National University of Engineering (UNI)

School of Sciences

School of Computer Science

Book of short descriptions by course

– 2026-I–

Lima: August 11, 2026

Equipo de Trabajo

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

Dr. Carlos Javier Solano Salinas
Director del Departamento de Ciencia de la Computación
e-mail: *jsolano@uni.edu.pe*

Dr. José Luis Segovia-Juárez
PhD en Ciencia de la Computación. Wayne State University (Detroit, USA) (2001)
Profesor del Departamento de Ciencia de la Computacion. UNI
email: *jsegoviaj@uni.edu.pe*
http://www.tecnociencia9.com/jlsegovia/es

Dr. Marcos Antonio Alania Vicente
PhD en Ciencias Universidad de Amberes. Bélgica (2017)
Profesor del Departamento de Ciencia de la Computacion. UNI
Desarrollador de algoritmos de IA para procesos complejos interdisiplinarios
email: *alania.vicente@gmail.com*

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

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

Cesar Martin Cruz-Salazar
Mag(c) en Automática e Instrumentación, UNI
Profesor de Ciencia de la Computación, UNI, Lima-Perú
email: *ccruz@uni.edu.pe*

Prof. Miguel Arrunategui Angulo
Docente del Departamento de Ciencia de la Computación
email: *garrunateguia@uni.edu.pe*

Mag. Yuri Ccoicca
Departmento de Ciencia de la Computación, UNI, Lima-Perú
email: *yccoicca@uni.edu.pe*

M.Sc. Raul Huarote Zegarra
Mag en Ciencia de la Computación, UNT (2014)
Profesor del Departamento de Ciencia de la Computación, UNI, Lima-Perú
email: *rhuarotez@uni.edu.pe*

M.Sc. Carlos Alberto Bazán Cabanillas
Magister en Ciencias, Mención matemáticas, UNI 1980
Licenciado en Matemáticas, UNI 1977
email: *cbazanc@uni.edu.pe*

Dr. 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*

Contents

1.1 BIC01. Introduction to Computing

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.

1.2 BMA101. Linear Algebra

Linear algebra is a fundamental mathematical tool shared by computer science and engineering disciplines, providing the foundations for structural analysis, circuit networks, control systems, data processing, and algorithm design. This course provides a solid foundation in the concepts and techniques of elementary and advanced linear algebra, with emphasis on its application in scientific and engineering contexts.

1.3 BMA102. Differential Calculus

Differential calculus is a fundamental mathematical tool shared by computer science and engineering disciplines, providing the foundations for the rigorous study of change, the analysis of rates of change, the optimization of systems and functions, and the modeling of continuous phenomena. This course provides a solid foundation in the concepts and techniques of differential calculus of one variable, with emphasis on its application in scientific and engineering contexts.

1.4 BCH101. Chemistry I

Chemistry provides a foundation for understanding the composition, structure, and properties of matter, essential for engineering disciplines in areas such as materials science, nanotechnology, and industrial processes. This course introduces the basic principles of general chemistry.

1.5 FG001. General Elective

General Education Elective

1.6 BEI101. Technical and professional English I

English is the primary language of science, engineering, and technology worldwide. For computing and engineering students, mastering English means gaining direct access to technical documentation, research papers, and professional collaboration without translation barriers. This course builds foundational English skills using contexts drawn from computing systems, hardware, and the everyday work of engineers and scientists.

2.1 CS100. Introduction to Computer Science

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.

2.2 CS112. Objects-oriented programming I

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.

2.3 CS1D1. Discrete Structures

Discrete structures provide the theoretical foundation for computer science. This course introduces the essential mathematical tools used to model computational problems, including logic, set theory, and functions. Understanding these concepts is critical for developing rigorous reasoning skills and for the subsequent study of algorithms, databases, and formal languages.

2.4 BMA103. Integral Calculus

Integral calculus is a fundamental mathematical tool shared by computer science and engineering disciplines, essential for modeling and solving problems involving accumulation, change, and areas under curves. This course provides the foundations of integral calculus, including integration techniques and their applications in scientific and engineering contexts.

2.5 BFI101. Physics I

Physics I introduces the foundations of classical mechanics as a mathematical and experimental discipline. Through the study of kinematics, Newtonian

dynamics, conservation principles, and rotational motion, students acquire the analytical tools needed to model and solve problems of the physical world. This course provides the indispensable foundation for the study of oscillatory, wave, and electromagnetic phenomena addressed in subsequent courses.

2.6 BEI102. Technical and professional English II

Building on foundational English skills, this course develops the student's ability to communicate in technical and scientific contexts with greater fluency. Grammar structures are practiced through reading and producing descriptions of technological systems, scientific phenomena, and engineering processes. Students gain vocabulary relevant to computing, data, and the physical sciences.

3.1 CS113. Objects-oriented programming II

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.

3.2 CS2B1. Platform Based Development

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.

3.3 AI161. Applied AI

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.

3.4 BMA104. Advanced Differential and Integral Calculus

Advanced differential and integral calculus provides the rigorous foundations for understanding multidimensional phenomena in engineering and computer science. This course covers vector functions, multivariable calculus, differentiable transformations, multiple integrals, and integral calculus on manifolds, including the fundamental theorems of Green, Stokes, and Gauss. These tools are essential for modeling complex systems, optimizing multivariable functions, and solving problems in computer graphics, computer vision, and physical simulations.

3.5 STA251. Probability Calculation

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

3.6 BEI201. Technical and professional English III

At this level, students engage with English as the primary medium for accessing and producing technical content in computing, engineering, and the sciences. Grammar structures are applied to reading scientific articles, writing technical reports, describing engineering processes, and participating in professional discussions. The course bridges everyday English fluency with the demands of technical communication.

4.1 CS210. Algorithms and Data Structures

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.

4.2 CS211. Theory of Computation

Theory of Computation provides the mathematical foundations to understand what can be computed and how efficiently. It introduces formal languages, automata, and the limits of algorithmic solvability. This knowledge is fundamental

for understanding compiler construction, formal verification, and the inherent complexity of computational problems.

4.3 CS221. Computer Systems Architecture

A deep understanding of computer architecture is essential for developing high-performance software and understanding the hardware-software interface. This course explores how computers execute instructions, manage memory, and communicate with external devices. By studying the internal organization of processors and the hierarchy of memory systems, students gain the ability to optimize code for specific hardware architectures and understand the trade-offs in modern system design.

4.4 CS271. Data Management

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.

4.5 CS401. Research Methodology

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

4.6 MA202. Numerical Methods

Numerical methods are a fundamental tool shared by computer science and engineering disciplines for approximating solutions to mathematical problems that cannot be solved analytically. This course provides an introduction to the most common numerical methods, including error analysis, solution of nonlinear equations, interpolation, numerical integration, and solution of differential equations, in scientific and engineering contexts.

4.7 BEI202. Technical and professional English IV

At the intermediate level, students apply English to the full range of technical communication required in computing and engineering practice: reading research papers, writing structured technical documents, analyzing case studies, and engaging in professional dialogue. Grammar structures are taught through

authentic computing and engineering contexts, reinforcing both linguistic competence and disciplinary knowledge.

5.1 CS212. Analysis and Design of Algorithms

Algorithms are the heart of computer science. This course provides the mathematical and analytical tools necessary to evaluate the efficiency of algorithms in terms of time and space. Students will explore fundamental data structures, complexity analysis, and various design paradigms—such as brute force, greedy, and dynamic programming. Understanding these techniques is essential for building software that is not only functional but also scalable and resource-efficient.

5.2 CS261. Artificial Intelligence

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.

5.3 CS272. Databases II

This course is a continuation of Data Management I, focusing on the internal mechanisms of database management systems (DBMS) and modern database technologies. It covers query optimization, indexing structures, and transactional integrity in depth. Additionally, it introduces students to non-relational models (NoSQL) and distributed architectures, which are essential for handling large-scale data in contemporary cloud environments.

5.4 CS291. Software Engineering I

Software engineering is a collaborative discipline that applies structured processes to build high-quality systems. This first course focuses on the foundational aspects of the software lifecycle, from initial requirements and human-centric design to systematic testing. By integrating teamwork principles and rigorous development fundamentals, students will learn to create software that is both technically sound and user-oriented.

5.5 CS2S1. Operating systems

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.

5.6 BEI203. Technical and professional English V

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. In the engineering and computing career, technical English is the most critical tool for accessing the forefront of knowledge and research, writing scientific papers, participating in international conferences, and collaborating in global teams. This course brings students to the level of autonomous technical communicators capable of producing and critically evaluating academic and professional content.

6.1 CS231. Networking and Communication

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.

6.2 CS311. Algorithms for Complex Problems

Competitive Programming trains students to solve complex algorithmic problems under time and resource constraints. This course bridges the gap between theoretical algorithm design and high-performance implementation. It covers advanced techniques in dynamic programming, graph theory, and mathematics, fostering the analytical thinking required for top-tier software engineering interviews and international programming contests like the ICPC.

6.3 CS312. Advanced Data Structures

Algorithms and data structures are a fundamental part of Computer Science that allow us to organize information efficiently. For any professional in the field, a solid foundation in this area is crucial. This course focuses on advanced structures such as Multidimensional Access Methods, Spatio-Temporal Access Methods, and Metric Access Methods, which are essential for high-performance applications in search engines, geographic information systems (GIS), and big data processing.

6.4 CS342. Compilers

This course provides a deep understanding of how high-level languages are transformed into executable machine code. The study of compilers allows students to understand language semantics, memory management, and code optimization, which are fundamental skills for designing domain-specific languages and high-performance software engineering.

6.5 AI263. Introduction to Machine Learning

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.

6.6 FI201. Computational Physics

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

7.1 CS251. Computer Graphics

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.

7.2 CS292. Software Engineering II

Software Engineering II builds upon the foundational principles of the first course, shifting focus towards advanced design, software evolution, and the management of large-scale projects. It explores the complexities of maintaining systems, ensuring software quality through rigorous validation, and the application of professional teamwork and project management techniques in collaborative environments.

7.3 CS2H1. User Experience (UX)

This course focuses on the design, evaluation, and implementation of interactive computing systems for human use. Beyond classical HCI principles, it addresses the emerging challenges of interacting with AI systems, intelligent agents, and natural language processing software. Students will develop competencies in user-centered design, accessibility, ethical accountability, and the evaluation of

both traditional interfaces and AI-driven conversational systems, acquiring skills essential for designing responsible and human-centered technology.

7.4 AI264. Deep Learning

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

7.5 FG211. Professional Ethics

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

8.1 CS281. Computing in Society

This course explores the multifaceted relationship between computing technologies and society. It provides a historical perspective on the evolution of the field while equipping students with ethical analysis frameworks to navigate contemporary challenges. Topics include intellectual property, privacy, professional responsibility, sustainability, and the socio-economic impact of automation. Students will develop the communication skills and ethical sensitivity required for professional practice in a diverse and globalized environment.

8.2 CS3I1. Computer Security

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.

8.3 CS3P1. Parallel and Distributed Computing

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.

8.4 CS402. Capstone Project I

This course aims to enable students to conduct a deep state-of-the-art study on a specific topic chosen for their thesis project. Students will produce a technical report with preliminary comparative experiments and formulate a research proposal, culminating this part alongside with a faculty advisor who specializes in the chosen research area.

8.5 FG106. Theater

Theater as a comprehensive performing art invites the student to recognize themselves as a creative and social human being. Through theatrical practice, the student develops their physical, verbal, and emotional expression; strengthens their self-confidence and personal security; and cultivates communication, empathy, and teamwork skills. This course provides practical tools for artistic expression that contribute to the comprehensive education of the university student, stimulating their sensitivity, creativity, and imagination. At the same time, it integrates them into their academic community through shared experiential activities, opening paths of self-discovery and authentic communication with others.

8.6 EX301. Extracurricular Activities

This course records the student's participation in activities that complement their academic training, such as workshops, conferences, social outreach, volunteering, or technical skills. Its objective is to promote comprehensive development, soft skills, and social commitment.

8.7 AI268. Computational Vision

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.

8.8 CS391. Software Engineering III

Software Engineering III focuses on advanced topics of the software lifecycle, emphasizing software quality, evolution, and deployment in modern architectures. This course prepares students to handle the complexities of maintaining systems, ensuring software quality through rigorous validation, and the application of professional tools and design practices for resilient and maintainable software products.

8.9 CS393. Information systems

This course provides a comprehensive overview of how information systems support business processes and decision-making within organizations. It covers the strategic role of information technology, the lifecycle of information systems, data management strategies, and the integration of enterprise applications to achieve competitive advantages in a globalized digital economy.

8.10 CS394. Software Architecture and DevOps for Cloud Software

Modern software design requires combining solid architectural principles with agile continuous delivery practices and cloud operations. This course integrates three areas that are inseparable in today's industry: software architecture, DevOps, and cloud computing.

Students will learn to design systems at scale using well-established architectural patterns, to automate build, test, and deployment cycles through CI/CD pipelines, and to provision and operate infrastructure on public cloud platforms using containers, orchestration, and infrastructure-as-code tools.

The course places special emphasis on the reliability and observability of distributed systems, covering metrics, distributed traces, error budgets, and site reliability engineering (SRE) practices as applied in high-performing technology organizations.

9.1 CS370. Big Data

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.

9.2 CS400. Pre-professional internships

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.

9.3 CS403. Capstone Project II

This course aims to enable students to develop their research proposal and execute formal experiments, producing the core chapters of their thesis under the supervision of their faculty advisor.

9.4 AI365. Advanced Generative AI Models

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

9.5 CB309. Bioinformatics

The use of computational methods in the biological sciences has become one of the key tools for the field of molecular biology, being a fundamental part of research in this area.
In Molecular Biology, there are several applications that involve both DNA, protein analysis or sequencing of the human genome, which depend on computational methods. Many of these problems are really complex and deal with large data sets.
This course can be used to see concrete use cases of several areas of knowledge of Computer Science such as Programming Languages (PL), Algorithms and Complexity (AL), Probabilities and Statistics, Information Management (IM), Intelligent Systems (IS).

9.6 AI369. Robotics

Robotics is the physical manifestation of Artificial Intelligence. This course introduces the fundamental concepts of robot design, including kinematics, perception, and control. Students will explore how robots interact with the physical world through sensors and actuators, and how they make autonomous decisions using probabilistic mapping and localization techniques. The course bridges the gap between abstract algorithms and physical execution.

9.7 CS351. Topics in Computer Graphics

Modern computer graphics go beyond simple image generation, encompassing real-time interaction, physical simulation, and immersive environments. This course explores advanced topics in the graphics pipeline, focusing on programmable shading, complex geometric modeling, and computer animation. Additionally, it addresses the intersection of graphics with physical computing and the ethical implications of synthetic content generation.

9.8 CS353. Quantum Computing

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.

9.9 CS392. Tópicos en Ingeniería de Software

Modern software development requires rigorous engineering practices to ensure quality, maintainability, and reliability. This course explores advanced topics in the software lifecycle, focusing on team collaboration, formal verification, and advanced refactoring. Students will master the use of professional tools to design, construct, and validate complex systems while ensuring high standards of reliability and formal correctness.

9.10 CS3P3. Internet of Things

The Internet of Things (IoT) has transformed how we interact with the physical world by connecting billions of devices. This course explores the architectural foundations of IoT, emphasizing device mobility, wireless communication, and the societal implications of pervasive sensing. Students will analyze the technical challenges of constrained systems and the ethical responsibilities associated with large-scale data collection in interconnected environments.

10.1 CS3P2. Cloud Computing

Modern computing relies heavily on distributed architectures and cloud-native platforms. This course provides an in-depth exploration of cloud computing services, distributed database concepts, and web platforms. Students will learn to design scalable architectures, understand the implications of distributed data consistency, and evaluate the privacy and security challenges inherent in cloud-based environments.

10.2 CS404. Research Workshop

This course aims to enable students to complete their thesis with sufficient quality for public defense, finishing experiments, final chapters, advisor revisions, and the formal institutional submission.

10.3 AI367. Topics in Artificial Intelligence

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.

10.4 AI368. Evolutionary Computing

Evolutionary Computing is a subfield of Artificial Intelligence inspired by natural selection and biological evolution. This course explores metaheuristic optimization algorithms such as Genetic Algorithms, Evolution Strategies, and

Genetic Programming. These techniques are essential for solving complex optimization and search problems where traditional analytical methods are insufficient. Students will learn to design, implement, and analyze evolutionary systems applied to engineering, data science, and autonomous agents.

10.5 FG350. Leadership and Performance

Contemporary organizations demand from their members the effective exercise of leadership: to assume with effectiveness and a spirit of service the challenges entrusted to them, in the pursuit of a more just and reconciled society. This challenge involves forming people with a sound knowledge of themselves, with the ability to judge reality objectively and to propose guidelines that positively modify the environment. The course combines theoretical foundations with the experiential development of competencies, training the student to lead teams with integrity, personal coherence, and genuine service to others.