issues regarding ethics, fairness, bias, trust, and explainability [Evaluate] • Describe some of the failure modes of current deep generative models for language or images, and how this could affect their use in an application [Describe]
Readings : [**Russell2020**], [**Alpaydin2020**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Parallel and Distributed Computing
1.2 Code	:	CS3P1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	CS210. Algorithms and Data Structures. (4^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

With the end of frequency scaling in single-core processors, parallel and distributed computing has become essential for high-performance software development. This course introduces students to the programming models, communication patterns, and coordination mechanisms required to exploit multi-core systems and distributed clusters. Students will learn to design scalable algorithms and evaluate their performance under different computational constraints.

4. GOALS

- Design and implement parallel programs using shared and distributed memory.
- Master communication and synchronization protocols in distributed systems.
- Evaluate performance and scalability through formal metrics.
- Apply parallel algorithms to solve complex computational problems.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Programs: Declarative Parallelism (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Parallelism – Declarative parallelism: Determining which actions can, or must not, be performed in parallel, at the level of instructions, functions, closures, compound actions, sessions, tasks, and services is the core idea underlying PDC algorithms; failing to do so is the primary source of errors. See also: Algorithms – Defining order: for example, using "happens-before" relationships or series-parallel directed acyclic graphs (DAGs) representing programs. – Independence: determining when order does not matter, in terms of commutativity, dependencies, preconditions. – Guaranteeing order between otherwise parallel actions when necessary, including locks, safe publication; and enforcing communication - sending a message happens before receiving it; and relaxing it when not necessary.	• Graphically represent (as a Directed Acyclic Graph - DAG) how to parallelize a compound numeric expression; for example, $a = (b + c) * (d + e)$ [Design] • Explain why consistency and fault tolerance concepts do not arise in purely sequential programs [Explain]
Readings : [**Pacheco2021**]	

Unit 2: Programs: Starting Activities (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Starting activities – Options allowing actions to be performed (eventually) at locations range from hardwired to configuration scripts; also establishing communication and resource management; these are expressed differently in languages and contexts, typically relying on automated provisioning and management by platforms. See also: Resource Management – Procedural: Allowing multiple actions to start at a given program point; for example, starting new threads, possibly delimiting their scope or organizing them into hierarchical groups. – Reactive: Enabling upon event occurrence by installing an event handler, with less control over when actions start or finish, and may apply even on single processors. – Dependent: Enabling upon completion of others; for example, sequencing sets of parallel actions. See also: Coordination. – Granularity: The execution cost of action bodies should outweigh the overhead of organizing them.	• Write a service that creates a thread (or other form of procedural activation) to return a requested web page to each new client [Write]
Readings : [**Pacheco2021**]	

Unit 3: Programs: Execution Properties (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Execution Properties – Non-deterministic execution of actions without order. – Consistency: Ensuring agreement among parties on values and predicates when necessary to avoid races, maintain safety and atomicity, or reach consensus. – Fault tolerance: Handling failures in parts or communication, including (Byzantine) misbehavior due to unreliable parties and protocols, when necessary to maintain progress or availability. See also: System Reliability. – Trade-offs are a focus of evaluation. See also: Evaluation.	• Write a function that counts events, such as network packet receptions, efficiently [Write]
Readings : [**VanSteen2023**]	

Unit 4: Programs: Distribution (2 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Distribution – Defining locations as devices that execute actions, including hardware components, remote hosts; may also include external uncontrolled devices, hosts, and users. See also: Interfacing and Communication – A device may time-share or emulate multiple parallel actions using fewer processors through scheduling and virtualization. See also: Scheduling – Naming or identifying locations (e.g., device IDs) and actions as parts (e.g., thread IDs). – Activities across locations may communicate through media. See also: Communication	• Write a filter/map/reduce program in multiple styles [Write]
Readings : [**VanSteen2023**]	

Unit 5: Programs: Implementation Mappings (2 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• One or more of the following mappings and mechanisms across layered systems: – Instruction-level and data parallelism in the CPU. See also: Functional Organization. – SIMD and heterogeneous data parallelism. See also: Heterogeneous Architectures. – Scheduled concurrency on multi-core, tasks, actors. See also: Scheduling. – Clusters, clouds; elastic provisioning. See also: Common Aspects: Platforms, APIs and Constraints, Common Aspects: Platform Languages and Web Programming Patterns. – Networked distributed systems. See also: Networked Applications. – Emerging technologies such as quantum computing and molecular computing.	• Explain the trade-offs between different mapping strategies in terms of performance, cost, and implementation complexity [Explain]
Readings : [**Pacheco2021**], [**Hwu2022**]	

Unit 6: GPU Programming (4 hours)

Competences Expected: ABET-2

Topics	**Learning Outcomes**
• General-Purpose GPU Computing (GPGPU) – GPU architecture overview: streaming multiprocessors (SMs), CUDA cores, and warp execution model. – SIMT (Single Instruction, Multiple Threads) execution model and its differences from CPU SIMD. – Host-device interaction: data transfer between CPU (host) and GPU (device) memory via PCIe. – Use cases: scientific computing, machine learning, image processing, and large-scale simulation. • CUDA Programming Model – CUDA execution hierarchy: grids, thread blocks, and individual threads; mapping to GPU hardware. – Kernel definition and launch syntax; configuration parameters (grid and block dimensions). – Thread indexing: threadIdx, blockIdx, blockDim, gridDim; computing global indices in 1D, 2D, and 3D. – Warp divergence: performance implications of branching within a warp. – Intra-block synchronization primitive: __syncthreads(). • CUDA Memory Hierarchy – Global memory: large, high-latency, accessible by all threads; coalesced access patterns for performance. – Shared memory: low-latency on-chip memory shared within a thread block; bank conflicts. – Registers and local memory: per-thread private storage. – Constant and texture memory: read-only caches optimized for specific access patterns. – Unified Memory: simplified programming model with automatic host-device data migration.	• Write a CUDA kernel that performs a data-parallel operation (e.g., vector addition, matrix multiplication) and correctly configure grid and block dimensions [Write] • Analyze the impact of global memory access patterns (coalesced vs. strided) and shared memory usage on GPU kernel performance [Analyze] • Design a parallel algorithm using the CUDA programming model, including thread hierarchy, memory allocation strategy, and host-device data transfers [Design]

Readings : [**Kirk2016**], [**CUDADocs**]

Unit 7: Communication (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Media – Varieties: channels (message passing or I/O), shared memory, heterogeneous, data stores – Reliance on the availability and nature of underlying hardware, connectivity, and protocols; language support, emulation. See also: Interfacing and Communication • Channels – Explicit (usually named) party-to-party communication media – APIs: Sockets, architectural, language-based, and toolkit constructs, such as Message Passing Interface (MPI), and layered constructs such as Remote Procedure Call (RPC). See also: Fundamentals of Networks and Communications – I/O channel APIs • Memory – Shared memory architectures in which parties directly communicate only with memory at given addresses, with extensions to heterogeneous memory supporting multiple memory stores with explicit data transfer across them; for example, GPU local and shared memory, Direct Memory Access (DMA) – Memory hierarchies: Multiple layers of sharing domains, scopes, and caches; locality: latency, false-sharing – Consistency properties: Bitwise atomicity limits, coherence, local ordering • Data Stores – Cooperatively maintained data structures implementing maps and related ADTs – Varieties: Owned, shared, sharded, replicated, immutable, versioned	• Explain the similarities and differences among: (1) Party A sends a message on channel X with contents 1 received by party B (2) A sets shared variable X to 1, read by B (3) A sets "X=1" in a distributed shared map accessed by B [Explain] • Write a program that distributes different segments of a data set to multiple workers, and collects results (for the simplest example, summing segments of an array) [Write] • Write a parallel program that requests data from multiple sites and summarizes them using some form of reduction [Write] • Compare the performance of buffered versus unbuffered versions of a producer-consumer program [Compare]
Readings : [**VanSteen2023**], [**Kleppmann2017**]	

Unit 8: Communication: Properties and Extensions (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• One or more of the following properties and extensions – Topologies: Unicast, Multicast, Mailboxes, Switches; Routing via hardware and software interconnection networks – Media concurrency properties: Ordering, consistency, idempotency, overlapping communication with computation – Media performance: Latency, bandwidth (throughput) contention (congestion), responsiveness (liveness), reliability (error and drop rates), protocol-based progress (acks, timeouts, mediation) – Media security properties: integrity, privacy, authentication, authorization. See also: Information Flow and Non-Interference, Injection and Input Validation, Memory Safety and Types, Malware Analysis and Advanced Security – Data formats: Marshaling, validation, encryption, compression – Channel policies: Endpoints, sessions, buffering, saturation response (waiting vs dropping), rate control – Multiplexing and demultiplexing many relatively slow I/O devices or parties; completion-based and scheduler-based techniques; async-await, select and polling APIs – Formalization and analysis of channel communication; for example, CSP – Applications of queuing theory to model and predict performance.	• Determine whether a given communication scheme provides sufficient security properties for a given usage [Determine] • Give an example of a scenario in which blocking message sends can deadlock [Create] • Describe at least one design technique for avoiding liveness failures in programs using multiple locks [Describe]
Readings : [**VanSteen2023**], [**Kleppmann2017**]	

Unit 9: Memory and Consistency (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Memory models, data consistency, and fault tolerance – Memory models: sequential and release/acquire consistency – Memory management; including reclamation of shared data; reference counts and alternatives – Bulk data placement and transfer; reducing message traffic and improving locality; overlapping data transfer and computation; impact of data layout such as array-of-structs vs struct-of-arrays – Emulating shared memory: distributed shared memory, Remote Direct Memory Access (RDMA) – Data store consistency: Atomicity, linearizability, transactionality, coherence, causal ordering, conflict resolution, eventual consistency, blockchains – Faults, partitioning, and partial failures; voting; protocols such as Paxos and Raft. – Design tradeoffs among consistency, availability, partition (fault) tolerance; impossibility of meeting all at once – Security and trust: Byzantine failures, proof of work and alternatives	• Give an example of an ordering of accesses among concurrent activities (e.g., program with a data race) that is not sequentially consistent [Create] • Write a program that illustrates memory-access or message reordering [Write] • Describe the relative merits of optimistic versus conservative concurrency control under different rates of contention among updates [Describe] • Give an example of a scenario in which an attempted optimistic update may never complete [Create] • Modify a concurrent system to use a more scalable, reliable, or available data store [Create] • Using an existing platform supporting replicated data stores, write a program that maintains a key-value mapping even when one or more hosts fail [Write]
Readings : [**VanSteen2023**], [**Kleppmann2017**]	

Unit 10: Coordination (7 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Dependencies – Initiation or progress of one activity may be dependent on other activities, so as to avoid race conditions, ensure termination, or meet other requirements – Ensuring progress by avoiding dependency cycles, using monotonic conditions, removing inessential dependencies • Control constructs and design patterns – Completion-based: Barriers, joins, including termination control – Data-enabled: Queues, producer-consumer designs – Condition-based: Polling, retrying, backoffs, helping, suspension, signaling, timeouts – Reactive: Enabling and triggering continuations	• Show how to ensure that a program correctly terminates when all of a set of concurrent tasks have completed [Design] • Write a function that efficiently counts events such as sensor inputs or networking packet receptions [Write] • Write a filter/map/reduce program in multiple styles [Write] • Write a program in which the termination of one set of parallel actions is followed by another [Write] • Write a service that creates a thread (or other procedural form of activation) to return a requested web page to each new client [Write]
Readings : [**Herlihy2020**], [**VanSteen2023**]	

Unit 11: Coordination: Synchronization and Atomicity (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Atomicity – Atomic instructions, enforced local access orderings – Locks and mutual exclusion; lock granularity – Using locks in a specific language; maintaining liveness without introducing races – Deadlock avoidance: Ordering, coarsening, randomized retries; backoffs, encapsulation via lock managers – Common errors: Failing to lock or unlock when necessary, holding locks while invoking unknown operations – Avoiding locks: replication, read-only, ownership, and non-blocking constructions	• Show how to avoid or repair a race error in a given program [Analyze]
Readings : [**Herlihy2020**]	

Unit 12: Coordination: Advanced Properties (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• One or more of the following properties and extensions – Progress properties including lock-free, wait-free, fairness, priority scheduling, interactions with consistency, reliability – Performance with respect to contention, granularity, convoying, scaling – Non-blocking data structures and algorithms – Ownership and resource control – Lock variants and alternatives: sequence locks, read-write locks; Read-Copy-Update (RCU), reentrancy; tickets; controlling spinning versus blocking – Transaction-based control: Optimistic and conservative – Distributed locking: reliability – Alternatives to barriers: Clocks; counters, virtual clocks; dataflow and continuations; futures and RPC; consensus-based, gathering results with reducers and collectors – Speculation, selection, cancellation; observability and security consequences – Resource control using semaphores and condition variables – Control flow: Scheduling computations, series-parallel loops with (possibly elected) leaders, pipelines and streams, nested parallelism – Exceptions and failures. Handlers, detection, timeouts, fault tolerance, voting	• Write a program that speculatively searches for a solution by multiple activities, terminating others when one is found [Write] • Write a program in which a numerical exception (such as divide by zero) in one activity causes termination of others [Write] • Write a program for multiple parties to agree upon the current time of day; discuss its limitations compared to protocols such as network transfer protocol (NTP) [Write]
Readings : [**Herlihy2020**]	

Unit 13: Evaluation (7 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Safety and liveness requirements in terms of temporal logic constructs to express "always" and "eventually" See also: Parallel and Distributed Computing • Identifying, testing for, and repairing violations, including common forms of errors such as failure to ensure necessary ordering (race errors), atomicity (including check-then-act errors), and termination (livelock) • Performance requirements metrics for throughput, responsiveness, latency, availability, energy consumption, scalability, resource usage, communication costs, waiting and rate control, fairness; service level agreements. See also: Latency, Cache and Memory Hierarchy, Virtualization and Isolation • Performance impact of design and implementation choices, including granularity, overhead, consensus costs, and energy consumption. See also: Sustainable Design and Pervasive Computing, Environmental Footprint of Computing Systems, Systemic Effects and Social Context • Estimating scalability limitations, for example using Amdahl's Law or Universal Scalability Law. See also: Events, Tools and Experimentation, Performance Metrics and Benchmarks, Analytical Performance Models	• Revise a specification to enable parallelism and distribution without violating other essential properties or features [Redesign] • Explain how concurrent notions of safety and liveness extend their sequential counterparts [Explain] • Specify a set of invariants that must hold at each bulk-parallel step of a computation [Analyze] • Write a test program that can reveal a data race error; for example, missing an update when two activities both try to increment a variable [Write] • In a given context, explain the extent to which introducing parallelism in an otherwise sequential program would be expected to improve throughput and/or reduce latency, and how it may impact energy efficiency [Explain] • Show how scaling and efficiency change for sample problems without and with the assumption of problem size changing with the number of processors; further explain whether and how scalability would change under relaxations of sequential dependencies [Design]
Readings : [**Sterling2024**], [**Pacheco2021**]	

Unit 14: Evaluation: Formal Methods (3 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Formal verification and analysis methods – Extensions to formal sequential requirements such as linearizability – Protocol, session, and transactional specifications – Use of tools such as Unified Modelling Language (UML), Temporal Logic of Actions (TLA), program logics – Security analysis: safety and liveness in the presence of hostile or buggy behaviors by other parties; required properties of communication mechanisms (for example lack of cross-layer leakage), input screening, rate limiting. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles – Static analysis applied to correctness, throughput, latency, resources, energy. See also: Sustainable Design and Pervasive Computing, Environmental Footprint of Computing Systems, Systemic Effects and Social Context – Directed Acyclic Graph (DAG) model analysis of algorithmic efficiency (work, span, critical paths)	• Specify and measure behavior when a service is requested by unexpectedly many clients [Analyze] • Identify and repair a performance problem due to sequential bottlenecks [Analyze] • Empirically compare throughput of two implementations of a common design (perhaps using an existing test harness framework) [Compare]
Readings : [**Sterling2024**], [**Pacheco2021**]	

Unit 15: Evaluation: Testing and Measurement (2 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Testing tools and measurement techniques – Testing and debugging; tools such as race detectors, fuzzers, lock dependency checkers, unit/stress/torture tests, visualizations, continuous integration, continuous deployment, and test generators – Measuring and comparing throughput, overhead, waiting, contention, communication, data movement, locality, resource usage, behavior in the presence of excessive numbers of events, clients, or threads. See also: Events, Tools and Experimentation, Performance Metrics and Benchmarks, Analytical Performance Models – Application domain specific analyses and evaluation techniques	• Identify and repair a performance problem due to communication or data latency [Analyze] • Identify and repair a performance problem due to resource management overhead [Analyze] • Identify and repair a reliability or availability problem [Analyze]
Readings : [**Sterling2024**], [**Pacheco2021**]	

Unit 16: Algorithms (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Expressing and implementing algorithms in given languages and frameworks, to initiate activities (for example threads), use shared memory constructs, and channel, socket, and/or remote procedure call APIs. See also: Parallel and Distributed Computing. – Data parallel examples including map/reduce. – Using channel, socket, and/or RPC APIs in a given language, with program control for sending (usually procedural) vs receiving. (usually reactive or RPC-based). – Using locks, barriers, and/or synchronizers to maintain liveness without introducing races.	• Implement a parallel/distributed component based on a known algorithm [Implement] • Write a data-parallel program that for example computes the average of an array of numbers [Write] • Write a producer-consumer program in which one component generates numbers, and another computes their average. Measure speedups when the numbers are small scalars versus large multi-precision values [Write]
Readings : [**Pacheco2021**], [**Herlihy2020**]	

Unit 17: Algorithms: Application Domains Survey (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Survey of common application domains across multicore, reactive, data parallel, cluster, cloud, open distributed systems, and frameworks (with reference to the following table). – Multicore: Typical Execution agents: Threads. Typical Communication mechanisms: Shared memory, Atomics, locks. Typical Algorithmic domains: Resource management, data processing. Typical Engineering goals: Throughput, latency, energy. – Reactive: Typical Execution agents: Handlers, threads. Typical Communication mechanisms: I/O Channels. Typical Algorithmic domains: Services, real-time. Typical Engineering goals: Latency. – Data parallel: Typical Execution agents: GPU, SIMD, accelerators, hybrid. Typical Communication mechanisms: Heterogeneous memory. Typical Algorithmic domains: Linear algebra, graphics, data analysis. Typical Engineering goals: Throughput, energy. – Cluster: Typical Execution agents: Managed hosts. Typical Communication mechanisms: Sockets, channels. Typical Algorithmic domains: Simulation, data analysis. Typical Engineering goals: Throughput. – Cloud: Typical Execution agents: Provisioned hosts. Typical Communication mechanisms: Service APIs. Typical Algorithmic domains: Web applications. Typical Engineering goals: Scalability. – Open distributed: Typical Execution agents: Autonomous hosts. Typical Communication mechanisms: Sockets, Data stores. Typical Algorithmic domains: Fault tolerant data stores and services. Typical Engineering goals: Reliability.	• Extend an event-driven sequential program by establishing a new activity in an event handler (for example a new thread in a GUI action handler) [Design] • Improve the performance of a sequential component by introducing parallelism and/or distribution [Create] • Choose among different parallel/distributed designs for components of a given system [Assess]
Readings : [**Pacheco2021**], [**Herlihy2020**]	

Unit 18: Algorithms: Algorithmic Domains (4 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• One of more of the following algorithmic domains. See also: Algorithmic Strategies: – Linear algebra: Vector and matrix operations, numerical precision/stability, applications in data analytics and machine learning. – Data processing: sorting, searching and retrieval, concurrent data structures. – Graphs, search, and combinatorics: Marking, edge-parallelization, bounding, speculation, network-based analytics. – Modeling and simulation: differential equations; randomization, N-body problems, genetic algorithms. – Computational logic: satisfiability (SAT), concurrent logic programming. – Graphics and computational geometry: Transforms, rendering, ray-tracing. – Resource management: Allocating, placing, recycling and scheduling processors, memory, channels, and hosts; exclusive vs shared resources; static, dynamic and elastic algorithms; Real-time constraints; Batching, prioritization, partitioning; decentralization via work-stealing and related techniques. – Services: Implementing web APIs, electronic currency, transaction systems, multiplayer games.	• Design, implement, analyze, and evaluate a component or application for X operating in a given context, where X is in one of the listed domains, for example, a genetic algorithm for factory floor design [Design] • Critique the design and implementation of an existing component or application, or one developed by classmates [Critique] • Compare the performance and energy efficiency of multiple implementations of a similar design, for example, multicore versus clustered versus GPU [Compare]
Readings : [**Pacheco2021**], [**Herlihy2020**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Partial Differential Equations
1.2 Code	:	MA213
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	• MA212. Ordinary Differential Equations. (6^{th} Sem) • MA211. Calculus IV. (5^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Partial Differential Equations course studies phenomena that depend on multiple variables, such as heat propagation, waves, and electrical potentials. The course introduces Fourier series and separation of variables methods to solve the classical equations of mathematical physics (Heat, Wave, and Laplace equations). It is a key component for advanced numerical simulation and continuum mechanics analysis.

4. GOALS

- Use Fourier series to represent periodic functions and solve boundary value problems.
- Solve linear partial differential equations using the separation of variables method.
- Classify second-order PDEs and understand their physical behavior.
- Apply finite difference numerical methods for solving PDEs.

5. COMPETENCES

No items for this table

6. TOPICS

Unit 1: Fourier Series and Sturm-Liouville Problems (24 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Orthogonal functions and sine and cosine Fourier series. • Complex Fourier series and Parseval's identity. • Eigenvalue problems and the Sturm-Liouville theorem. • Application of series to solving ODEs with boundary conditions.	• Implement basic ODE solvers (Euler, Runge-Kutta) [Usage] • Analyze the stability and convergence properties of ODE numerical methods [Assessment] • Select an appropriate numerical method for a given ODE based on stability and accuracy requirements [Assessment] • Solve systems of ODEs and convert higher-order ODEs to first-order systems [Usage]
Readings : [**Boyce2017**], [**Zill2018**]	

Unit 2: Diffusion Equations (Heat) (24 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Derivation of the heat equation in one dimension. • Separation of variables method for Dirichlet and Neumann conditions. • Heat problems in rods with insulated ends or radiation. • Heat equation in two and three dimensions (Cartesian coordinates).	• Implement root-finding algorithms (bisection, Newton, secant) and analyze their convergence [Usage] • Classify the order of convergence of iterative methods [Assessment] • Apply gradient descent to solve simple univariate optimization problems [Usage] • Design appropriate stopping criteria for iterative methods [Usage]
Readings : [**Boyce2017**], [**Zill2018**]	

Unit 3: Wave and Laplace Equations (24 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• The one-dimensional wave equation: Vibrating string. • D'Alembert's solution and traveling waves. • Laplace's equation in Cartesian and polar coordinates (Dirichlet Problem). • Steady-state applications and electrostatics.	• Construct interpolating polynomials using different methods [Usage] • Implement cubic spline interpolation [Usage] • Apply least squares method for data fitting [Usage] • Explain Runge's phenomenon and strategies to mitigate it [Familiarity] • Analyze the error of an interpolation scheme [Assessment]
Readings : [**Boyce2017**], [**Zill2018**]	

Unit 4: Numerical Methods for PDE and Transforms (24 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Introduction to the finite difference method for parabolic and elliptic PDEs. • Use of the Fourier Transform to solve PDEs in infinite domains. • Stability and convergence of basic numerical schemes. • Modeling of real-world problems using scientific computing software.	• Compute derivatives of functions using finite difference schemes [Usage] • Apply Richardson extrapolation to improve the accuracy of numerical approximations [Usage] • Implement numerical integration rules (Trapezoidal, Simpson, Gaussian) [Usage] • Compare the accuracy and efficiency of different numerical integration methods [Assessment] • Analyze the error for numerical differentiation and integration formulas [Assessment]
Readings : [**Boyce2017**], [**Zill2018**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Numerical Analysis III
1.2 Code	:	MA326
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA325. Numerical Analysis II. (6^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Numerical Analysis 3 focuses on the computational solution of Ordinary Differential Equations (ODEs). In computational mathematics, many systems cannot be solved analytically; therefore, understanding the convergence, stability, and efficiency of numerical integrators is vital. This course covers from basic one-step methods to advanced Runge-Kutta schemes and the specialized treatment of stiff equations, which are critical in simulating chemical reactions, electrical circuits, and structural mechanics.

4. GOALS

- Understand the fundamental theory of numerical methods for Initial Value Problems (IVPs).
- Master the construction and analysis of high-order Runge-Kutta methods.
- Evaluate the stability and convergence of numerical schemes.
- Implement specialized algorithms for Stiff Differential Equations.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Numerical Solution of ODEs: Foundations (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Review of Existence and Uniqueness (Lipschitz continuity) • Elementary methods: Euler, Heun, and Taylor series methods • Error analysis: Local truncation error and Global error • Consistency, Convergence, and Zero-stability	• Formulate discrete approximations for continuous differential systems [Usage] • Derive the local truncation error for basic one-step methods [Assessment] • Analyze the impact of step size on numerical stability [Familiarity]
Readings : [**Burden2016**], [**Heath2002**]	

Unit 2: Runge-Kutta Methods (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• General structure of Runge-Kutta (RK) methods • Butcher Tableaus and order conditions • Explicit and Implicit Runge-Kutta methods • Adaptive step-size control: Runge-Kutta-Fehlberg (RK45)	• Construct high-order RK methods using Butcher Tableaus [Usage] • Implement adaptive algorithms to optimize computational cost [Assessment] • Prove the order of convergence for specific RK schemes [Familiarity]
Readings : [**Quarteroni2007**], [**Butcher2016**]	

Unit 3: Linear Multi-step Methods (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Adams-Bashforth and Adams-Moulton formulas • Predictor-Corrector methods • General Theory of Linear Multi-step methods (Dahlquist theorems) • Comparison between RK and Multi-step methods	• Develop multi-step codes for efficient long-term integration [Usage] • Verify the root condition for zero-stability [Assessment] • Select appropriate methods based on the trade-off between memory and accuracy [Familiarity]
Readings : [**Burden2016**], [**Quarteroni2007**]	

Unit 4: Stiff Equations and Absolute Stability (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Definition and characterization of Stiff Systems • Linear Stability Analysis: The test equation and Stability Regions • A-stability and L-stability • Backward Differentiation Formulas (BDF) and Implicit methods for stiffness	• Identify stiffness in physical systems through eigenvalue analysis [Assessment] • Plot and interpret stability regions in the complex plane [Usage] • Apply implicit solvers to overcome step-size restrictions in stiff problems [Assessment]
Readings : [**Heath2002**], [**Butcher2016**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Mathematical Modeling
1.2 Code	:	MA361
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA325. Numerical Analysis II. (6^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Mathematical Model Formulation (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Modeling process and cycle • Continuous vs. discrete models • Deterministic vs. stochastic models • Assumptions identification and simplification techniques • Dimensional analysis and scaling	• Describe the complete modeling cycle from problem definition to solution interpretation [Familiarity] • Classify models as continuous/discrete and deterministic/stochastic [Assessment] • Identify and justify appropriate assumptions for model simplification [Assessment] • Apply dimensional analysis to reduce complexity of physical models [Usage]
Readings : [**Giordano2013**], [**Lin2009**]	

Unit 2: Continuous Systems Simulation (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Ordinary Differential Equations (ODEs) models • Partial Differential Equations (PDEs) models • Finite difference methods for PDEs • Solver selection and implementation • Stiff systems and adaptive methods	• Formulate ODE and PDE models for physical systems [Usage] • Implement finite difference methods for simple PDEs [Usage] • Select appropriate numerical solvers for different types of ODEs/PDEs [Assessment] • Handle stiff systems using appropriate numerical methods [Usage]
Readings : [**Braun2013**], [**Giordano2013**]	

Unit 3: Discrete Systems Simulation (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Queueing theory and models • Cellular automata and agent-based models • Monte Carlo simulation methods • Markov chains and processes • Discrete-event simulation	• Formulate and analyze simple queueing models [Usage] • Implement cellular automata and agent-based simulations [Usage] • Apply Monte Carlo methods for probabilistic simulation [Usage] • Analyze Markov chains for state transitions and steady-state behavior [Assessment] • Design discrete-event simulations for system analysis [Usage]
Readings : [**Giordano2013**], [**Meerschaert2013**]	

Unit 4: Sensitivity Analysis and Robustness (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Local sensitivity analysis (partial derivatives) • Global sensitivity analysis (Sobol indices) • Morris method and screening techniques • Robustness analysis and worst-case scenarios • Uncertainty propagation	• Perform local sensitivity analysis using derivative-based methods [Usage] • Apply global sensitivity analysis techniques (e.g., Sobol indices) [Usage] • Use Morris method for factor screening [Usage] • Assess model robustness under parameter variations [Assessment] • Quantify uncertainty propagation through models [Assessment]
Readings : [**Banks2010**], [**Giordano2013**]	

Unit 5: Model Validation and Verification (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Verification: numerical code correctness • Validation: comparison with experimental data • Uncertainty Quantification (UQ) in models • Model calibration and parameter estimation • Goodness-of-fit tests and metrics	• Distinguish between verification and validation in modeling [Familiarity] • Apply verification techniques to ensure code correctness [Usage] • Validate models using experimental data and statistical tests [Usage] • Calibrate model parameters using optimization techniques [Usage] • Apply uncertainty quantification methods to assess model reliability [Usage]
Readings : [**Giordano2013**], [**Lin2009**], [**Meerschaert2013**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Technical and professional English V
1.2 Code	:	ID203
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	ID202. Technical and professional English IV. (6^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 6 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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

4. 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.

5. COMPETENCES

ABET-4) An ability to communicate effectively with a range of audiences. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Familiarity)

6. TOPICS

Unit 1: It's a wonderful world (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unit 2: Happiness! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unit 3: Telling tales! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unit 4: Doing the right thing! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unit 5: On the move! (0 hours)	
Competences Expected: ABET-4,ABET-6	
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 : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

Unit 6: I just love it! (0 hours)	
Competences Expected: ABET-4,ABET-6	
Topics	**Learning Outcomes**
• Questions with Shapes Like • Patrones Verbales II • Vocabulario de Comida, Lugares y ocupaciones • Palabras que van unidas en contexto • Expresiones para vistas y sonidos • Composición de Impresiones personales	• At the end of the sixth unit, students having learned the basics of structuring questions with like and with verbal patterns work applied to appropriate contexts. They emphasize the difference between meals, places and people. Describe sights and sounds. They use expressions to compare daily life in different places. They assume the idea of different lifestyles.
Readings : [**Soars023S**], [**Soars023W**], [**Soars023T**], [**Cambridge06**], [**MacGrew99**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Introduction to Machine Learning
1.2 Code	:	AI263
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS261. Artificial Intelligence. (7^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

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

4. GOALS

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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: ML Fundamentals (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Supervised vs unsupervised learning • Overfitting and regularization • Cross-validation and learning curves	• Explain the bias-variance tradeoff [Familiarizarse] • Implement k-fold cross-validation [Usar]
Readings : [**Bishop2006**], [**Goodfellow2016**]	

Unit 2: Linear Models (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Linear and logistic regression • Support Vector Machines (SVM) • Linear discriminant analysis	• Program linear models with scikit-learn [Usar] • Interpret regression coefficients [Evaluar]
Readings : [**Hastie2009**], [**Murphy2012**]	

Unit 3: Ensembles and Trees (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Decision trees and random forests • Gradient Boosting (XGBoost, LightGBM) • Stacking and voting	• Optimize hyperparameters with GridSearch [Usar] • Visualize decision trees [Evaluar]
Readings : [**Hastie2009**], [**Geron2022**]	

Unit 4: Basic Neural Networks (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Multilayer perceptrons (MLP) • Backpropagation and optimizers • Introduction to Keras/TensorFlow	• Build simple neural networks [Usar] • Monitor training with TensorBoard [Evaluar]
Readings : [**Goodfellow2016**], [**Chollet2021**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Capstone Project I
1.2 Code	:	MA401
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	• MA361. Mathematical Modeling. (7^{th} Sem) • MA371. Functional analysis I. (6^{th} Sem) • 140cr
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Assessment)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Assessment)

ABET-4) An ability to communicate effectively with a range of audiences. (Assessment)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Assessment)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Assessment)

6. TOPICS

Unit 1: Project Formulation and Planning (12 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Design management plans for scientific projects [Usar] • Conduct problem studies using research methods [Usar] • Evaluate technical and scientific feasibility of projects [Evaluar]
Readings : [**Babbie2016**], [**Kothari2004**]	

Unit 2: Methodological Development and Tools (12 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Work effectively in multidisciplinary teams [Usar] • Communicate technical progress effectively [Usar] • Select appropriate tools for the project [Evaluar]
Readings : [**Oates2006**], [**Pressman2014**]	

Unit 3: Implementation and Initial Development (12 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Conduct studies using research methods [Usar] • Implement mathematical processes on computer [Usar] • Adapt solutions based on preliminary results [Evaluar]
Readings : [**Sommerville2015**], [**McConnell2004**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Operating systems
1.2 Code	:	CS2S1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	CS210. Algorithms and Data Structures. (4^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The operating system is a collection of services needed to safely interface the hardware with applications. This course explores the mechanisms and policies needed to virtualize computation, memory, and I/O. Students will understand core concepts such as concurrency, persistence, and trust boundaries, which are fundamental for developing robust and efficient software systems.

4. GOALS

- Understand the role and purpose of modern operating systems.
- Master process scheduling and the lifecycle of execution units.
- Analyze memory management, including physical and virtual memory virtualization.
- Evaluate storage systems and file management techniques.
- Comprehend the basics of security and protection within an OS environment.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Usage)

6. TOPICS

Unit 1: Role and Purpose of Operating Systems (8 hours)

Competences Expected: ABET-2,ABET-6

Topics	Learning Outcomes
• Operating systems mediate between general purpose hardware and application-specific software. • Universal operating system functions (e.g., process, user and device interfaces, persistence of data) • Extended and/or specialized operating system functions (e.g., embedded systems, server types such as file, web, multimedia, boot loaders and boot security) • Design issues (e.g., efficiency, robustness, flexibility, portability, security, compatibility, power, safety, tradeoffs between error checking and performance, flexibility and performance, and security and performance). See also: Security Design and Controls Engineering, Threat Analysis and Security Engineering, Trusted Computing and Privacy Engineering • Influences of security, networking, multimedia, parallel and distributed computing • Overarching concern of security/protection: Neglecting to consider security at every layer creates an opportunity to inappropriately access resources. – Unauthorized access to files on an unencrypted drive can be achieved by moving the media to another computer. – Operating systems enforced security can be defeated by infiltrating the boot layer before the operating system is loaded. – Process isolation can be subverted by inadequate authorization checking at API boundaries. – Vulnerabilities in system firmware can provide attack vectors that bypass the operating system entirely. – Improper isolation of virtual machine memory, computing, and hardware can expose the host system to attacks from guest systems. – The operating system may need to mitigate exploitation of hardware and firmware vulnerabilities, leading to potential performance reductions (e.g., Spectre and Meltdown mitigations). • Exposure of operating systems functions in shells and systems programming. See also: Shell Scripting	• Understand the objectives and functions of modern operating systems [Explain] • Evaluate the design issues in different usage scenarios (e.g., real time OS, mobile, server) [Evaluate] • Understand the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve [Explain] • Understand how evolution and stability are desirable and mutually antagonistic in operating systems function [Explain]

Readings : [**Silberschatz2018**], [**Tanenbaum2014**]

Unit 2: Shell Scripting (8 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Error/exception handling • Piping • System commands: – Interface with operating systems • Environment variables • File abstraction and operators • Data structures, such as arrays and lists. • Regular expressions • Programs and processes • Workflow	• Create and execute automated scripts to manage various system tasks [Create] • Solve various text processing problems through scripting [Solve]
Readings : [**Robbins2005**], [**Silberschatz2018**]	

Unit 3: Scheduling (16 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Preemptive and non-preemptive scheduling • Schedulers and policies (e.g., first come, first serve, shortest job first, priority, round robin, multilevel). See also: Resource Management • Concepts of Symmetric Multi-Processor (SMP) scheduling and cache coherence. See also: Memory Hierarchy • Timers (e.g., building many timers out of finite hardware timers). See also: Assembly Level Machine Organization • Fairness and starvation • Subtopics of operating systems such as energy-aware scheduling and real-time scheduling. See also: Performance and Energy Efficiency, Embedded Characteristics and Architectures, Real-Time, Memory and Verification, Systems, Programming and Sensors, Mobile Platforms: Capabilities and Asynchronous Computing, Mobile Platforms: Architecture and Constraints, Mobile Platforms: Development Fundamentals • Cooperative scheduling, such as Linux futexes and userland scheduling.	• Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes [Compare] • Explain the relationships between scheduling algorithms and application domains [Explain] • Explain the distinctions among types of processor scheduler such as short-term, medium-term, long-term, and I/O [Explain] • Evaluate a problem or solution to determine appropriateness for asymmetric and/or symmetric multiprocessing [Evaluate] • Evaluate a problem or solution to determine appropriateness as a process vs threads [Evaluate] • List some contexts benefitting from preemption and deadline scheduling [List] • Explain the ways that the logic embodied in scheduling algorithms are applicable to other operating systems mechanisms, such as first come first serve or priority to disk I/O, network scheduling, project scheduling, and problems beyond computing [Explain]
Readings : [**Silberschatz2018**], [**Stallings2017**]	

Unit 4: Memory Management (16 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Review of physical memory, address translation and memory management hardware See also: Memory Hierarchy, Discrete Mathematics • Impact of memory hierarchy including cache concept, cache lookup, and per-CPU caching on operating system mechanisms and policy See also: Memory Hierarchy, Latency, Cache and Memory Hierarchy, Virtualization and Isolation • Logical and physical addressing, address space virtualization. See also: Memory Hierarchy, Discrete Mathematics • Concepts of paging, page replacement, thrashing and allocation of pages and frames • Allocation/deallocation/storage techniques (algorithms and data structure) performance and flexibility – Arenas, slab allocators, free lists, size classes, heterogeneously sized pages (huge pages) • Memory caching and cache coherence and the effect of flushing the cache to avoid speculative execution vulnerabilities. See also: Functional Organization, Memory Hierarchy, Latency, Cache and Memory Hierarchy, Virtualization and Isolation • Security mechanisms and concepts in memory management including sandboxing, protection, isolation, and relevant vectors of attack. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles • Virtual memory: leveraging virtual memory hardware for OS services and efficiency	• Explain memory hierarchy and cost-performance tradeoffs [Explain] • Summarize the principles of virtual memory as applied to caching and paging [Summarize] • Evaluate the tradeoffs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Evaluate] • Describe the reason for and use of cache memory (performance and proximity, how caches complicate isolation and virtual machine abstraction) [Describe] • Code/Develop efficient programs that consider the effects of page replacement and frame allocation on the performance of a process and the system in which it executes [Create] • Explain how hardware is utilized for efficient virtualization [Explain]
Readings : [**Silberschatz2018**], [**ArpaciDusseau2018**]	

Unit 5: File Systems API and Implementation (12 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Concept of a file including data, metadata, operations, and access-mode • File system mounting • File access control • File sharing • Basic file allocation methods, including linked allocation table • File system structures comprising file allocation including various directory structures and methods for uniquely identifying files (e.g., name, identified or metadata storage location) • Allocation/deallocation/storage techniques (algorithms and data structure) impact on performance and flexibility (i.e., internal and external fragmentation and compaction) • Free space management such as using bit tables vs linking • Implementation of directories to segment and track file location	• Explain the choices to be made in designing file systems [Explain] • Evaluate different approaches to file organization, recognizing the strengths and weaknesses of each [Evaluate] • Apply software constructs appropriately given knowledge of the file system implementation [Apply]
Readings : [**Silberschatz2018**], [**Stallings2017**]	

Unit 6: Protection and Safety (12 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Overview of operating system security mechanisms. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles • Attacks and antagonism (scheduling, etc.). See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles • Review of major vulnerabilities in real operating systems. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles • Operating systems mitigation strategies such as backups. See also: System Reliability • Policy/mechanism separation. See also: Risk and Data Management, Security Governance, Management and Policy • Security methods and devices. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles – Rings of protection (history from Multics to virtualized x86) – x86_64 rings -1 and -2 (hypervisor and ME/PSP) • Protection, access control, and authentication. See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles, Protocols, Applications and Advanced Cryptography, Cryptographic Algorithms, Cryptography Fundamentals	• Understand the requirement for protection and security mechanisms in operating systems [Explain] • List and describe the attack vectors that leverage OS vulnerabilities [List] • Understand the mechanisms available in an OS to control access to resources [Explain] • Summarize the features and limitations of an operating system that impact protection and security [Summarize]
Readings : [**Silberschatz2018**], [**Tanenbaum2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Computer Security
1.2 Code	:	CS3I1
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	CS210. Algorithms and Data Structures. (4^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

In an increasingly digital world, security is no longer an optional feature but a fundamental requirement. This course provides a comprehensive introduction to the principles and practices of cybersecurity, covering everything from theoretical foundations to secure coding and digital forensics. Students will gain the skills necessary to design, implement, and govern secure systems while understanding the ethical and professional responsibilities involved in protecting information and infrastructure.

4. GOALS

- Understand the fundamental principles and mathematical bases of security.
- Apply secure coding techniques and cryptographic methods to software.
- Analyze system vulnerabilities and perform forensic investigations.
- Evaluate security governance and professional ethics in cybersecurity.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Security Mindset and Principles (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Developing a security mindset incorporating cross-cutting concepts: confidentiality, integrity, availability, risk assessment, systems thinking, adversarial thinking, human-centered thinking. • Vulnerabilities, threats, attack surfaces, and attack vectors Protection and Safety. • Denial of Service (DoS) and Distributed Denial of Service (DDoS) Protection and Safety. • Principles and practices of protection, e.g., least privilege, open design, fail-safe defaults, defense in depth, and zero trust; and how they can be implemented Principles of Operating System, Protection and Safety, Coding Practices, Large-Scale Construction and Process, Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Optimization considerations between security, privacy, performance, and other design goals System Design, Software Development Practices, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Concepts of trust and trustworthiness.	• Evaluate a system for possible attacks that can be launched by an adversary [Evaluate] • Design and develop approaches to protect a system from a set of identified threats [Design] • Describe how harm to user privacy can be avoided [Describe] • Develop a system that incorporates various principles of security and privacy [Create]
Readings : [**Pfleeger2015**], [**Stallings2017**]	

Unit 2: Cryptography Fundamentals (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Differences between algorithmic, applied, and mathematical views of cryptography. • Mathematical preliminaries: modular arithmetic, Euclidean algorithm, probabilistic independence, linear algebra basics, number theory, finite fields, complexity, asymptotic analysis Discrete Mathematics, Linear Algebra. • Basic cryptography: symmetric key and public key cryptography Foundational Data Structures, Foundational Algorithms, Advanced Algorithms, Discrete Mathematics. • Classical cryptosystems, such as shift, substitution, transposition ciphers, code books, and machines Discrete Mathematics. • Kerckhoff's principle and use of vetted libraries Coding Practices, Large-Scale Construction and Process.	• Explain the role of cryptography in supporting security and privacy [Explain] • Discuss the risks of inventing one's own cryptographic methods [Debate] • Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Debate] • Implement and cryptanalyze classical ciphers [Implement]
Readings : [**Stallings2017**]	

Unit 3: Injection and Input Validation (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Common vulnerabilities and weaknesses. • SQL injection and other injection attacks. • Cross-site scripting techniques and mitigations. • Input validation and data sanitization Protection and Safety, Basic Programming Concepts and Constructs, Programming Practices: Testing, Debugging, and Documentation, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices.	• Identify underlying problems in given examples of an enumeration of common weaknesses and explain how they can be circumvented [Analyze] • Apply input validation and data sanitization techniques to enhance security of a program [Apply] • Evaluate and prevent SQL injections attacks on a database application [Evaluate] • Evaluate and prevent cross-site scripting attacks against a website [Evaluate] • Construct a program that tests for all input handling errors [Create]
Readings : [**Pfleeger2015**]	

Unit 4: Malware Analysis and Advanced Security (6 hours)	
Competences Expected: ABET-2	
Topics	**Learning Outcomes**
• Offensive security techniques as a defense. • AI-assisted malware detection techniques. • Ransomware: creation, prevention, and mitigation. • Secure use of third-party components Coding Practices, Large-Scale Construction and Process, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Malware: varieties, creation, reverse engineering, and defense against them Language Translation and Execution, Systems Execution and Memory Model. • Assurance: testing (including fuzzing and penetration testing), verification, and validation Protection and Safety, Basic Programming Concepts and Constructs, Programming Practices: Testing, Debugging, and Documentation, Coding Practices, Large-Scale Construction and Process, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Static and dynamic analyses Program Analysis and Analyzers, Protection and Safety, Evaluation, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Secure compilers and secure code generation Language Translation and Execution, Run-time Behavior and Systems.	• Describe different kinds of malicious software [Describe] • Explain the risks of misusing interfaces with third-party code and how to correctly use third-party code [Explain] • Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix [Debate] • Apply static and dynamic tools to identify programming faults [Apply] • Evaluate a system for the existence of malware and remove it [Evaluate] • Implement preventive techniques to reduce the occurrence of ransomware [Implement]
Readings : [**SikorskiHonig2012**], [**Eilam2005**], [**NIST800-53**]	

Unit 5: Threat Analysis and Security Engineering (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Security engineering goals: building systems that remain dependable despite errors, accidents, or malicious adversaries Coding Practices, Large-Scale Construction and Process, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices, Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Problem analysis and situational analysis to address system security Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Security analysis, covering security requirements analysis; security controls analysis; threat analysis; and vulnerability analysis Program Analysis and Analyzers, Evaluation. • Security attack domains and attack surfaces, e.g., communications and networking, hardware, physical, social engineering, software, and supply chain Network Security. • Security attack modes, techniques, and tactics, e.g., authentication abuse; brute force; buffer manipulation; code injection; content insertion; denial of service; eavesdropping; function bypass; impersonation; integrity attack; interception; phishing; protocol analysis; privilege abuse; spoofing; and traffic injection Network Security, Protection and Safety, Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices.	• Create a threat model for a system or system design [Create] • Apply situational analysis to develop secure solutions under a specified scenario [Apply] • Identify and mitigate security vulnerabilities and weaknesses in a system [Analyze]
Readings : [**Anderson2020**]	

Unit 6: Digital Forensics Techniques (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Basic principles and methodologies for digital forensics. • System design for forensics. • Forensics in different situations: operating systems, file systems, application forensics, web forensics, network forensics, mobile device forensics, use of database auditing Network Security. • Attacks on forensics and preventing such attacks.	• Explain what a digital investigation is and how it can be implemented Computer Crimes and Attacks, Security Laws, Policies and Responsibility [Explain] • Design and implement software to support forensics [Design] • Extract data from a hard drive to comply with the law Computer Crimes and Attacks, Security Laws, Policies and Responsibility [Extract] • Recover data based on a given search term from an imaged system [Analyze] • Reconstruct data and events from an application history, or a web artifact, or a cloud database, or a mobile device. Mobile Platforms: Capabilities and Asynchronous Computing, Mobile Platforms: Architecture and Constraints, Mobile Platforms: Development Fundamentals, Cloud Architecture and Data Management, Frontend, Frameworks and Web Standards [Apply] • Capture and analyze network traffic. Network Security [Analyze] • Develop approaches to address the challenges associated with mobile device forensics [Create] • Apply forensics tools to investigate security breaches [Apply] • Identify and mitigate anti-forensic methods [Analyze]
Readings : [**Pfleeger2015**]	

Unit 7: Security Governance, Management and Policy (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Protecting critical assets from threats. • Security governance: organizational objectives and general risk assessment. • Security management: achieve and maintain appropriate levels of confidentiality, integrity, availability, accountability, authenticity, and reliability Performance Testing and Benchmarking, Testing Tools and Analysis, Verification and validation Concepts, Test Planning and Types, Advanced Testing Practices. • Security policy: organizational policies, issue-specific policies, system-specific policies.	• Describe critical assets and how they can be protected [Describe] • Differentiate between security governance, management, and controls, giving examples of each [Differentiate] • Describe a technical control and implement it to mitigate specific threats [Describe]
Readings : [**Anderson2020**]	

Unit 8: Security Ethics, Law and Society (6 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Societal impacts on breakdowns in security and privacy Digital Equity and Accessibility, Social Impact and AI, Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation, Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Applicability of laws and regulations on security and privacy Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Professional ethical considerations when designing secure systems and maintaining privacy; ethical hacking Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation, Ethics Codes and Legal Accountability, Professional Responsibility and Public Policy, Professional Values, Nature and Continuing Development, Ethical Dissent and Cultural Context, Workplace Ethics, Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Security and privacy implications of malicious AI/machine learning actors, e.g., identifying deep fakes Machine Learning Formulation and Evaluation, Deep Learning, RL and Ethics, Introduction to AI: Concepts and Intelligent Behavior, Problem Characteristics and Agent Nature, Machine Learning Fundamentals, Machine Learning Data, Representations and Evaluation, Neural Networks and Ethics, Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation, Computer Crimes and Attacks, Security Laws, Policies and Responsibility. • Societal impacts of Internet of Things (IoT) devices and other emerging technologies on security and privacy Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation, Computer Crimes and Attacks, Security Laws, Policies and Responsibility.	• Calculate the impact of a breakdown in security of a given system [Calculate] • Construct a system that conforms to security laws [Create] • Apply a set of privacy regulations to design a system that protects privacy [Apply] • Evaluate the legal ramifications of a system not corresponding to applicable laws and regulations [Evaluate]
Readings : [**Anderson2020**], [**Pfleeger2015**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Quantum Mechanics
1.2 Code	:	QC251
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• FI101. General Physics. (2^{nd} Sem) • MA371. Functional analysis I. (6^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	4 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Quantum mechanics is fundamental for understanding the physical principles that govern atomic and subatomic systems, which form the basis of modern computational technologies such as quantum computing, quantum cryptography, semiconductors, and solid-state devices. This course provides the mathematical and conceptual foundations of quantum mechanics, with special emphasis on its applications in computer science, including quantum algorithms, quantum simulation, and quantum hardware development.

4. GOALS

- Understand the fundamental postulates of quantum mechanics and their mathematical formulation.
- Apply quantum theory to solve basic problems in few-level quantum systems.
- Relate quantum concepts to applications in quantum computing and information technologies.
- Develop the ability to simulate simple quantum systems using computational tools.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Fundamentals of Quantum Mechanics (12 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Crisis of classical physics: blackbody radiation, photoelectric effect. • Wave-particle duality and complementarity principle. • Postulates of quantum mechanics: wave function, operators, and expectation values. • Time-dependent and time-independent Schrödinger equation. • Born's probabilistic interpretation.	• Explain the limitations of classical physics that motivated the development of quantum mechanics. [Familiarizarse] • State the fundamental postulates of quantum mechanics. [Familiarizarse] • Interpret the physical meaning of the wave function and its relation to probabilities. [Usar] • Formulate simple problems using the Schrödinger equation. [Evaluar]
Readings : [**Griffiths2018**], [**Nielsen2010**]	

Unit 2: Simple Quantum Systems (14 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Particle in a one-dimensional and three-dimensional box. • Quantum harmonic oscillator: algebraic solution and Hermite polynomials. • Quantum tunneling through potential barriers. • Finite potential well and bound states. • Ladder operators for the harmonic oscillator.	• Solve the Schrödinger equation analytically for simple quantum systems. [Familiarizarse] • Calculate energy levels and wave functions for symmetric potential wells. [Usar] • Explain the quantum tunneling phenomenon and its technological applications. [Evaluar] • Use ladder operators to manipulate quantum states. [Evaluar]
Readings : [**Griffiths2018**], [**McIntyre2012**]	

Unit 3: Linear Algebra in Quantum Mechanics (10 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Hilbert spaces and Dirac notation. • Linear operators, Hermitian and unitary operators. • Eigenvalues and eigenvectors in quantum systems. • Matrix representations of quantum operators. • Tensor product and composite spaces.	• Use Dirac notation to represent quantum states. [Familiarizarse] • Differentiate between Hermitian, unitary, and projection operators. [Usar] • Construct matrix representations of simple quantum operators. [Usar] • Apply the tensor product to describe composite quantum systems. [Evaluar]
Readings : [**Shankar2012**], [**Nielsen2010**]	

Unit 4: Angular Momentum and Spin (8 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Orbital angular momentum: operators and eigenfunctions. • Spin: the intrinsic degree of freedom. • Pauli matrices and spin-1/2 representation. • Coupling of angular momenta. • Identical particles and Pauli exclusion principle.	• Calculate angular momentum eigenvalues for simple systems. [Familiarizarse] • Manipulate spin states using Pauli matrices. [Usar] • Apply the Pauli exclusion principle to fermion systems. [Evaluar] • Explain the importance of spin in technologies such as magnetic resonance. [Evaluar]
Readings : [**Griffiths2018**], [**McIntyre2012**]	

Unit 5: Perturbation Theory (10 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Time-independent perturbations: nondegenerate and degenerate theory. • Corrections to energy levels and wave functions. • Time-dependent perturbations: Fermi's golden rule. • Radiation-induced transitions. • Variational method for ground states.	• Apply nondegenerate perturbation theory to simple quantum systems. [Familiarizarse] • Calculate energy level corrections due to small perturbations. [Usar] • Use Fermi's golden rule to calculate transition rates. [Evaluar] • Apply the variational method to estimate ground state energies. [Evaluar]
Readings : [**Griffiths2018**], [**Shankar2012**]	

Unit 6: Applications in Quantum Computing (10 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Qubits: representation and basic operations. • Quantum gates and elementary quantum circuits. • Quantum entanglement and no-cloning theorem. • Simple quantum algorithms: Deutsch-Jozsa and quantum teleportation. • Simulation of quantum systems on classical computers. • Implementation of quantum algorithms using frameworks like Qiskit or Cirq.	• Represent and manipulate qubits using Dirac notation. [Familiarizarse] • Explain the concept of quantum entanglement and its applications. [Usar] • Implement simple quantum circuits using quantum programming frameworks. [Usar] • Simulate few-qubit quantum systems in computational environments. [Evaluar] • Describe the fundamental advantages of quantum computing over classical computing. [Evaluar]
Readings : [**Nielsen2010**], [**Rieffel2011**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Exploratory data analysis
1.2 Code	:	CS253
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA351. Statistics and Probabilities. (6^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Exploratory Data Analysis (EDA) is a critical stage in any data science project. This course provides the tools to understand the structure, detect anomalies, and extract initial patterns from data before formal modeling. It focuses on the use of descriptive statistics, data cleaning, and preliminary visualization to validate assumptions and guide the selection of complex models, ensuring the integrity and quality of scientific findings.

4. GOALS

- Apply data cleaning and preprocessing techniques to ensure data quality.
- Use descriptive statistical methods to summarize complex datasets.
- Identify patterns, outliers, and relationships between variables through visual analysis.
- Formulate hypotheses based on the initial exploration of the data.
- Communicate preliminary findings effectively through technical reports.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Foundations of Exploratory Data Analysis (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Philosophy and objectives of exploratory analysis. • Data types: structured, semi-structured, and unstructured. • Measurement scales: nominal, ordinal, interval, and ratio. • Workflow in data exploration.	• Identify the importance of EDA in the data life cycle [Familiarizarse]. • Classify variables according to their measurement scale and type [Usar].
Readings : [**Tukey1977**], [**Peng2016**]	

Unit 2: Data Cleaning and Preparation (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• ”Tidy Data” concept and table manipulation. • Handling missing values: imputation versus removal. • Detection and treatment of outliers. • Normalization and standardization of numerical variables.	• Implement cleaning routines for noisy datasets [Usar]. • Evaluate the impact of different imputation strategies [Evaluar].
Readings : [**Wickham2014**], [**McKinney2022**]	

Unit 3: Univariate Descriptive Statistics (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Measures of central tendency, dispersion, and shape. • Frequency distributions and density functions. • Skewness and kurtosis analysis in real variables. • Univariate visualization: Histograms, Boxplots, and Kernel Density Estimation (KDE).	• Numerically summarize the distribution of a variable [Usar]. • Interpret density plots to identify biases in the data [Evaluar].
Readings : [**Healy2018**], [**Tukey1977**]	

Unit 4: Bivariate and Multivariate Exploration (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Pearson, Spearman, and Kendall correlation. • Contingency tables and measures of categorical association. • Scatter plots, correlation matrices, and heatmaps. • Conceptual introduction to dimensionality reduction (PCA).	• Quantify the relationship between multiple variables in a dataset [Usar]. • Use heatmaps to detect collinearity in datasets [Evaluar].
Readings : [**Healy2018**], [**Wickham2016**]	

Unit 5: Computational Tools for Exploration (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Computational ecosystems for analysis (Python/Pandas or R/Tidyverse). • Efficient manipulation of large-volume DataFrames. • Automatic generation of data profiling reports.	• Implement automated EDA processes through programming [Usar]. • Optimize data processing for efficient analysis [Evaluar].
Readings : [**McKinney2022**], [**Wickham2016**]	

Unit 6: Exploratory Hypothesis Testing (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Normality tests and variance assumptions. • Group comparison: T-test and ANOVA in an exploratory context. • Non-parametric methods and resampling (Bootstrapping).	• Validate statistical assumptions necessary for subsequent modeling [Usar]. • Design exploratory experiments to confirm visual findings [Evaluar].
Readings : [**Peng2016**], [**Healy2018**]	

Unit 7: Data Ethics and Responsibility (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Identification of selection and confirmation biases in data. • Data privacy: Anonymization and masking techniques. • Ethics in communicating preliminary findings and associated risks.	• Identify potential ethical biases in real-world datasets [Familiarizarse]. • Apply basic anonymization techniques during analysis [Usar].
Readings : [**Healy2018**]	

Unit 8: Result Communication and Data Storytelling (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Structure of a technical data quality report. • Storytelling based on exploratory discovery. • Selection of key visualizations for decision-making.	• Write effective technical reports on data quality [Evaluar]. • Perform presentations of exploratory findings to technical audiences [Evaluar].
Readings : [**Wickham2016**], [**Healy2018**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Big Data and Analytics
1.2 Code	:	CS370
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	CS271. Data Management. (4^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

In the current era, the volume, velocity, and variety of data exceed the capabilities of traditional storage and processing systems. This course introduces the fundamental concepts and technologies of Big Data. It covers distributed file systems, parallel processing frameworks, NoSQL databases, and real-time data streaming. Students will learn to design scalable architectures to extract value from massive datasets.

4. GOALS

- Understand the 5 V's of Big Data and their impact on system design.
- Design and implement scalable solutions using the MapReduce paradigm.
- Master the use of distributed file systems like HDFS.
- Analyze different architectural patterns for Big Data (Lambda and Kappa).
- Apply stream processing techniques for real-time data analysis.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Core Database System Concepts (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Purpose and advantages of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture, data independence, and data abstraction • Transaction management • Normalization • Approaches for managing large volumes of data (e.g., NoSQL database systems, use of MapReduce) • How to support CRUD-only applications • Distributed databases/cloud-based systems • Structured, semi-structured, and unstructured data • Use of a declarative query language • Systems supporting structured and/or stream content	• Identify at least four advantages that using a database system provides [Analyze] • Enumerate the components of a (relational) database system [Enumerate] • Follow a query as it is processed by the components of a (relational) database system [Analyze] • Defend the value of data independence [Defend] • Compose a simple select-project-join query in SQL [Compose] • Enumerate the four properties of a correct transaction manager [Enumerate] • Describe the advantages for eliminating duplicate repeated data [Describe] • Outline how MapReduce uses parallelism to process data efficiently [Outline] • Evaluate the differences between structured and semi/unstructured databases [Evaluate]
Readings : [**White2015**], [**Karau2017**]	

Unit 2: Distributed Databases/Cloud Computing (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Distributed DBMS: – Distributed data storage – Distributed query processing – Distributed transaction model – Homogeneous and heterogeneous solutions – Client-server distributed databases • Parallel DBMS: – Parallel DBMS architectures: shared memory, shared disk, shared nothing – Speedup and scale-up, e.g., use of the MapReduce processing model – Data replication and weak consistency models	• Describe the key components of a distributed DBMS, including distributed data storage, query processing, and transaction management [Describe] • Analyze the tradeoffs between parallel DBMS architectures: shared memory, shared disk, and shared nothing [Analyze] • Describe data replication strategies and weak consistency models in distributed database systems [Describe]
Readings : [**White2015**], [**Lin2010**]	

Unit 3: NoSQL Systems (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Why NoSQL? (e.g., Impedance mismatch between Application [CRUD] and RDBMS) • Key-Value and Document data model • Storage systems (e.g., Key-Value systems, Data Lakes) • Distribution Models (Sharding and Replication) • Graph Databases • Consistency Models (Update and Read, Quorum consistency, CAP theorem) • Processing model (e.g., Map-Reduce, multi-stage map-reduce, incremental map-reduce) • Case Studies: Cloud storage system (e.g., S3); Graph databases; "When not to use NoSQL"	• Develop a use case for the use of NoSQL over RDBMS [Create] • Describe the defining characteristics behind Key-Value and Document-based data models [Describe]
Readings : [**Sadalage2012**], [**Kleppmann2017**]	

Unit 4: Data Analytics (12 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Exploratory data techniques (motivation, representation, descriptive statistics, visualizations) • Data science lifecycle: business understanding, data understanding, data preparation, modeling, evaluation, deployment, and user acceptance • Data mining and machine learning algorithms: e.g., classification, clustering, association, regression • Data acquisition and governance • Data security and privacy considerations • Data fairness and bias • Data visualization techniques and their use in data analytics • Entity Resolution	• Describe several data exploration approaches, including visualization, to understanding unfamiliar datasets [Describe] • Apply several data exploration approaches to understanding unfamiliar datasets [Apply] • Describe basic machine learning/data mining algorithms and when they are appropriate for use [Describe] • Apply several machine learning/data mining algorithms [Apply] • Describe legal and ethical considerations in acquiring, using, and modifying datasets [Describe] • Describe issues of fairness and bias in data collection and usage [Describe]
Readings : [**zaharia2012resilient**], [**chambers2018spark**], [**Kleppmann2017**], [**Psaltis2017**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Inferential Statistics
1.2 Code	:	STA252
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA351. Statistics and Probabilities. (6^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Inferential statistics is crucial in computer science for analyzing data, drawing conclusions, and making evidence-based decisions. This course covers the principles of statistical inference, including point estimation, confidence intervals, and hypothesis testing.

4. GOALS

- Understand the fundamental concepts of inferential statistics.
- Apply point estimation techniques and confidence intervals.
- Conduct hypothesis tests to make data-driven decisions.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Familiarity)

6. TOPICS

Unit 1: Introduction to Statistical Inference (6 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Sampling and sampling distributions. • Point estimation. • Confidence intervals.	• Describe the principles of sampling and sampling distributions. [Familiarizarse] • Calculate point estimates for population parameters. [Usar] • Construct and interpret confidence intervals. [Evaluar]
Readings : [**Devore2020**], [**Montgomery2019**]	

Unit 2: Hypothesis Testing (10 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Null and alternative hypotheses. • Type I and Type II errors. • P-value. • Hypothesis tests for means and proportions.	• Formulate null and alternative hypotheses. [Familiarizarse] • Calculate and interpret the p-value. [Usar] • Conduct hypothesis tests for means and proportions. [Evaluar]
Readings : [**Devore2020**], [**Montgomery2019**]	

Unit 3: Analysis of Variance (ANOVA) (8 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• One-way ANOVA. • Two-way ANOVA.	• Apply one-way ANOVA to compare means of multiple groups. [Familiarizarse] • Conduct two-way ANOVA to analyze the influence of two variables. [Usar]
Readings : [**Devore2020**], [**Montgomery2019**]	

Unit 4: Linear Regression (8 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Simple linear regression. • Multiple linear regression.	• Fit a simple linear regression model. [Familiarizarse] • Interpret the coefficients of a linear regression model. [Usar] • Apply multiple linear regression to model the relationship between multiple variables. [Evaluar]
Readings : [**Devore2020**], [**Montgomery2019**]	

Unit 5: Non-parametric Tests (8 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Chi-square test. • Mann-Whitney U test. • Kruskal-Wallis test.	• Apply non-parametric tests when the assumptions of parametric tests are not met. [Familiarizarse] • Perform the chi-square test for independence. [Usar] • Use the Mann-Whitney U and Kruskal-Wallis tests to compare groups. [Evaluar]
Readings : [**Devore2020**]	

Unit 6: Applications in Computing (8 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Data analysis in computational experiments. • Model validation. • Machine learning (e.g., model evaluation).	• Apply inferential statistics techniques to analyze data in computational experiments. [Familiarizarse] • Use statistical methods to validate computational models. [Usar] • Apply inferential statistics in evaluating machine learning models. [Evaluar]
Readings : [**Devore2020**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Numerical Analysis IV
1.2 Code	:	MA327
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• MA326. Numerical Analysis III. (7^{th} Sem) • MA213. Partial Differential Equations. (7^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	4 HT; 2 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Numerical Analysis 4 provides the computational tools necessary to solve Partial Differential Equations (PDEs), which describe a wide range of phenomena in physics, engineering, and finance. The course focuses on the three fundamental types of second-order PDEs: elliptic, parabolic, and hyperbolic. Students will learn to discretize continuous domains, analyze the stability and convergence of numerical schemes, and implement algorithms for complex boundary value problems.

4. GOALS

- Classify partial differential equations and understand their physical implications.
- Develop and implement Finite Difference Methods (FDM) for different PDE types.
- Analyze the stability (Von Neumann), consistency, and convergence of numerical schemes.
- Introduce advanced discretization techniques such as Finite Element and Finite Volume methods.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Elliptic Equations and Boundary Value Problems (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Classification of second-order PDEs • Finite Difference Method for Laplace and Poisson equations • Boundary conditions: Dirichlet, Neumann, and Robin • Iterative solvers for large sparse systems (Jacobi, Gauss-Seidel, SOR) • Introduction to Finite Element Method (FEM) in 1D	• Discretize steady-state problems using five-point stencils [Usage] • Solve resulting linear systems using optimized iterative methods [Assessment] • Evaluate the effect of grid refinement on the solution accuracy [Familiarity]
Readings : [**Burden2016**], [**Quarteroni2007**]	

Unit 2: Parabolic Equations and Evolution Problems (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• The Heat Equation in 1D and 2D • Explicit schemes and stability restrictions (CFL condition) • Implicit schemes: Crank-Nicolson and Backward Euler • Consistency and Von Neumann stability analysis	• Implement time-stepping algorithms for diffusion processes [Usage] • Determine the stability limits of explicit numerical schemes [Assessment] • Apply the Thomas algorithm for tridiagonal systems in implicit methods [Usage]
Readings : [**Heath2002**], [**Quarteroni2007**]	

Unit 3: Hyperbolic Equations and Wave Phenomena (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• First-order advection equations • The Wave Equation and D'Alembert's solution • Upwind, Lax-Wendroff, and Leapfrog schemes • Dispersion and dissipation errors in wave propagation	• Design numerical schemes for transport and wave equations [Usage] • Analyze numerical diffusion and oscillation in high-order schemes [Assessment] • Verify the Courant-Friedrichs-Lewy (CFL) condition for hyperbolic systems [Usage]
Readings : [**LeVeque2007**], [**Heath2002**]	

Unit 4: Advanced Topics in Numerical PDEs (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Non-linear PDEs and Burgers' equation • Introduction to Finite Volume Method (FVM) for conservation laws • Multidimensional problems and Alternating Direction Implicit (ADI) methods • Computational implementation using Python/MATLAB	• Implement numerical solvers for non-linear conservation laws [Usage] • Generalize 1D methods to complex 2D/3D geometries [Familiarity] • Perform comparative studies of different discretization techniques [Assessment]
Readings : [**Quarteroni2007**], [**LeVeque2007**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Stochastic processes
1.2 Code	:	MA352
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• MA351. Statistics and Probabilities. (6^{th} Sem) • MA210. Calculus III. (4^{th} Sem) • CS261. Artificial Intelligence. (7^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Advanced Probability and Stochastic Processes (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Measure-theoretic probability foundations • Random variables and distributions • Stochastic processes characterization • Markov chains and Markov processes • Brownian motion and diffusion processes	• Apply measure-theoretic concepts to probability spaces [Familiarity] • Characterize different types of stochastic processes [Assessment] • Analyze Markov chains for steady-state and transient behavior [Usage] • Model systems using continuous-time stochastic processes [Usage]
Readings : [**Ross2019**], [**Karlin2014**]	

Unit 2: Discrete Systems Simulation (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Queueing theory and models • Cellular automata and agent-based models • Monte Carlo simulation methods • Markov chains and processes • Discrete-event simulation	• Formulate and analyze simple queueing models [Usage] • Implement cellular automata and agent-based simulations [Usage] • Apply Monte Carlo methods for probabilistic simulation [Usage] • Analyze Markov chains for state transitions and steady-state behavior [Assessment] • Design discrete-event simulations for system analysis [Usage]
Readings : [**Ross2019**], [**Karlin2014**]	

Unit 3: Computational Statistical Inference (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Maximum Likelihood Estimation (MLE) • Bayesian inference methods • Markov Chain Monte Carlo (MCMC) methods • Expectation-Maximization (EM) algorithm • Bootstrap and jackknife resampling	• Implement Maximum Likelihood Estimation for various distributions [Usage] • Apply Bayesian methods for parameter estimation [Usage] • Implement MCMC algorithms for posterior sampling [Usage] • Use EM algorithm for models with latent variables [Usage] • Apply bootstrap methods for confidence intervals [Usage]
Readings : [**Ross2019**], [**Karlin2014**]	

Unit 4: Continuous Systems Simulation (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Ordinary Differential Equations (ODEs) models • Partial Differential Equations (PDEs) models • Finite difference methods for PDEs • Solver selection and implementation • Stiff systems and adaptive methods	• Formulate ODE and PDE models for physical systems [Usage] • Implement finite difference methods for simple PDEs [Usage] • Select appropriate numerical solvers for different types of ODEs/PDEs [Assessment] • Handle stiff systems using appropriate numerical methods [Usage]
Readings : [**Ross2019**], [**Karlin2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Functional analysis II
1.2 Code	:	MA372
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA371. Functional analysis I. (6^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This course deepens the theoretical foundations of measure and integration, providing essential tools for advanced analysis in infinite-dimensional spaces. It focuses on the rigorous construction of abstract measures and the operational aspects of the Lebesgue integral. Finally, the course explores the geometric structure and operator theory in Hilbert spaces, which are fundamental for developing computational methods and models in mathematical physics.

4. GOALS

- Understand and construct abstract measures and measure extensions in general spaces.
- Master Lebesgue integration theory and its fundamental convergence theorems.
- Analyze measure duality and decomposition through the Radon-Nikodym theorem.
- Study advanced geometry and operator theory within Hilbert spaces.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Measure Theory (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Algebras, σ-algebras, and measures • Outer measures and the Carathéodory Extension Theorem • Borel measures on the real line • Signed measures and the Hahn-Jordan decomposition • Complex measures and total variation	• Construct measures from outer measures using the Carathéodory criterion [Assessment] • Apply the Jordan decomposition to analyze signed measures [Usage] • Prove regularity properties of Borel measures [Familiarity]
Readings : [**Folland1999**], [**Royden2010**]	

Unit 2: Integration Theory (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Measurable functions and simple functions • The Lebesgue integral: definition and fundamental properties • Convergence theorems: Monotone Convergence, Dominated Convergence, and Fatou's Lemma • The Radon-Nikodym Theorem and Lebesgue-Stieltjes property • Differentiation of measures and the Fundamental Theorem of Calculus in measure spaces	• Evaluate the convergence of function sequences under the Lebesgue integral [Assessment] • Use the Radon-Nikodym Theorem to find probability densities or measure derivatives [Usage] • Compare Riemann and Lebesgue integrals in terms of applicability [Familiarity]
Readings : [**Folland1999**], [**Brezis2011**]	

Unit 3: Hilbert Spaces (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Review of inner products and completeness • Orthogonality, orthonormal bases, and the Gram-Schmidt process • Riesz Representation Theorem and linear functionals • Adjoint, self-adjoint, and unitary operators in Hilbert spaces • Introduction to spectral theory for compact operators	• Apply the projection theorem to solve best approximation problems [Usage] • Identify the adjoint of classical linear operators [Assessment] • Analyze the spectral structure of self-adjoint operators [Familiarity]
Readings : [**Kreyszig1989**], [**Conway1990**]	

Unit 4: Operator Theory (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Bounded and unbounded operators: definitions and domains • The spectrum of an operator: point, continuous, and residual spectrum • Normal operators and the Spectral Theorem for compact operators • Applications of operator theory in numerical analysis	• Calculate the spectrum of various types of linear operators [Assessment] • Apply the spectral theorem in operator decomposition [Usage] • Solve operational problems using basic functional calculus [Assessment]
Readings : [**Brezis2011**], [**Yosida1980**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Deep Learning
1.2 Code	:	AI264
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	AI263. Introduction to Machine Learning. (8^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

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

4. GOALS

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

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Assessment)

6. TOPICS

Unit 1: Neural Network Training and Optimization (10 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Modern backpropagation: computational graphs and automatic differentiation • Advanced optimizers: momentum, RMSProp, Adam, and learning-rate schedules	• Implement a deep neural network from scratch, including its backpropagation pass [Apply] • Select and tune an optimizer and its learning-rate schedule for training stability [Apply]
Readings : [**goodfellow2016dl**]	

Unit 2: Regularization Techniques for Deep Networks (5 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Dropout regularization • Batch normalization	• Diagnose overfitting in a training run using regularization diagnostics [Analyze] • Apply dropout and batch normalization to improve a model's generalization [Apply]
Readings : [**goodfellow2016dl**]	

Unit 3: Convolutional and Residual Network Architectures (9 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Deep convolutional neural network architectures • Residual connections and ResNets	• Design a deep CNN architecture for a given vision task [Apply] • Adapt a residual architecture to a specific vision domain [Apply]
Readings : [**he2016resnet**]	

Unit 4: Transformer Architectures and Generative Models (16 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Transformers and the attention mechanism • Generative models: variational autoencoders (VAEs) and generative adversarial networks (GANs)	• Train a transformer model on a real dataset [Apply] • Analyze and contrast VAE and GAN generative-model architectures [Analyze]
Readings : [**vaswani2017attention**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Capstone Project II
1.2 Code	:	MA402
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	• MA401. Capstone Project I. (8^{th} Sem) • ID203. Technical and professional English V. (7^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 8 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Assessment)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Assessment)

ABET-4) An ability to communicate effectively with a range of audiences. (Assessment)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Assessment)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Assessment)

6. TOPICS

Unit 1: Advanced Implementation and Optimization (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Formally demonstrate properties of solutions [Evaluar] • Implement complex mathematical processes on computer [Usar] • Optimize computational solutions for efficiency [Evaluar]
Readings : [**McConnell2004**], [**Knuth1997**]	

Unit 2: Results Analysis and Validation (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Conduct rigorous methodological validations [Evaluar] • Recognize formal structure of complex problems [Evaluar] • Interpret results in applied context [Usar]
Readings : [**Montgomery2017**], [**Box2005**]	

Unit 3: Professional Documentation and Communication (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Effectively communicate complex results [Evaluar] • Learn continuously through feedback [Usar] • Document processes professionally [Evaluar]
Readings : [**Alley1996**], [**Matthews2008**]	

Unit 4: Defense and Final Evaluation (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
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] • Manage complex scientific projects [Evaluar] • Solve advanced mathematical problems [Evaluar] • Respond to criticism constructively [Usar]
Readings : [**Turabian2018**], [**Anholt2006**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Networking and Communication
1.2 Code	:	CS231
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS2S1. Operating systems . (8^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Networking and communication are foundational to modern computing systems, distributed applications, and the global internet. This course provides a comprehensive introduction to the full CS2023 Networking and Communication Body of Knowledge—from physical transmission and data link protocols, through reliable transport and routing, to application services, security, mobility, and emerging paradigms such as SDN and quantum networking. Students will develop both theoretical understanding and practical skills applicable to the design and analysis of modern networked systems.

4. GOALS

- Understand the layered architecture of the TCP/IP and OSI models, including the role and responsibilities of each layer.
- Explain single-hop communication mechanisms including MAC protocols, encoding, Ethernet, WiFi, and LAN topologies.
- Master the principles of reliable data transfer, flow control, and congestion control.
- Analyze routing paradigms, forwarding mechanisms, and IP scalability in local and wide area networks.
- Comprehend fundamental application-layer protocols of the Internet (DNS, HTTP, TCP, UDP) and implement socket-based applications.
- Understand network security threats and the countermeasures used to protect networked systems.
- Describe principles of mobile networking including cellular networks, wireless LANs, and device-to-device communication.
- Identify emerging networking paradigms such as SDN, network virtualization, and quantum networking.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Usage)

6. TOPICS

Unit 1: Fundamentals of Networks and Communications (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Importance of networking in contemporary computing, and associated challenges. See also: Digital Equity and Accessibility, Social Impact and AI, Civil Liberties and Expression, Data, Surveillance and Privacy Technology, Privacy Philosophical Foundations and Legislation • Organization of the internet (e.g., users, Internet Service Providers, autonomous systems, content providers, content delivery networks) • Switching techniques (e.g., circuit and packet) • Layers and their roles (application, transport, network, datalink, and physical) • Layering principles (e.g., encapsulation and hourglass model). See also: Digital Systems and Logic, Sequential and Parallel Processing, State Machines • Network elements (e.g., routers, switches, hubs, access points, and hosts) • Basic queueing concepts (e.g., relationship with latency, congestion, service levels, etc.)	• Articulate the organization of the internet [Articulate] • List and define the appropriate network terminology [List] • Describe the layered structure of a typical networked architecture [Describe] • Identify the different types of complexity in a network (edges, core, etc.) [Analyze]
Readings : [**Kurose2021**], [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unit 2: Single Hop Communication (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Introduction to modulation, bandwidth, and communication media • Encoding and Framing • Medium Access Control (MAC) (e.g., random access and scheduled access) • Ethernet and WiFi • Switching (e.g., spanning trees, VLANS). • Local Area Network Topologies (e.g., data center, campus networks).	• Describe some basic aspects of modulation, bandwidth, and communication media [Describe] • Describe in detail a MAC protocol [Describe] • Demonstrate understanding of encoding and framing solution tradeoffs [Demonstrate] • Describe details of the implementation of Ethernet [Describe] • Describe how switching works [Describe] • Describe one kind of a LAN topology [Describe]
Readings : [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unit 3: Networked Applications (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Naming and address schemes (e.g., DNS, and Uniform Resource Identifiers) • Distributed application paradigms (e.g., client/server, peer-to-peer, cloud, edge, and fog). See also: Communication, Coordination • Diversity of networked application demands (e.g., latency, bandwidth, and loss tolerance). See also: Communication, Sustainable Design and Pervasive Computing, Environmental Footprint of Computing Systems, Systemic Effects and Social Context, Digital Equity and Accessibility, Social Impact and AI • Coverage of application-layer protocols (e.g., HTTP) • Interactions with TCP, UDP, and Socket APIs. See also: Programs: Distribution, Programs: Execution Properties, GPU Programming, Programs: Implementation Mappings, Programs: Declarative Parallelism, Programs: Starting Activities	• Define the principles of naming, addressing, resource location [Define] • Analyze the needs of specific networked application demands [Analyze] • Describe the details of one application layer protocol [Describe] • Implement a simple client-server socket-based application [Implement]
Readings : [**Kurose2021**], [**PetersonDavie2022**]	

Unit 4: Reliability Support (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Unreliable delivery (e.g., UDP) • Principles of reliability (e.g., delivery without loss, duplication, or out of order). See also: System Reliability • Error control (e.g., retransmission, error correction) • Flow control (e.g., stop and wait, window based) • Congestion control (e.g., implicit and explicit congestion notification) • TCP and performance issues (e.g., Tahoe, Reno, Vegas, Cubic)	• Describe the operation of reliable delivery protocols [Describe] • List the factors that affect the performance of reliable delivery protocols [List] • Describe some TCP reliability design issues [Describe] • Design and implement a simple reliable protocol [Design]
Readings : [**Kurose2021**], [**Tanenbaum2021**]	

Unit 5: Routing and Forwarding (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Routing paradigms and hierarchy (e.g., intra/inter domain, centralized and decentralized, source routing, virtual circuits, QoS) • Forwarding methods (e.g., forwarding tables and matching algorithms) • IP and Scalability issues (e.g., NAT, CIDR, BGP, different versions of IP)	• Describe various routing paradigms and hierarchies [Describe] • Describe how packets are forwarded in an IP network [Describe] • Describe how the Internet tackles scalability challenges [Describe]
Readings : [**Kurose2021**], [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unit 6: Network Security (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• General intro about security (Threats, vulnerabilities, and countermeasures). See also: Computer Crimes and Attacks, Security Laws, Policies and Responsibility, AI and Hardware Security, Access Control and Applications, Security Mindset and Principles, Security Design and Controls Engineering, Threat Analysis and Security Engineering, Trusted Computing and Privacy Engineering • Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, attacker-in-the-middle, message integrity attacks, routing attacks, ransomware, and traffic analysis). See also: AI and Hardware Security, Access Control and Applications, Security Mindset and Principles, Security Design and Controls Engineering, Threat Analysis and Security Engineering, Trusted Computing and Privacy Engineering • Countermeasures: – Cryptography (e.g. SSL, TLS, symmetric/asymmetric) – Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, DMZ, Zero Trust Network Access, hyper network security, anonymous communication protocols, isolation) – Network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks, BGP Sec, RPKI	• Describe some of the threat models of network security [Describe] • Describe specific network-based countermeasures [Describe] • Analyze various aspects of network security from a case study [Analyze]
Readings : [**Stallings2022**], [**Kurose2021**]	

Unit 7: Mobility Support (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Principles of cellular communication (e.g., 4G, 5G) • Principles of Wireless LANs (mainly 802.11) • Device to device communication (e.g., IoT communication) • Multi-hop wireless networks (e.g., ad hoc networks, opportunistic, delay tolerant)	• Describe some aspects of cellular communication such as registration [Describe] • Describe how 802.11 supports mobile users [Describe] • Describe practical uses of device-to-device communication, as well as multihop [Describe] • Describe one type of mobile network such as ad hoc [Describe]
Readings : [**Tanenbaum2021**], [**PetersonDavie2022**]	

Unit 8: Emerging Topics (6 hours)	
Competences Expected: ABET-2,ABET-6	
Topics	**Learning Outcomes**
• Middleboxes (e.g., advances in usage of AI, intent-based networking, filtering, deep packet inspection, load balancing, NAT, CDN) • Network Virtualization (e.g., SDN, Data Center Networks) • Quantum Networking (e.g., Intro to the domain, teleportation, security, Quantum Internet) • Satellite, mmWave, Visible Light	• Describe the value of advances in middleboxes in networks [Describe] • Describe the importance of Software Defined Networks [Describe] • Describe some of the added value achieved by using Quantum Networking [Describe]
Readings : [**PetersonDavie2022**], [**Tanenbaum2021**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Algebraic Structures
1.2 Code	:	MA381
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA281. Numbers theory. (5^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Algebraic Structures provides the formal and abstract foundation essential for advanced mathematics and its computational applications. This course introduces the fundamental structures of modern algebra—groups, rings, and fields—developing the ability to think in terms of axioms, homomorphisms, and invariants. Mastery of these concepts is crucial for understanding the theoretical underpinnings of areas such as coding theory, cryptography, algorithm design, and algebraic geometry. The course emphasizes the interplay between abstract theory and concrete examples, particularly those arising in computational and discrete mathematics.

4. GOALS

- Master the axiomatic definitions and basic properties of groups, rings, and fields.
- Understand and apply the concepts of subgroups, normal subgroups, quotient groups, and the fundamental homomorphism theorems for groups.
- Analyze the structure of rings, including ideals, quotient rings, integral domains, and fields.
- Apply the theory of finite fields to practical problems in coding theory and cryptography.
- Develop the ability to construct rigorous proofs and recognize algebraic structures in various mathematical and computational contexts.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Groups and Subgroups (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Binary operations and the definition of a group. Basic examples: cyclic, symmetric, dihedral, and matrix groups. • Elementary properties of groups: uniqueness of identity and inverses, cancellation laws. • Subgroups and subgroup tests. Centralizer, normalizer, and center of a group. • Cyclic groups: classification, structure, and subgroups. • Permutation groups: cycle notation, transpositions, and parity.	• State the group axioms and verify whether a given set with an operation forms a group [Familiarizarse]. • Apply subgroup tests and generate subgroups, including cyclic subgroups of a given element [Usar]. • Analyze the structure of finite cyclic groups and decompose permutations into disjoint cycles [Evaluar].
Readings : [**Dummit2004**], [**Gallian2021**]	

Unit 2: Cosets, Normal Subgroups, and Homomorphisms (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Cosets and Lagrange's Theorem. Applications to the order of elements. • Normal subgroups and their characterization. Quotient (factor) groups. • Group homomorphisms and isomorphisms: definitions, properties, and examples. • The Fundamental Homomorphism Theorem (First Isomorphism Theorem) for groups. • Introduction to group actions: definition and basic examples (orbit-stabilizer).	• Compute cosets and use Lagrange's Theorem to deduce properties of finite groups [Familiarizarse]. • Determine if a subgroup is normal and construct the corresponding quotient group [Usar]. • Use the Fundamental Homomorphism Theorem to establish isomorphisms between quotient groups and known groups [Evaluar].
Readings : [**Dummit2004**], [**Judson2021**]	

Unit 3: Rings, Ideals, and Integral Domains (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Definition and basic examples of rings, commutative rings, and rings with unity. • Ring homomorphisms, isomorphisms, and basic properties. • Ideals and quotient rings. The Fundamental Homomorphism Theorem for rings. • Operations on ideals: sum, product, and intersection. • Integral domains, fields, and characteristic of a ring. Finite integral domains are fields.	• Identify the defining properties of rings, integral domains, and fields from given examples [Familiarizarse]. • Verify whether a subset of a ring is an ideal and construct the corresponding quotient ring [Usar]. • Apply ring homomorphism theorems to analyze ring structures and prove basic results about integral domains [Evaluar].
Readings : [**Dummit2004**], [**Hungerford2014**]	

Unit 4: Fields and Applications (16 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Field extensions: algebraic and transcendental elements. Degree of an extension. • Finite fields (Galois fields): existence, uniqueness, and structure. The field $GF(p^n)$. • Polynomial rings over a field. Irreducible polynomials and construction of finite fields. • Primitive elements in finite fields and their application in cryptography (e.g., Diffie-Hellman, AES). • Brief introduction to error-correcting codes: Hamming distance and linear codes as vector spaces over finite fields.	• Describe the classification and basic properties of finite fields [Familiarizarse]. • Construct small finite fields using irreducible polynomials over $\mathbb{Z}_p$ [Usar]. • Explain how the structure of finite fields underpins modern cryptographic protocols and coding theory [Evaluar].
Readings : [**Gallian2021**], [**Lidl1994**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Natural Language Processing
1.2 Code	:	AI262
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	AI263. Introduction to Machine Learning. (8^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

This course introduces the theoretical and practical fundamentals of Natural Language Processing (NLP), covering from classical techniques to modern deep learning models. Students will learn to implement complete NLP pipelines for applications like sentiment analysis, machine translation, and chatbots.

4. GOALS

- Understand computational representation of human language.
- Implement text processing algorithms in Python.
- Apply NLP models to real-world problems using modern frameworks.

5. COMPETENCES

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: NLP Fundamentals (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Tokenization and text representation • N-gram models and smoothing • Feature extraction (TF-IDF, word embeddings)	• Preprocess text for NLP models [Usar] • Explain limitations of classical statistical models [Familiarizarse]
Readings : [**Jurafsky2021**], [**Manning1999**]	

Unit 2: Language Modeling (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Word2Vec, GloVe and FastText • Recurrent Neural Networks (RNNs) for NLP • Sequence-to-sequence models (seq2seq)	• Train word embeddings [Usar] • Implement basic RNNs for text generation [Evaluar]
Readings : [**Goldberg2017**], [**Mikolov2013**]	

Unit 3: Transformers and Modern NLP (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Transformer architecture (self-attention) • BERT and pre-trained models • Fine-tuning for specific tasks	• Apply pre-trained models with HuggingFace [Usar] • Evaluate NLP performance metrics [Evaluar]
Readings : [**Vaswani2017**], [**Devlin2019**]	

Unit 4: Practical Applications (15 hours)	
Competences Expected: ABET-1,ABET-2	
Topics	**Learning Outcomes**
• Sentiment analysis with transformers • Question-answering systems • Model deployment with Flask/FastAPI	• Build a complete NLP pipeline [Usar] • Optimize models for production [Evaluar]
Readings : [**Jurafsky2021**], [**Chollet2021**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Time series analysis
1.2 Code	:	CS273
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA351. Statistics and Probabilities. (6^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

In various fields such as finance, meteorology, and engineering, data is collected sequentially over time, presenting unique temporal dependencies. This course introduces the theoretical and practical foundations for analyzing, modeling, and forecasting time series. It covers everything from classical decomposition methods to advanced stochastic models (ARIMA, GARCH), allowing the student to extract relevant information from dynamic processes and make decisions based on rigorous predictions.

4. GOALS

- Understand the fundamental components of a time series (trend, seasonality, cycle, and residual).
- Identify and transform series to achieve stationarity.
- Model stochastic processes using Box-Jenkins methodologies.
- Evaluate forecast quality through error metrics and temporal cross-validation.
- Implement advanced predictive models for series with volatility or complex dependencies.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Introduction and Fundamental Concepts (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Definition and examples of time series. • Series components: Trend, Seasonality, and Noise. • Lag and difference operators. • Smoothing operations: Simple and weighted moving averages.	• Decompose a time series into its basic components [Usar]. • Identify visual patterns of temporal dependence in real data [Familiarizarse].
Readings : [**Hyndman2021**], [**Box2015**]	

Unit 2: Stationarity and Autocorrelation Functions (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Weak and strong stationarity. • Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). • White noise and Random Walk. • Unit root tests (Augmented Dickey-Fuller).	• Determine if a series is stationary using statistical tests [Usar]. • Interpret correlograms to identify possible stochastic models [Evaluar].
Readings : [**Shumway2017**], [**Box2015**]	

Unit 3: Linear Process Models (AR and MA) (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Autoregressive processes (AR). • Moving Average processes (MA). • Invertibility and stability conditions. • Parameter estimation and information criteria (AIC, BIC).	• Model linear dependencies in stationary time series [Usar]. • Select the optimal model order based on statistical criteria [Evaluar].
Readings : [**Shumway2017**], [**Brockwell2016**]	

Unit 4: Box-Jenkins Methodology and ARIMA Models (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Integrated models (ARIMA). • Identification, estimation, and diagnostic checking. • Residual analysis and Ljung-Box test. • Seasonal models (SARIMA).	• Implement the full Box-Jenkins modeling cycle [Usar]. • Validate model adequacy through residual analysis [Evaluar].
Readings : [**Box2015**], [**Hyndman2021**]	

Unit 5: Forecasting and Model Evaluation (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Point forecasts and confidence intervals. • Error metrics: MAE, RMSE, MAPE, and MASE. • Time Series Cross-Validation (Time Series Split). • Forecast combinations.	• Generate future predictions with their respective uncertainty margins [Usar]. • Rigorously compare the performance of multiple predictive models [Evaluar].
Readings : [**Hyndman2021**], [**Brockwell2016**]	

Unit 6: Volatility Models (ARCH and GARCH) (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Conditional heteroskedasticity in financial series. • ARCH (Autoregressive Conditional Heteroskedasticity) models. • GARCH models and their extensions. • Risk estimation and Value at Risk (VaR).	• Identify the presence of volatility clustering in financial data [Usar]. • Model conditional variance of complex dynamic processes [Evaluar].
Readings : [**Tsay2010**], [**Shumway2017**]	

Unit 7: Ethics and Responsibility in Forecasting (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Socioeconomic consequences of erroneous forecasts. • Transparency in communicating uncertainty. • Ethical use of sensitive temporal data and privacy. • Biases in automated prediction algorithms.	• Recognize the ethical impact of decisions based on predictions [Familiarizarse]. • Communicate model limitations and risks to stakeholders [Usar].
Readings : [**Hyndman2021**]	

Unit 8: Implementation and Real-World Applications (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Computational tools (R: 'forecast', 'fable'; Python: 'statsmodels'). • Automation of forecasting workflows. • Applications in energy demand, inventory, and markets. • Presentation of a final prediction project.	• Develop automated forecasting systems using specialized software [Evaluar]. • Solve a real prediction problem by integrating all course stages [Evaluar].
Readings : [**Tsay2010**], [**Hyndman2021**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Optimization I
1.2 Code	:	MA341
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	MA221. Numerical Analysis I. (5^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Convex and Non-Convex Optimization (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Convex sets and functions • Optimality conditions • Gradient methods for convex optimization • Non-convex optimization challenges • Convergence analysis	• Characterize convex optimization problems [Assessment] • Apply optimality conditions to verify solutions [Usage] • Implement gradient-based methods for convex problems [Usage] • Analyze challenges in non-convex optimization [Assessment]
Readings : [**Nocedal2006**], [**Boyd2004**]	

Unit 2: Constrained Optimization (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Linear programming (simplex method) • Integer programming • Lagrange multipliers and KKT conditions • Penalty and barrier methods • Duality theory	• Solve linear programming problems [Usage] • Formulate and solve integer programming problems [Usage] • Apply KKT conditions to constrained problems [Usage] • Use penalty methods for constrained optimization [Usage]
Readings : [**Nocedal2006**], [**Boyd2004**]	

Unit 3: Optimization for Machine Learning (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Stochastic gradient descent • Momentum and adaptive methods (Adam, RMSProp) • Regularization and sparsity • Distributed optimization for ML • Second-order methods for ML	• Implement stochastic gradient descent variants [Usage] • Apply adaptive optimization methods [Usage] • Use regularization for model generalization [Usage] • Design distributed optimization strategies [Usage]
Readings : [**Nocedal2006**], [**Bertsekas2016**]	

Unit 4: Metaheuristics and Bio-Inspired Optimization (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Genetic algorithms and evolutionary strategies • Swarm intelligence (PSO, ant colony) • Simulated annealing • Tabu search • Hybrid metaheuristics	• Implement genetic algorithms [Usage] • Apply swarm intelligence methods [Usage] • Use simulated annealing for global optimization [Usage] • Design hybrid metaheuristic approaches [Usage]
Readings : [**Talbi2009**], [**Mirjalili2019**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Statistical Learning Theory
1.2 Code	:	MA355
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	• MA352. Stochastic processes. (8^{th} Sem) • MA372. Functional analysis II. (8^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. 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.

4. 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.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Foundations of Statistical Learning (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Bias-variance tradeoff • Vapnik-Chervonenkis (VC) theory • Empirical risk minimization • Generalization bounds • Learning theory fundamentals	• Analyze the bias-variance tradeoff in model selection [Assessment] • Apply VC theory to analyze model complexity [Usage] • Implement empirical risk minimization [Usage] • Interpret generalization bounds [Familiarity]
Readings : [**Vapnik1998**], [**Hastie2009**]	

Unit 2: Supervised Learning Methods (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Linear models and extensions • Support Vector Machines (SVM) • Decision trees and ensemble methods • Kernel methods • Probabilistic supervised learning	• Implement and analyze linear models [Usage] • Apply SVM for classification and regression [Usage] • Construct ensemble methods (bagging, boosting) [Usage] • Use kernel methods for nonlinear problems [Usage]
Readings : [**Hastie2009**], [**James2013**]	

Unit 3: Unsupervised Learning Methods (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Clustering algorithms (k-means, hierarchical) • Dimensionality reduction (PCA, t-SNE) • Density estimation • Manifold learning • Autoencoders and deep unsupervised learning	• Apply clustering algorithms to data exploration [Usage] • Implement PCA and other dimensionality reduction methods [Usage] • Use density estimation for unsupervised learning [Usage] • Analyze manifold learning techniques [Assessment]
Readings : [**Murphy2012**], [**Bishop2006**]	

Unit 4: Model Evaluation and Selection (20 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Cross-validation strategies • Performance metrics for classification and regression • Model comparison and statistical tests • Hyperparameter optimization • Learning curves and diagnostic tools	• Implement cross-validation for model assessment [Usage] • Select appropriate performance metrics [Assessment] • Compare models using statistical tests [Usage] • Optimize hyperparameters using systematic methods [Usage]
Readings : [**Hastie2009**], [**James2013**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Pre-professional internships
1.2 Code	:	CS400
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	3
1.5 Prerrequisites	:	110Cr
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

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

4. GOALS

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

5. COMPETENCES

No items for this table

6. TOPICS

Unit 1: Performance Evaluation in Work Environment (48 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Student integration into a professional work team. • Application of technical knowledge in real projects. • Evaluation of ethical and professional competencies in the work environment. • Presentation of reports and feedback from the company and university.	• Demonstrate technical skills in a real work environment [Evaluar]. • Apply ethical and professional principles in their activities [Evaluar]. • Work effectively in teams and communicate professionally [Evaluar].
Readings : [**ACMInternshipGuide**], [**IEEEProfessionalPractices**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Computational Vision
1.2 Code	:	AI268
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	AI264. Deep Learning. (9^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Computer Vision is a key area of Artificial Intelligence that enables machines to interpret and understand the visual world. This course covers fundamental techniques for image processing, feature extraction, and object recognition. With the advent of deep learning, computer vision has become essential for autonomous vehicles, medical imaging, and augmented reality. Students will learn the mathematical foundations and algorithmic approaches to solve real-world visual perception problems.

4. GOALS

- Understand the fundamentals of image formation and digital representation.
- Master image processing techniques such as filtering and edge detection.
- Apply feature extraction methods to identify key patterns in images.
- Implement object detection and recognition using deep learning models.
- Analyze 3D vision and motion estimation techniques.

5. COMPETENCES

No items for this table

6. TOPICS

Unit 1: Introduction to AI: Concepts and Intelligent Behavior (5 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Overview of AI problems, Examples of successful recent AI applications • Definitions of agents with examples (e.g., reactive, deliberative) • What is intelligent behavior?: – The Turing test and its flaws – Multimodal input and output – Simulation of intelligent behavior – Rational versus non-rational reasoning • Overview of AI Applications, growth, and impact (economic, societal, ethics) • Philosophical issues • History of AI	• Describe the Turing test and the "Chinese Room" thought experiment [Explain] • Differentiate between optimal reasoning/behavior and human-like reasoning/behavior [Evaluate] • Differentiate the terms: AI, machine learning, and deep learning [Explain]
Readings : [**Szeliski2022**], [**Gonzales2018**]	

Unit 2: Problem Characteristics and Agent Nature (5 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Problem characteristics: – Fully versus partially observable – Single versus multi-agent – Deterministic versus stochastic – Static versus dynamic – Discrete versus continuous • Nature of agents: – Autonomous, semi-autonomous, mixed-initiative autonomy – Reflexive, goal-based, and utility-based – Decision making under uncertainty and with incomplete information – The importance of perception and environmental interactions – Learning-based agents – Embodied agents: ∗ sensors, dynamics, effectors • Practice identifying problem characteristics in example environments • Additional depth on nature of agents with examples • Additional depth on AI Applications, Growth, and Impact (economic, societal, ethics, security)	• Enumerate the characteristics of a specific problem [Apply]
Readings : [**Szeliski2022**], [**Gonzales2018**]	

Unit 3: Perception and Computer Vision (12 hours)

Competences Expected:

Topics	**Learning Outcomes**

• Computer vision:
  – Image acquisition, representation, processing, and properties
  – Shape representation, object recognition, and segmentation
  – Motion analysis
  – Generative models
• Audio and speech recognition
• Touch and proprioception
• Other modalities (e.g., olfaction)
• Modularity in recognition
• Approaches to pattern recognition:
  – Classification algorithms and measures of classification quality
  – Statistical techniques
  – Deep learning techniques
	• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Summarize]
	• List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses [List]
	• Implement 2d object recognition based on contour-based and/or region-based shape representations [Apply]
	• Distinguish the goals of sound-recognition, speech-recognition, and speaker-recognition and identify how the raw audio signal will be handled differently in each of these cases [Distinguish]
	• Provide at least two examples of a transformation of a data source from one sensory domain to another, e.g., tactile data interpreted as single-band 2d images [Generate]
	• Implement a feature-extraction algorithm on real data, e.g., an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal [Apply]
	• Implement an algorithm combining features into higher-level percepts, e.g., a contour or polygon from visual primitives or phoneme hypotheses from an audio signal [Apply]
	• Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification [Apply]
	• Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above [Evaluate]
	• Describe at least three classification approaches, their prerequisites for applicability, their strengths, and their shortcomings [Describe]
	• Implement and evaluate a deep learning solution to problems in computer vision, such as object or scene recognition [Apply]

Readings : [**Szeliski2022**], [**Prince2012**]	

Unit 4: Machine Learning Fundamentals (5 hours)

Competences Expected:

Topics	**Learning Outcomes**
• Definition and examples of a broad variety of machine learning tasks: – Supervised learning: ∗ Classification ∗ Regression – Reinforcement learning – Unsupervised learning: ∗ Clustering • Fundamental ideas: – No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences. – Sources of error and undecidability in machine learning • A simple statistical-based supervised learning such as linear regression or decision trees: – Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly • The overfitting problem/controlling solution complexity (regularization, pruning - intuition only): – The bias (underfitting) - variance (overfitting) tradeoff	• Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain [Explain] • Differentiate the terms of AI, machine learning, and deep learning [Evaluate] • Frame an application as a classification problem, including the available input features and output to be predicted (e.g., identifying alphabetic characters from pixel grid input) [Apply] • Identify overfitting in the context of a problem and learning curves and describe solutions to overfitting [Evaluate] • Explain how machine learning works as an optimization/search process [Explain]

Readings : [**Goodfellow2016**], [**Russell2020**]

Unit 5: Machine Learning Data, Representations and Evaluation (5 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Working with Data: – Data preprocessing: ∗ Importance and pitfalls of preprocessing choices – Handling missing values (imputing, flag-as-missing): ∗ Implications of imputing vs flag-as-missing – Encoding categorical variables, encoding real-valued data – Normalization/standardization – Emphasis on real data, not textbook examples • Representations: – Hypothesis spaces and complexity – Simple basis feature expansion, such as squaring univariate features – Learned feature representations • Machine learning evaluation: – Separation of train, validation, and test sets – Performance metrics for classifiers – Estimation of test performance on held-out data – Tuning the parameters of a machine learning model with a validation set – Importance of understanding what a model is doing, where its pitfalls/shortcomings are, and the implications of its decisions	• Apply two or more simple statistical learning algorithms to a classification task and measure the classifiers' accuracy [Apply] • Implement a statistical learning algorithm and the corresponding optimization process to train the classifier and obtain a prediction on new data [Apply] • Explain proper ML evaluation procedures, including the differences between training and testing performance, and what can go wrong with the evaluation process leading to inaccurate reporting of ML performance [Explain] • Compare two machine learning algorithms on a dataset, implementing the data preprocessing and evaluation methodology (e.g., metrics and handling of train/test splits) from scratch [Apply]
Readings : [**Goodfellow2016**], [**Russell2020**]	

Unit 6: Neural Networks and Ethics (4 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Basic neural networks: – Fundamentals of understanding how neural networks work and their training process, without details of the calculations – Basic introduction to generative neural networks (e.g., large language models) • Ethics for Machine Learning: – Focus on real data, real scenarios, and case studies – Dataset/algorithmic/evaluation bias and unintended consequences	• Describe the neural network training process and resulting learned representations [Explain]
Readings : [**Goodfellow2016**], [**Russell2020**]	

Unit 7: Machine Learning Formulation and Evaluation (3 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Formulation of simple machine learning as an optimization problem, such as least squares linear regression or logistic regression: – Objective function – Gradient descent – Regularization to avoid overfitting (mathematical formulation) • Ensembles of models: – Simple weighted majority combination • Performance evaluation: – Other metrics for classification (e.g., error, precision, recall) – Performance metrics for regressors – Confusion matrix – Cross-validation: ∗ Parameter tuning (grid/random search, via cross-validation)	• Compare and contrast several learning techniques (e.g., decision trees, logistic regression, naive Bayes, neural networks, and belief networks), providing examples of when each strategy is superior [Compare] • Evaluate the performance of a simple learning system on a real-world dataset [Evaluate] • Characterize the state of the art in learning theory, including its achievements and shortcomings [Explain] • Explain the problem of overfitting, along with techniques for detecting and managing the problem [Explain] • Explain the triple tradeoff among the size of a hypothesis space, the size of the training set, and performance accuracy [Explain]
Readings : [**Goodfellow2016**]	

Unit 8: Deep Learning, RL and Ethics (3 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Deep learning: – Deep feed-forward networks (intuition only, no mathematics) – Convolutional neural networks (intuition only, no mathematics) – Visualization of learned feature representations from deep nets – Other architectures (generative NN, recurrent NN, transformers, etc.) • Overview of reinforcement learning methods • Two or more applications of machine learning algorithms: – E.g., medicine and health, economics, vision, natural language, robotics, game play • Ethics for Machine Learning: – Continued focus on real data, real scenarios, and case studies – Privacy – Fairness – Intellectual property – Explainability	• Visualize the training progress of a neural network through learning curves in a well-established toolkit (e.g., TensorBoard) and visualize the learned features of the network [Apply] • Given a real-world application of machine learning, describe ethical issues regarding the choices of data, preprocessing steps, algorithm selection, and visualization/presentation of results [Apply]
Readings : [**Goodfellow2016**]	

Unit 9: Perception and Computer Vision (12 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Computer vision: – Image acquisition, representation, processing, and properties – Shape representation, object recognition, and segmentation – Motion analysis – Generative models • Audio and speech recognition • Touch and proprioception • Other modalities (e.g., olfaction) • Modularity in recognition • Approaches to pattern recognition: – Classification algorithms and measures of classification quality – Statistical techniques – Deep learning techniques	• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Summarize] • List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses [List] • Implement 2d object recognition based on contour-based and/or region-based shape representations [Apply] • Distinguish the goals of sound-recognition, speech-recognition, and speaker-recognition and identify how the raw audio signal will be handled differently in each of these cases [Distinguish] • Provide at least two examples of a transformation of a data source from one sensory domain to another, e.g., tactile data interpreted as single-band 2d images [Generate] • Implement a feature-extraction algorithm on real data, e.g., an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal [Apply] • Implement an algorithm combining features into higher-level percepts, e.g., a contour or polygon from visual primitives or phoneme hypotheses from an audio signal [Apply] • Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification [Apply] • Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above [Evaluate] • Describe at least three classification approaches, their prerequisites for applicability, their strengths, and their shortcomings [Describe] • Implement and evaluate a deep learning solution to problems in computer vision, such as object or scene recognition [Apply]
Readings : [**Szeliski2022**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Topics in Artificial Intelligence
1.2 Code	:	AI367
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	AI264. Deep Learning. (9^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	3 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

Artificial Intelligence is a rapidly evolving field. This course explores advanced and contemporary topics that are currently shaping the industry and research. Focus is placed on Generative AI, Large Language Models (LLMs), and Advanced Reinforcement Learning. Students will analyze the architectures behind modern AI systems, such as Transformers, and discuss the technical and societal challenges of deploying autonomous intelligent agents at scale.

4. GOALS

- Understand the architecture and training process of Large Language Models.
- Master the principles of Generative AI for text and image synthesis.
- Implement advanced Reinforcement Learning algorithms for decision making.
- Apply prompt engineering and fine-tuning techniques to existing models.
- Critically evaluate the latest research papers and trends in AI.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

6. TOPICS

Unit 1: Machine Learning Fundamentals (6 hours)

Competences Expected: ABET-1

Topics	**Learning Outcomes**
• Definition and examples of a broad variety of machine learning tasks: – Supervised learning: ∗ Classification ∗ Regression – Reinforcement learning – Unsupervised learning: ∗ Clustering • Fundamental ideas: – No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences. – Sources of error and undecidability in machine learning • A simple statistical-based supervised learning such as linear regression or decision trees: – Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly • The overfitting problem/controlling solution complexity (regularization, pruning - intuition only): – The bias (underfitting) - variance (overfitting) tradeoff	• Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain [Explain] • Differentiate the terms of AI, machine learning, and deep learning [Evaluate] • Frame an application as a classification problem, including the available input features and output to be predicted (e.g., identifying alphabetic characters from pixel grid input) [Apply] • Identify overfitting in the context of a problem and learning curves and describe solutions to overfitting [Evaluate] • Explain how machine learning works as an optimization/search process [Explain]

Readings : [**Goodfellow2016**], [**Foster2022**]

Unit 2: Machine Learning Data, Representations and Evaluation (5 hours)

Competences Expected: ABET-1

Topics	**Learning Outcomes**
• Working with Data: – Data preprocessing: ∗ Importance and pitfalls of preprocessing choices – Handling missing values (imputing, flag-as-missing): ∗ Implications of imputing vs flag-as-missing – Encoding categorical variables, encoding real-valued data – Normalization/standardization – Emphasis on real data, not textbook examples • Representations: – Hypothesis spaces and complexity – Simple basis feature expansion, such as squaring univariate features – Learned feature representations • Machine learning evaluation: – Separation of train, validation, and test sets – Performance metrics for classifiers – Estimation of test performance on held-out data – Tuning the parameters of a machine learning model with a validation set – Importance of understanding what a model is doing, where its pitfalls/shortcomings are, and the implications of its decisions	• Apply two or more simple statistical learning algorithms to a classification task and measure the classifiers' accuracy [Apply] • Implement a statistical learning algorithm and the corresponding optimization process to train the classifier and obtain a prediction on new data [Apply] • Explain proper ML evaluation procedures, including the differences between training and testing performance, and what can go wrong with the evaluation process leading to inaccurate reporting of ML performance [Explain] • Compare two machine learning algorithms on a dataset, implementing the data preprocessing and evaluation methodology (e.g., metrics and handling of train/test splits) from scratch [Apply]

Readings : [**Goodfellow2016**], [**Foster2022**]

Unit 3: Neural Networks and Ethics (5 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Basic neural networks: – Fundamentals of understanding how neural networks work and their training process, without details of the calculations – Basic introduction to generative neural networks (e.g., large language models) • Ethics for Machine Learning: – Focus on real data, real scenarios, and case studies – Dataset/algorithmic/evaluation bias and unintended consequences	• Describe the neural network training process and resulting learned representations [Explain]
Readings : [**Goodfellow2016**], [**Foster2022**]	

Unit 4: Machine Learning Formulation and Evaluation (3 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Formulation of simple machine learning as an optimization problem, such as least squares linear regression or logistic regression: – Objective function – Gradient descent – Regularization to avoid overfitting (mathematical formulation) • Ensembles of models: – Simple weighted majority combination • Performance evaluation: – Other metrics for classification (e.g., error, precision, recall) – Performance metrics for regressors – Confusion matrix – Cross-validation: ∗ Parameter tuning (grid/random search, via cross-validation)	• Compare and contrast several learning techniques (e.g., decision trees, logistic regression, naive Bayes, neural networks, and belief networks), providing examples of when each strategy is superior [Compare] • Evaluate the performance of a simple learning system on a real-world dataset [Evaluate] • Characterize the state of the art in learning theory, including its achievements and shortcomings [Explain] • Explain the problem of overfitting, along with techniques for detecting and managing the problem [Explain] • Explain the triple tradeoff among the size of a hypothesis space, the size of the training set, and performance accuracy [Explain]
Readings : [**Goodfellow2016**]	

Unit 5: Deep Learning, RL and Ethics (3 hours)	
Competences Expected:	
Topics	**Learning Outcomes**
• Deep learning: – Deep feed-forward networks (intuition only, no mathematics) – Convolutional neural networks (intuition only, no mathematics) – Visualization of learned feature representations from deep nets – Other architectures (generative NN, recurrent NN, transformers, etc.) • Overview of reinforcement learning methods • Two or more applications of machine learning algorithms: – E.g., medicine and health, economics, vision, natural language, robotics, game play • Ethics for Machine Learning: – Continued focus on real data, real scenarios, and case studies – Privacy – Fairness – Intellectual property – Explainability	• Visualize the training progress of a neural network through learning curves in a well-established toolkit (e.g., TensorBoard) and visualize the learned features of the network [Apply] • Given a real-world application of machine learning, describe ethical issues regarding the choices of data, preprocessing steps, algorithm selection, and visualization/presentation of results [Apply]
Readings : [**Goodfellow2016**]	

Unit 6: Natural Language Processing (20 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Deterministic and stochastic grammars • Parsing algorithms: – CFGs and chart parsers (e.g., CYK) – Probabilistic CFGs and weighted CYK • Representing meaning/Semantics: – Logic-based knowledge representations – Semantic roles – Temporal representations – Beliefs, desires, and intentions • Corpus-based methods • N-grams and HMMs • Smoothing and backoff • Examples of use: POS tagging and morphology • Information retrieval: – Vector space model: ∗ TF & IDF – Precision and recall • Information extraction • Language translation • Text classification, categorization: – Bag of words model • Deep learning for NLP: – RNNs – Transformers – Multi-modal embeddings (e.g., images + text) – Generative language models	• Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each [Define] • Simulate, apply, or implement classic and stochastic algorithms for parsing natural language [Apply] • Identify the challenges of representing meaning [Analyze] • List the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks [List] • Identify techniques for information retrieval, language translation, and text classification [Analyze] • Implement a TF/IDF transform, use it to extract features from a corpus, and train an off-the-shelf machine learning algorithm using those features to do text classification [Apply]
Readings : [**Tunstall2022**], [**Russell2020**]	

Unit 7: Machine Learning Fundamentals (6 hours)

Competences Expected: ABET-1

Topics	**Learning Outcomes**
• Definition and examples of a broad variety of machine learning tasks: – Supervised learning: ∗ Classification ∗ Regression – Reinforcement learning – Unsupervised learning: ∗ Clustering • Fundamental ideas: – No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences. – Sources of error and undecidability in machine learning • A simple statistical-based supervised learning such as linear regression or decision trees: – Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly • The overfitting problem/controlling solution complexity (regularization, pruning - intuition only): – The bias (underfitting) - variance (overfitting) tradeoff	• Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain [Explain] • Differentiate the terms of AI, machine learning, and deep learning [Evaluate] • Frame an application as a classification problem, including the available input features and output to be predicted (e.g., identifying alphabetic characters from pixel grid input) [Apply] • Identify overfitting in the context of a problem and learning curves and describe solutions to overfitting [Evaluate] • Explain how machine learning works as an optimization/search process [Explain]

Readings : [**Sutton2018**]

Unit 8: Machine Learning Data, Representations and Evaluation (5 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Working with Data: – Data preprocessing: ∗ Importance and pitfalls of preprocessing choices – Handling missing values (imputing, flag-as-missing): ∗ Implications of imputing vs flag-as-missing – Encoding categorical variables, encoding real-valued data – Normalization/standardization – Emphasis on real data, not textbook examples • Representations: – Hypothesis spaces and complexity – Simple basis feature expansion, such as squaring univariate features – Learned feature representations • Machine learning evaluation: – Separation of train, validation, and test sets – Performance metrics for classifiers – Estimation of test performance on held-out data – Tuning the parameters of a machine learning model with a validation set – Importance of understanding what a model is doing, where its pitfalls/shortcomings are, and the implications of its decisions	• Apply two or more simple statistical learning algorithms to a classification task and measure the classifiers' accuracy [Apply] • Implement a statistical learning algorithm and the corresponding optimization process to train the classifier and obtain a prediction on new data [Apply] • Explain proper ML evaluation procedures, including the differences between training and testing performance, and what can go wrong with the evaluation process leading to inaccurate reporting of ML performance [Explain] • Compare two machine learning algorithms on a dataset, implementing the data preprocessing and evaluation methodology (e.g., metrics and handling of train/test splits) from scratch [Apply]
Readings : [**Sutton2018**]	

Unit 9: Neural Networks and Ethics (5 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• Basic neural networks: – Fundamentals of understanding how neural networks work and their training process, without details of the calculations – Basic introduction to generative neural networks (e.g., large language models) • Ethics for Machine Learning: – Focus on real data, real scenarios, and case studies – Dataset/algorithmic/evaluation bias and unintended consequences	• Describe the neural network training process and resulting learned representations [Explain]
Readings : [**Sutton2018**]	

Unit 10: AI Applications and Societal Impact (6 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• At least one application of AI to a specific problem and field, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose at least one for the CS Core): – Formulating and evaluating a specific application as an AI problem: ∗ How to deal with underspecified or ill-posed problems – Data availability/scarcity and cleanliness: ∗ Basic data cleaning and preprocessing ∗ Data set bias – Algorithmic bias – Evaluation bias – Assessment of societal implications of the application • Deployed deep generative models: – High-level overview of deep image generative models (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion, etc.), their uses, and their shortcomings/pitfalls. – High-level overview of large language models (e.g., as of 2023, ChatGPT, Bard, etc.), their uses, and their shortcomings/pitfalls. • Overview of societal impact of AI: – Ethics – Fairness – Trust/explainability – Privacy and usage of training data – Human autonomy and oversight/regulations/legal requirements – Sustainability	• Given a real-world application domain and problem, formulate an AI solution to it, identifying proper data/input, preprocessing, representations, AI techniques, and evaluation metrics/methodology [Apply]
Readings : [**Russell2020**], [**Bostrom2014**]	

Unit 11: Additional AI Applications and Societal Impact (6 hours)	
Competences Expected: ABET-1	
Topics	**Learning Outcomes**
• One or more additional applications of AI to a broad set of problems and diverse fields, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose a different area from that chosen for the CS Core): – Formulating and evaluating a specific application as an AI problem: ∗ How to deal with underspecified or ill-posed problems – Data availability/scarcity and cleanliness: ∗ Basic data cleaning and preprocessing ∗ Data set bias – Algorithmic bias – Evaluation bias – Assessment of societal implications of the application • Additional depth on deployed deep generative models: – Introduction to how deep image generative models work, (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion) including discussion of attention – Introduction to how large language models work, (e.g., as of 2023, ChatGPT, Bard) including discussion of attention – Idea of foundational models, how to use them, and the benefits/issues with training them from big data • Analysis and discussion of the societal impact of AI: – Ethics – Fairness – Trust/explainability – Privacy and usage of training data – Human autonomy and oversight/regulations/legal requirements – Sustainability	• Analyze the societal impact of one or more specific real-world AI applications, identifying issues regarding ethics, fairness, bias, trust, and explainability [Evaluate] • Describe some of the failure modes of current deep generative models for language or images, and how this could affect their use in an application [Describe]
Readings : [**Russell2020**], [**Bostrom2014**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Capstone Project III
1.2 Code	:	MA403
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	5
1.5 Prerrequisites	:	MA402. Capstone Project II. (9^{th} Sem)
1.6 Type of Course	:	Mandatory
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	1 HT; 8 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

The Undergraduate Thesis in Computational Mathematics (MA403) course is the culmination of the student's academic training and research process. This course focuses on the consolidation of the final thesis report, ensuring compliance with scientific standards and academic integrity. The student must demonstrate a total integration of mathematical knowledge and computational skills to solve a research problem, finalizing with the formal presentation of the manuscript and technical preparation for the defense before the academic and professional community.

4. GOALS

- Fully implement the designed computational solution, applying advanced computational mathematics techniques.
- Perform exhaustive 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 mastery of the subject and technical communication skills.
- Demonstrate capacity for autonomous work, project management, and professional ethics in the development of applied research.

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Assessment)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Assessment)

ABET-4) An ability to communicate effectively with a range of audiences. (Assessment)

ABET-5) An ability to understand ethical and professional responsibilities and the impact of technical and/or scientific solutions in global, economic, environmental, and societal contexts. (Assessment)

ABET-6) An ability to function effectively on teams that establish goals, plan tasks, meet deadlines, and analyze risk and uncertainty. (Assessment)

6. TOPICS

Unit 1: Technical Consolidation and Final Optimization (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
Topics	**Learning Outcomes**
• Final review of algorithmic implementation and computational efficiency • Code optimization and scalability of the solution • Formal verification and validation of numerical results • Software repository management and scientific reproducibility	• Conduct complete studies using research methods [Evaluar] • Formally demonstrate properties of solutions [Evaluar] • Implement complex mathematical processes on a computer [Usar] • Optimize computational solutions for efficiency [Evaluar]
Readings : [**McConnell2004**], [**Knuth1997**]	

Unit 2: Methodological Validation and Critical Analysis (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
Topics	**Learning Outcomes**
• Comparative analysis of results against the state of the art • In-depth discussion on the impact and applications of the model • Final statistical validation and error analysis • Evaluation of the original contribution of the work	• Analyze social impact of scientific solutions [Evaluar] • Conduct rigorous methodological validations [Evaluar] • Recognize formal structure of complex problems [Evaluar] • Interpret results in an applied context [Usar]
Readings : [**Montgomery2017**], [**Box2005**]	

Unit 3: Academic Writing and Thesis Report (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
Topics	**Learning Outcomes**
• Final structure of the thesis manuscript according to university standards • Scientific writing techniques for mathematics and computing • Citation management, bibliography, and academic integrity • Peer review and editorial correction processes	• Apply ethical principles in scientific communication [Usar] • Effectively communicate complex results [Evaluar] • Learn continuously through feedback [Usar] • Document processes professionally [Evaluar]
Readings : [**Alley1996**], [**Matthews2008**]	

Unit 4: Defense and Professional Projection (20 hours)	
Competences Expected: ABET-1,ABET-2,ABET-4,ABET-5,ABET-6	
Topics	**Learning Outcomes**
• Preparation of the public defense protocol • Technical speaking and argumentation before academic peers • Advanced data visualization for high-impact presentations • Publication plan for results in specialized forums	• Perform effectively in public defense [Evaluar] • Manage complex scientific projects [Evaluar] • Solve advanced mathematical problems [Evaluar] • Respond to criticism constructively [Usar]
Readings : [**Turabian2018**], [**Anholt2006**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY



NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Introduction to Quantum Computing
1.2 Code	:	QC381
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	• QC251. Quantum Mechanics. (8^{th} Sem) • MA381. Algebraic Structures. (9^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

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

4. GOALS

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

5. COMPETENCES

ABET-1) An ability to identify, formulate, and solve broadly defined technical or scientific problems by applying knowledge of mathematics and science and/or technical topics to areas relevant to the discipline. (Usage)

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

6. TOPICS

Unit 1: Foundational concepts (10 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Qubits and Bloch sphere representation. • Postulates of quantum mechanics. • Quantum gates (Hadamard, CNOT, Toffoli).	• Mathematically model a qubit [Usar]. • Simulate basic operations in Qiskit [Evaluar].
Readings : [**NielsenChuang**], [**QiskitTextbook**]	

Unit 2: Quantum Architectures (10 hours)

Competences Expected: ABET-1,ABET-3

Topics	**Learning Outcomes**
• Principles: – The wave-particle duality principle – The uncertainty principle in the double-slit experiment – What is a Qubit? Superposition, interference, and measurement. Photons as qubits – Systems of two qubits, Entanglement, Bell states, The No-Signaling theorem • Axioms of QM: superposition principle, measurement axiom, unitary evolution • Single qubit gates for the circuit model of quantum computation: X, Z, H • Two qubit gates and tensor products, working with matrices • The No-Cloning Theorem. The Quantum Teleportation protocol • Algorithms: – Simple quantum algorithms: Bernstein-Vazirani, Simon's algorithm – Implementing Deutsch-Josza with Mach-Zehnder Interferometers – Quantum factoring (Shor's Algorithm) – Quantum search (Grover's Algorithm) • Implementation aspects: – The physical implementation of qubits – Classical control of a Quantum Processing Unit (QPU) – Error mitigation and control, NISQ and beyond – Measurement approaches • Emerging Applications: – Post-quantum encryption – The Quantum Internet – Adiabatic quantum computation (AQC) and quantum annealing	• Discuss how a quantum object produced as a particle propagates like a wave and is detected as a particle with a probability distribution corresponding to the wave [Debate] • Discuss the quantum-level nature that is inherently probabilistic [Debate] • Express your view on entanglement that can be used to create non-classical correlations, but there is no way to use quantum entanglement to send messages faster than the speed of light [Evaluate] • Describe quantum parallelism and the role of constructive vs destructive interference in quantum algorithms given the probabilistic nature of measurement(s) [Describe] • Analyze a code snippet providing the role of quantum Fourier transform (QFT) in Shor's algorithm [Analyze] • Write a program to implement Shor's algorithm in a simulator, highlighting the classical components and aspects of Shor's algorithm [Write] • Enumerate the specifics of each qubit modality (e.g., trapped ion, superconducting, silicon spin, photonic, quantum dot, neutral atom, topological, color center, electron-on-helium) [Enumerate] • Contrast AQC with the gate model of quantum computation and the problems each is better suited to solve [Contrast]

Readings : [**QuantumHardwareBook**], [**IBMQuantumRoadmap2023**]

Unit 3: Algorithmic Strategies (12 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Quantum computing • Deutsch-Jozsa algorithm. • Grover's search (applied to combinatorial optimization). • Shor's factorization and its impact on RSA.	• Implement Grover for SAT problems [Usar].
Readings : [**PreskillNotes**], [**QiskitTextbook**]	

Unit 4: Formal Languages and Automata (10 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Quantum computation: – Postulates of quantum mechanics: ∗ State space ∗ State evolution ∗ State composition ∗ State measurement – Column vector representations of qubits – Matrix representations of quantum operations – Simple quantum gates (e.g., XNOT, CNOT) • Quantum Approximate Optimization Algorithm (QAOA). • Logistics/finance applications (Portfolio Optimization).	• For a quantum system give examples that explain the following postulates: – State Space - system state represented as a unit vector in Hilbert space – State Evolution - the use of unitary operators to evolve system state – State Composition - the use of tensor product to compose systems states – State Measurement - the probabilistic output of measuring a system state. [Familiarizarse]. • Design a QAOA for routing problems [Evaluar].
Readings : [**QMLBook**], [**Bravyi2022**]	

Unit 5: Ethics and Post-Quantum Security (6 hours)	
Competences Expected: ABET-1,ABET-3	
Topics	**Learning Outcomes**
• Post-quantum cryptography (NIST PQC Standardization). • Quantum supremacy: technical/ethical debate. • Patents and quantum geopolitics.	• Propose post-quantum migration strategies [Familiarizarse].
Readings : [**QuantumSupremacyPaper**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY

NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF MATHEMATICAL SCIENCES

SYLLABUS

1. GENERAL INFORMATION

1.1 Course name	:	Advanced Data Visualization
1.2 Code	:	CS254
1.3 Academic Program	:	Scientific Computing
1.4 Credits	:	4
1.5 Prerrequisites	:	CS253. Exploratory data analysis. (8^{th} Sem)
1.6 Type of Course	:	Elective
1.7 Area of Studies	:	Estudios Generales
1.8 Number of Weeks	:	16
1.9 Hours per week	:	2 HT; 4 HP;
1.10 Cycle	:	I
1.11 Learning modality	:	Face to face
1.12 Academic Period	:	2026-I
1.13 Semester Start Date	:	
1.14 Semester End Date	:	

2. PROFESSORS

Meetings after coordination with the professor

3. INTRODUCTION TO THE COURSE

In the Big Data era, the ability to communicate complex information visually and effectively is fundamental. This course delves into the principles of visual perception, interactive dashboard design, and the use of advanced tools for data storytelling. It focuses on transforming abstract analysis into visual representations that facilitate strategic decision-making, integrating aesthetics, functionality, and ethics in information management.

4. GOALS

- Apply principles of Gestalt psychology and visual perception to graphic design.
- Develop interactive visualizations using cutting-edge libraries.
- Design data narratives (storytelling) that facilitate the understanding of complex problems.
- Implement visualizations for multidimensional and geospatial data.
- Critically evaluate the effectiveness and ethics of visual representations.

5. COMPETENCES

ABET-3) An ability to develop and conduct experiments or test hypotheses, analyze and interpret data, and use scientific judgment to draw conclusions. (Usage)

ABET-2) An ability to formulate or design a system, process, procedure, or program for the intended purpose. (Usage)

6. TOPICS

Unit 1: Visual Perception and Design Principles (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Color theory and luminance in visualization. • Gestalt principles applied to information design. • Pre-attentive attributes and cognitive load. • Visualization criticism: Clarity vs. Decoration (Chartjunk).	• Explain how the human visual system processes graphic information [Familiarizarse]. • Select color palettes and typographies that improve readability [Usar].
Readings : [**Cairo2016**], [**Ware2020**]	

Unit 2: Interactive and Dynamic Visualization (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Introduction to D3.js and advanced grammar of graphics. • Interactions: Filtering, zoom, brushing, and linking. • Animation to show temporal changes. • Responsive design for different devices.	• Implement interactive elements that allow user exploration [Usar]. • Build functional prototypes of dynamic visualization [Evaluar].
Readings : [**Munzner2014**], [**Murray2017**]	

Unit 3: Spatial and Temporal Data Visualization (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Map projections and choropleth maps. • Bubble maps, density maps (heatmaps), and flows. • Advanced time series: Horizon charts and streamgraphs. • Network and graph visualization.	• Represent geographic data avoiding common distortions [Usar]. • Analyze complex temporal patterns using specialized charts [Evaluar].
Readings : [**Wilke2019**], [**Cairo2016**]	

Unit 4: Multivariate and High-Dimensional Data (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Parallel coordinates and radar charts. • Scatterplot matrices (SPLOM). • Projection techniques: t-SNE and UMAP for visualization. • Hierarchy visualization: Treemaps and sunbursts.	• Reduce dimensionality to visualize hidden structures [Usar]. • Design representations for large-scale hierarchical data [Evaluar].
Readings : [**Munzner2014**], [**Wilke2019**]	

Unit 5: Dashboard Design and Business Intelligence (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Information architecture in control dashboards. • Key Performance Indicators (KPI) and visual alerts. • BI Tools: Tableau, PowerBI, and their extensibility. • Usability and user experience (UX) in visualization.	• Integrate multiple visualizations into a coherent dashboard [Usar]. • Evaluate the effectiveness of a dashboard through usability testing [Evaluar].
Readings : [**Few2013**], [**Cairo2019**]	

Unit 6: Data Storytelling and Communication (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Narrative structures for data: Scrollytelling. • Noise removal and emphasis on the main message. • Annotations and visual guidance for the viewer. • The role of the analyst as a scientific communicator.	• Create a compelling story from statistical findings [Usar]. • Design visual presentations for non-technical audiences [Evaluar].
Readings : [**Knaflic2015**], [**Cairo2019**]	

Unit 7: Ethics and Visual Deception (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Misleading charts and axis manipulation. • Cognitive biases in visual interpretation. • Accessibility: Design for color blindness and functional diversity. • Responsibility in representing sensitive data.	• Identify and correct visualizations that distort reality [Evaluar]. • Implement accessibility standards in visual products [Usar].
Readings : [**Cairo2019**], [**Few2013**]	

Unit 8: Final Project: Visualization Portfolio (8 hours)	
Competences Expected: ABET-2,ABET-3	
Topics	**Learning Outcomes**
• Selection of a complex and real-world dataset. • Iterative design process and feedback. • Technical documentation and aesthetic justification. • Project defense before a specialized jury.	• Build a complete professional-level visualization solution [Evaluar]. • Support technical and design decisions based on theory [Evaluar].
Readings : [**Knaflic2015**], [**Murray2017**]	

7. WORKPLAN

7.1 Methodology

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

7.2 Theory Sessions

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

7.3 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.

8. EVALUATION SYSTEM

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

9. BASIC BIBLIOGRAPHY