Hybrid Professional Master's Degree

Advanced Software Engineering

tech global university

Hybrid Professional Master's Degree
Advanced Software Engineering

Modality: **Hybrid (Online + Internship)**
Duration: **12 months**
Certificate: **TECH Global University**
Accreditation: **60 + 4 ECTS**
Website: **www.techtitute.com/us/information-technology/hybrid-professional-master-degree/hybrid-professional-master-degree-advanced-software-engineering**

Index

01 Introduction

Advanced Software Engineering has emerged as a crucial component in the development of complex and robust systems that are fundamental to the modern digital economy. In an environment where the demand for high quality and scalable software is constantly increasing, advanced software engineering techniques enable developers to create efficient and sustainable solutions. It is therefore essential that IT professionals update their knowledge regularly to incorporate in their daily practice the most innovative methodologies to improve software quality and accelerate the development cycle. In this context, TECH launches a revolutionary university degree focused on the most cutting-edge techniques of Advanced Software Engineering.

With this Hybrid Professional Master's Degree, you will design the most scalable, robust and maintainable software systems"

Software quality and security have become critical elements in Advanced Software Engineering. In this regard, a report by the Organization for Economic Cooperation and Development reveals that 58% of companies experience security failures due to the lack of adequate quality assurance practices. This can have serious consequences, ranging from loss of revenue or lack of customer confidence to significant fines. Hence the importance of professionals incorporating the following strategies into their practice.

Against this backdrop, TECH presents an innovative Hybrid Professional Master's Degree in Advanced Software Engineering. It is a program that combines in 1,920 hours of the best theoretical content with 3 weeks of practical stay in a leading company in this field. Made up of 10 complete modules, the academic itinerary will delve into subjects such as Requirements Engineering, Web Application Design or Information Systems Auditing. In addition, the syllabus will provide students with state-of-the-art techniques for system protection and the development of highly secure codes. In this way, graduates will acquire advanced skills to analyze complex software problems and develop efficient and innovative solutions.

On the other hand, this university degree foresees that graduates develop a practical internship in a prestigious institution in the field of Advanced Software Engineering. Therefore, students will actively participate in the projects that are being developed at that time. It should be noted that a specialized tutor will guide students during the academic experience, guaranteeing the completion of a plan of activities that will allow them to improve their skills in an exponential way and based on the requirements of the current demand of the labor market.

This **Hybrid Professional Master's Degree in Advanced Software Engineering** contains the most complete and up-to-date program on the market. The most important features include:

- Development of more than 100 case studies presented by experts in Computer Engineering
- Its graphic, schematic and practical contents provide essential information on those disciplines that are indispensable for professional practice
- Practical exercises where the self-assessment process can be carried out to improve learning
- Its special emphasis on innovative methodologies
- All of this will be complemented by theoretical lessons, questions to the expert, debate forums on controversial topics, and individual reflection assignments
- Content that is accessible from any fixed or portable device with an Internet connection
- Furthermore, you will be able to carry out an internship in one of the best companies

> *You will lead Software Development projects, from conception to implementation and evaluation"*

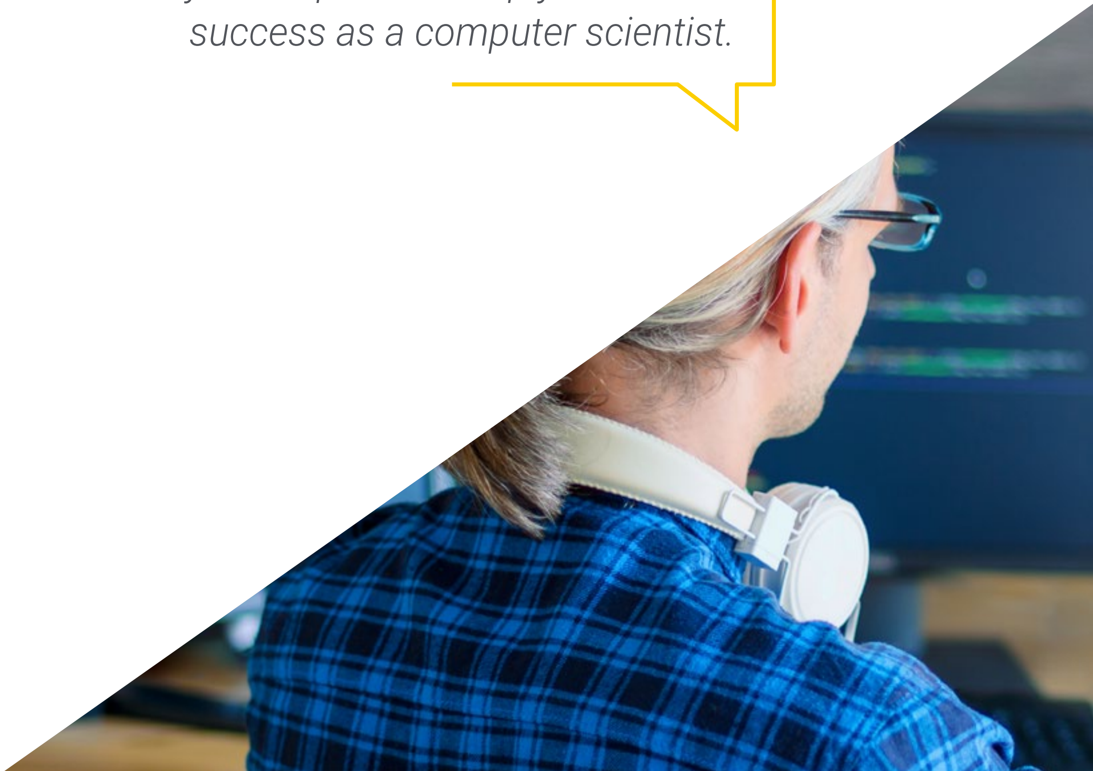
> *You will spend an intensive 3-week internship in a reference institution in the field of Advanced Software Engineering"*

Update your knowledge in Advanced Software Engineering through innovative multimedia content.

TECH is a technological vanguard university that puts all its resources at your disposal to help you achieve success as a computer scientist.

In this Hybrid Professional Master's Degree proposal, of professionalizing character and blended learning modality, the program is aimed at updating Computer Science professionals. The contents are based on the latest scientific evidence, and oriented in a didactic way to integrate theoretical knowledge into practice, and the theoretical-practical elements will facilitate the updating of knowledge.

Thanks to its multimedia content elaborated with the latest educational technology, it will allow the IT professional a situated and contextual learning, that is to say, a simulated environment that will provide an immersive learning programmed to specialize in real situations. This program is designed around Problem-Based Learning, whereby the physician must try to solve the different professional practice situations that arise during the course. For this purpose, the students will be assisted by an innovative interactive video system created by renowned and experienced experts.



02

Why Study this Hybrid Professional Master's Degree?

One of the main priorities of companies is to innovate and maintain their competitiveness in a global market. For this reason, organizations are looking to incorporate highly specialized computer scientists in Advanced Software Engineering, capable of developing new applications that provide them with competitive advantages. To take advantage of these opportunities, professionals need to keep abreast of the latest developments in the field. With this in mind, TECH has created a unique and disruptive academic program in the current educational landscape, which will allow specialists to immerse themselves in a real working environment and apply the latest procedures and techniques in Advanced Software Engineering.

> *You will perform your practical stay in a reference entity, where you will analyze the latest advances in Advanced Software Engineering"*

1. Updating from the latest technology available

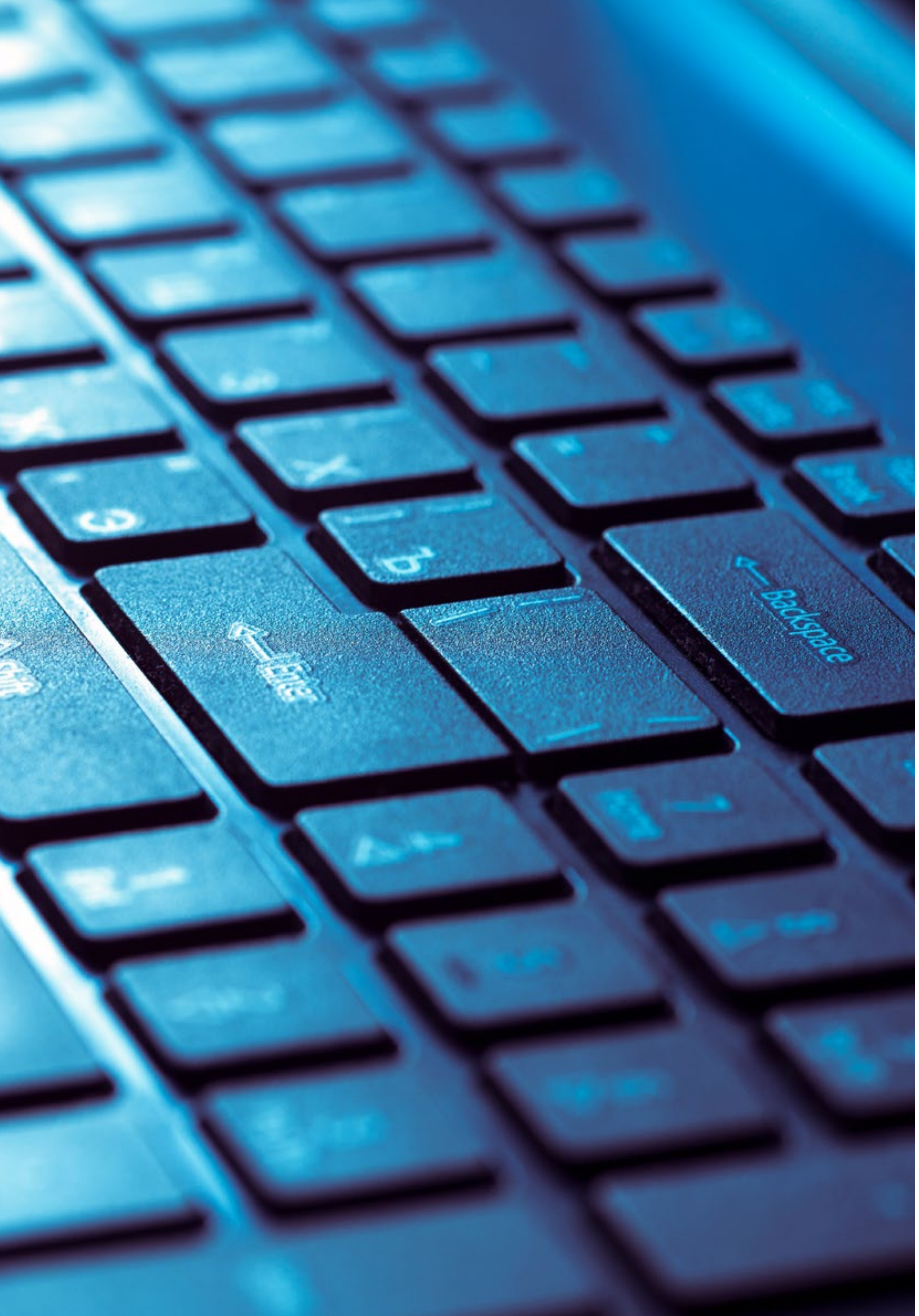
New technologies have significantly transformed the field of Advanced Software Engineering, increasing efficiency, quality and innovativeness in software development. One example of this is automated testing, which allows professionals to quickly detect and correct errors. With the purpose of familiarizing specialists with these tools, TECH offers this Internship Program, allowing students to immerse themselves in a cutting-edge work environment, where they will have access to state-of-the-art technology in this field.

2. Gaining in-depth knowledge from the experience of top specialists

Throughout the Internship Program, a team of experts in Advanced Software Engineering will guide students, helping them to make the most of this academic experience. In this sense, these professionals will transmit to the graduates all the knowledge they need to propel their career as computer scientists to the top.

3. Entering first-class professional environments

TECH's main objective is to offer high quality university programs that are accessible to everyone. For this reason, TECH carefully selects the centers where graduates will do their internships. This ensures that computer scientists have access to leading institutions in the field of Advanced Software Engineering. Therefore, graduates will be able to experience firsthand the daily work in a demanding and rigorous environment, always applying the most advanced techniques and methodologies.



4. Combining the best theory with state-of-the-art practice

In today's academic landscape, it is common to find university programs that only offer theoretical content, leaving aside the importance of practice so that students can apply their knowledge in real work situations. In contrast, TECH presents a completely practical learning model, which allows IT professionals to gain practical experience and face real challenges that may arise in their professional career.

5. Expanding the boundaries of knowledge

TECH offers graduates the opportunity to pursue this Hybrid Professional Master's Degree in prestigious international institutions. This allows computer scientists to broaden their horizons and update their knowledge with the best professionals working in top-level companies. It is an exceptional opportunity that only TECH, the largest digital university in the world, could provide.

> *You will have full practical immersion at the center of your choice"*

03

Objectives

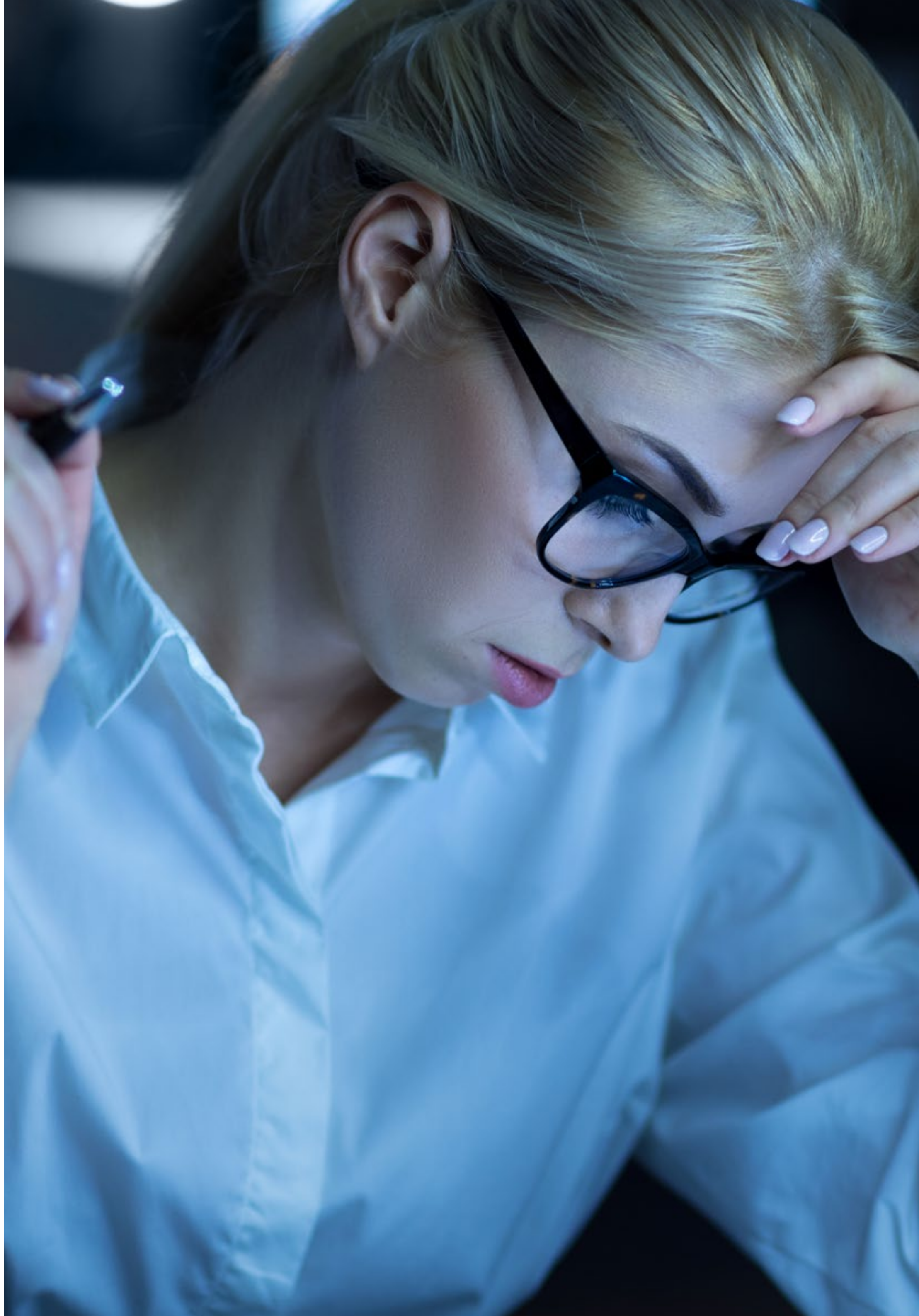
Upon completion of this Hybrid Professional Master's Degree, computer scientists will have a comprehensive understanding of the design, architecture, implementation and maintenance of complex software systems. In this sense, graduates will apply both agile and traditional software development methodologies in order to manage projects effectively. In addition, professionals will use metrics to measure and improve the quality of applications.

Incorporate into your practice the most innovative secure development techniques to protect software systems against threats and vulnerabilities"

General Objective

- This Hybrid Professional Master's Degree in Advanced Software Engineering will provide IT professionals with advanced skills to design scalable, robust and maintainable software architectures using design patterns and software architecture principles. At the same time, students will apply modeling and simulation techniques to efficiently predict and solve potential problems in software systems

> *You will learn through real cases and by solving complex situations in simulated learning environments"*



Specific Objectives

Module 1. Software Engineering

- Lay the foundations of software engineering and modeling, learning the main processes and concepts
- Understand the software process and the different models for its development including agile technologies

Module 2. Advanced Software Engineering

- Know in depth the different agile methodologies used in software engineering
- Learn to develop using scrum, extreme programming and reuse-based software development techniques
- Understand the concepts and processes of software design, learning also about architecture design and design at component level and based on patterns
- Introduce the DevOps concept and its main practices
- Learn how to test software, with methodologies such as Test-Driven Development, Acceptance Test-Driven Development, Behavior-Driven Development, BDD and Cucumber
- Understand the different patterns of system architectures and software design, as well as the architecture of cloud applications

Module 3. Requirements Engineering

- Understand requirements engineering, their development, elaboration, negotiation and validation
- Learn the modeling of requirements and the different elements such as scenarios, information, analysis classes, flow, behavior and patterns
- Understand the importance of requirements engineering in the software development process
- Learn how to perform requirements analysis, as well as how to properly document them
- Have an in-depth knowledge of the requirements sources and requirements elicitation techniques, as they are an essential part of the process
- Understand requirements validation and negotiation processes, as well as requirements modeling and management

Module 4. Software Engineering Processes

- Delve into the improvement of the software development process and software quality using ISO/IEC standards
- Understand and apply prototyping as an essential part of the development process
- Know the software engineering framework and the ISO/IEC 12207 standard
- Learn the characteristics of the unified software development process and planning in the context of agile software development
- Learn the different styles of distributed software design and service-oriented software architectures
- Learn the essential concepts in graphical user interface design

Module 5. Quality and Auditing of Information Systems

- Delve into software testing strategies and techniques, software quality factors and different metrics used
- Acquire the essential knowledge of IT security management systems
- Introduce the concepts of intellectual property in information management systems
- Prepare students in the creation of business continuity and disaster recovery plans
- Learn how to plan the management of the security and to handle the principal mechanisms for the protection of assets information
- Learn about the different types of audits and the process carried out during the IT audit

Module 6. Integration Systems

- Acquire the essential concepts related to information systems in the enterprise, as well as identify the opportunities and needs of information systems within the enterprise
- Learn the basics of Business Intelligence, its strategies and implementation, as well as the present and future of BI
- Understand the functioning of systems for integrated enterprise resource planning
- Understand digital transformation, from the point of view of business innovation, financial and production management, marketing and human resources management

Module 7. Software Reuse

- Know the big picture in software reuse strategy
- Learn the different patterns related to software reuse, both in terms of design, creation, structure and behavior
- Learn about the concept of framework, as well as to the main types such as those for graphical user interface design, web application development and object persistence management in databases
- Understand the current widely used Model View Controller (MVC) pattern

Module 8. Information Technology Services

- Get qualified in ICT investment decision making and information systems planning
- Know the control objectives for information and related technologies (COBIT)
- Learn how the Information Technology Infrastructure Library (ITIL) works, strategies, service design, transitions and operations
- Delve into the service management system, knowing the basic principles of UNE-ISO/IEC 20000-1, the structure of the ISO/IEC 20000 series of standards and the requirements of the Service Management System (SMS)
- Understand the functioning of information systems and technologies, their components, classifications, architectures and forms of system integration
- Learn the ISO/IEC 12207 standard, the analysis, design, implementation and acceptance of information systems

Module 9. Information Systems Security

- Learn schedule development for time management, budget development and risk response
- Analyze the nature of network attacks and the different types of security architectures
- Understand the various techniques of system protection and secure code development
- Know the essential components of botnets and spam, as well as malware and malicious code
- Lay the foundations for forensic analysis in the world of software and computer audits
- Understand the fundamentals of symmetric cryptography and asymmetric cryptography, as well as their main algorithms

Module 10. Project Management

- Understand how quality management works in projects, including planning, assurance, control, statistical concepts and available tools
- Understand the functioning of the processes of procurement, execution, monitoring, control and closure of a project
- Acquire the essential knowledge related to the professional responsibility derived from project management
- Know the fundamental concepts of project management and the project management life cycle

04

Skills

Through this university degree, IT professionals will acquire advanced skills to design scalable, maintainable and robust software systems using design architecture principles. In line with this, graduates will be highly qualified to apply agile methodologies (such as Scrum or Kanban) in the software development life cycle. In addition, experts will implement quality assurance processes in order to guarantee the quality of the applications through code reviews, audits and thorough testing.

> *"You will use Frameworks to maintain consistency in Software design and implementation"*

General Skills

- Respond to the current needs of the Advanced Software Engineering area
- Master the different working systems in Advanced Software Engineering

> *You will have at your disposal a library full of educational resources available 24 hours a day and with material that stands out for its quality"*

Specific Skills

- Develop in-depth knowledge of all facets of human-computer interaction and how they involve computer developments
- Be proficient in the use of databases
- Develop different types of network applications
- Work as a software engineer
- Control the use of advanced databases
- Perform advanced programming
- Know how to reuse software
- Create interfaces and network applications

> *You will combine theory and professional practice through a demanding and rewarding educational approach"*

05

Educational Plan

The teaching materials that make up this Hybrid Professional Master's Degree have been prepared by a team of professionals highly specialized in Advanced Software Engineering. Therefore, they have developed a syllabus that stands out both for its excellent quality and for being in line with the requirements of today's labor market. Composed of 10 specialized modules, the curriculum will delve into the latest innovations in subjects such as Requirements Modeling, Software Architecture or Graphical User Interface design. In addition, the program will provide graduates with the most cutting-edge techniques for software maintenance.

```
f.deactivate(true);
urn; // this is a modified line

.keyCode == 13) {
topImmediatePropagation();
reventDefault();
f.search();
f.deactivate();
urn;

down
.keyCode == 38 || e.keyCode == 40) {
reventDefault();
topImmediatePropagation();
(e.keyCode == 38) { // up
/ show previous search que
f (hist.currentIndex ==
 hist.temporaryQuer
 // skip previo
 if (hist.te
   hist.
}
```

> *You will master Agile Methodologies to improve the efficiency, flexibility and responsiveness of development teams to changes"*

Module 1. Software Engineering

1.1. Introduction to Software Engineering and Modeling
- 1.1.1. The Nature of Software
- 1.1.2. The Unique Nature of Webapps
- 1.1.3. Software Engineering
- 1.1.4. The Software Process
- 1.1.5. Software Engineering Practice
- 1.1.6. Software Myths
- 1.1.7. How It All Begins
- 1.1.8. Object-Oriented Concepts
- 1.1.9. Introduction to UML

1.2. The Software Process
- 1.2.1. A General Process Model
- 1.2.2. Prescriptive Process Models
- 1.2.3. Specialized Process Models
- 1.2.4. The Unified Process
- 1.2.5. Personal and Team Process Models
- 1.2.6. What is Agility?
- 1.2.7. What is an Agile Process?
- 1.2.8. Scrum
- 1.2.9. Agile Process Toolkit

1.3. Software Engineering Guiding Principles
- 1.3.1. Principles Guiding the Process
- 1.3.2. Principles Guiding the Practice
- 1.3.3. Principles of Communication
- 1.3.4. Planning Principles
- 1.3.5. Modeling Principles
- 1.3.6. Construction Principles
- 1.3.7. Deployment Principles

```
(settings[0].compareTo("s") ==
  if (name.compareTo("") != 0)
    name += "_";
  }
  name += etr.getString(s
} else if (settings[0].co
  if (name.compareTo("
    name += "_";
  }
  name += DateUti
} else if (settin
  if (name.com
    name +=
```

1.4. Understanding the Requirements
- 1.4.1. Requirements Engineering
- 1.4.2. Establish the Basis
- 1.4.3. Inquiry of Requirements
- 1.4.4. Development of Cases Studies
- 1.4.5. Elaboration of the Requirements Model
- 1.4.6. Negotiation of Requirements
- 1.4.7. Validation of Requirements

1.5. Requirements Modeling I: Scenarios, Information and Analysis Classes
- 1.5.1. Analysis of Requirements
- 1.5.2. Scenario-Based Modeling
- 1.5.3. UML Models that provide the Case Study
- 1.5.4. Data Modeling Concepts
- 1.5.5. Class-Based Modeling
- 1.5.6. Class Diagrams

1.6. Requirements Modeling II: Flow, Behavior and Patterns
- 1.6.1. Requirements that Shape Strategies
- 1.6.2. Flow-Oriented Modeling
- 1.6.3. Status Diagrams
- 1.6.4. Creation of a Behavioral Model
- 1.6.5. Sequence Diagrams
- 1.6.6. Communication Diagrams
- 1.6.7. Patterns for Requirements Modeling

1.7. Design Concepts
- 1.7.1. Design in Software Engineering
- 1.7.2. The Design Process
- 1.7.3. Design Concepts
- 1.7.4. Object-Oriented Design Concepts
- 1.7.5. Model of the Design

1.8. Designing the Architecture
- 1.8.1. Software Architecture
- 1.8.2. Architectural Genres
- 1.8.3. Architectural Styles
- 1.8.4. Architectural Design
- 1.8.5. Evolution of Alternative Designs for Architecture
- 1.8.6. Mapping the Architecture Using the Data Flow

1.9. Component-Level and Pattern-Based Design
- 1.9.1. What is a Component?
- 1.9.2. Class-Based Component Design
- 1.9.3. Realization of the Design at the Component Level
- 1.9.4. Design of Traditional Components
- 1.9.5. Component-Based Development
- 1.9.6. Design Patterns
- 1.9.7. Pattern-Based Software Design
- 1.9.8. Architectural Patterns
- 1.9.9. Design Patterns at the Component Level
- 1.9.10 User Interface Design Patterns

1.10. Software Quality and Project Management
- 1.10.1. Quality
- 1.10.2. Software Quality
- 1.10.3. The Software Quality Dilemma
- 1.10.4. Achieving Software Quality
- 1.10.5. Software Quality Assurance
- 1.10.6. The Administrative Spectrum
- 1.10.7. The Staff
- 1.10.8. The product
- 1.10.9. The Process
- 1.10.10. The Project
- 1.10.11. Principles and Practices

Module 2. Advanced Software Engineering

2.1. Introduction to Agile Methodologies
- 2.1.1. Process Models and Methodologies
- 2.1.2. Agility and Agile Processes
- 2.1.3. Agile Manifesto
- 2.1.4. Some Agile Methodologies
- 2.1.5. Agile vs. Traditional

2.2. Scrum
- 2.2.1. Origins and Philosophy of Scrum
- 2.2.2. Scrum Values
- 2.2.3. Scrum Process Flow
- 2.2.4. Scrum Roles
- 2.2.5. Scrum Artifacts
- 2.2.6. Scrum Events
- 2.2.7. User Stories
- 2.2.8. Scrum Extensions
- 2.2.9. Agile Estimates
- 2.2.10 Scrum Scaling

2.3. Extreme Programming
- 2.3.1. Justification and Overview of XP
- 2.3.2. The XP Life Cycle
- 2.3.3. The Five Core Values
- 2.3.4. The Twelve Basic Practices in XP
- 2.3.5. Roles of Participants
- 2.3.6. XP Industrial
- 2.3.7. Critical Assessment of XP

2.4. Software Development Based on Reusability
- 2.4.1. Software Reuse
- 2.4.2. Code Reuse Levels
- 2.4.3. Specific Reuse Techniques
- 2.4.4. Component-Based Development
- 2.4.5. Benefits and Problems of Reuse
- 2.4.6. Reuse Planning

2.5. System Architecture and Software Design Patterns
- 2.5.1. Architectural Design
- 2.5.2. General Architectural Patterns
- 2.5.3. Fault Tolerant Architectures
- 2.5.4. Distributed Systems Architectures
- 2.5.5. Design Patterns
- 2.5.6. Gamma Patterns
- 2.5.7. Interaction Design Patterns

2.6. Cloud Application Architecture
- 2.6.1. Cloud Computing Fundamentals
- 2.6.2. Cloud Application Quality
- 2.6.3. Architectural Styles
- 2.6.4. Design Patterns

2.7. Software Testing: TDD, ATDD and BDD
- 2.7.1. Software Verification and Validation
- 2.7.2. Software Testing
- 2.7.3. Test Driven Development (TDD)
- 2.7.4. Acceptance Test Driven Development (ATDD)
- 2.7.5. Behavior Driven Development (BDD)
- 2.7.6. BDD and Cucumber

2.8. Software Process Improvement
- 2.8.1. Software Process Improvement
- 2.8.2. The Process Improvement Approach
- 2.8.3. Maturity Models
- 2.8.4. The CMMI Model
- 2.8.5. CMMI V2.0
- 2.8.6. CMMI and Agile

2.9. The Quality of the Software Product: SQuaRE
- 2.9.1. Software Quality
- 2.9.2. Software Product Quality Models
- 2.9.3. ISO/IEC 25000 Family
- 2.9.4. ISO/IEC 25010: Quality Model and Quality Characteristics
- 2.9.5. ISO/IEC 25012: the Quality of the Data
- 2.9.6. ISO/IEC 25020 Software Quality Measurement.
- 2.9.7. ISO/IEC 25022, 25023 and 25024: Software and Data Quality Metrics
- 2.9.8. ISO/IEC 25040 Software Assessment
- 2.9.9. Accreditation Process

2.10. Introduction to DevOps
- 2.10.1. DevOps Concept
- 2.10.2. Core Practices

Module 3. Requirements Engineering

3.1. Introduction to Requirements Engineering
- 3.1.1. The Importance of Requirements
- 3.1.2. Concept of Requirement
- 3.1.3. Dimensions of Requirements
- 3.1.4. Levels and Types of Requirements
- 3.1.5. Requirements Characteristics
- 3.1.6. Requirements Engineering
- 3.1.7. The Requirements Engineering Process
- 3.1.8. Frameworks for Requirements Engineering
- 3.1.9. Best Practices in Requirements Engineering
- 3.1.10 The Business Analyst

3.2. Sources of Requirements
- 3.2.1. The Requirements Network
- 3.2.2. The Stakeholders
- 3.2.3. Business Requirements
- 3.2.4. Vision and Scope Document

3.3. Requirements Elicitation Techniques

- 3.3.1. Elicitation of Requirements
- 3.3.2. Problems of Requirements Elicitation
- 3.3.3. Contexts of Discovery
- 3.3.4. Interviews
- 3.3.5. Observation and Learning
- 3.3.6. Ethnography
- 3.3.7. Workshops
- 3.3.8. Focus groups
- 3.3.9. Questionnaires
- 3.3.10 Brainstorming and Creative Techniques
- 3.3.11. Group Media
- 3.3.12. Analysis of System Interfaces
- 3.3.13. Document Analysis and "Archeology".
- 3.3.14. Case Studies and Scenarios
- 3.3.15. Prototypes
- 3.3.16. Reverse Engineering
- 3.3.17. Reuse of Requirements
- 3.3.18. Good Elicitation Practices

3.4. User Requirements

- 3.4.1. Person
- 3.4.2. Case Studies and User Stories
- 3.4.3. Scenarios
- 3.4.5. Types of Scenarios
- 3.4.6. How to Discover Scenarios

3.5. Prototyping Techniques

- 3.5.1. Prototyping
- 3.5.2. Prototypes According to their Scope
- 3.5.3. Prototypes According to their Seasonality
- 3.5.4. The Fidelity of a Prototype
- 3.5.5. User Interface Prototypes
- 3.5.6. Evaluation of Prototypes

3.6. Requirements Analysis

- 3.6.1. Requirements Analysis
- 3.6.2. Requirements Analysis Best Practices
- 3.6.3. The Data Dictionary
- 3.6.4. Prioritization of Requirements

3.7. Documentation of Requirements

- 3.7.1. The Requirements Specification Document
- 3.7.2. Structure and Contents of an SRS
- 3.7.3. Natural Language Documentation
- 3.7.4. EARS: Easy Approach to Requirements Syntax
- 3.7.5. Non-Functional Requirements
- 3.7.6. Attributes and Templates in Table Form
- 3.7.7. Good Specifications Practices

3.8. Validation and Negotiation of Requirements

- 3.8.1. Validation of Requirements
- 3.8.2. Requirements Validation Techniques
- 3.8.3. Negotiation of Requirements

3.9. Modeling and Requirements Management

- 3.9.1. Requirements Modeling
- 3.9.2. The User's Perspective
- 3.9.3. The Data Perspective
- 3.9.4. The Functional or Flow-Oriented Perspective
- 3.9.5. The Behavioral Perspective
- 3.9.6. Volatility of Requirements
- 3.9.7. Requirements Management Process
- 3.9.8. Tools for Requirements Management
- 3.9.9. Best Practices in Requirements Management

3.10. Critical Systems and Formal Specification

- 3.10.1. Critical Systems
- 3.10.2. Risk-Driven Specification
- 3.10.3. Formal Specification

Module 4. Software Engineering Processes

4.1. Software Engineering Framework
- 4.1.1. Software Features
- 4.1.2. The Main Processes in Software Engineering
- 4.1.3. Software Development Process Models
- 4.1.4. Standard Reference Framework for the Software Development Process: ISO/IEC 12207 Regulations

4.2. Unified Software Development Process
- 4.2.1. The Unified Process
- 4.2.2. Dimensions of the Unified Process
- 4.2.3. Case Studies Driven Development Process
- 4.2.4. Fundamental Unified Process Workflows

4.3. Planning in the Context of Agile Software Development
- 4.3.1. Characteristics of Agile Software Development
- 4.3.2. Different Planning Time Horizons in Agile Development
- 4.3.3. Scrum Agile Development Framework and Planning Time Horizons
- 4.3.4. User Stories as a Planning and Estimating Unit
- 4.3.5. Common Techniques for Deriving an Estimate
- 4.3.6. Scales for Interpreting Estimates
- 4.3.7. Planning Poker
- 4.3.8. Common Scheduling Types: Delivery Scheduling and Iteration Scheduling

4.4. Distributed Software Design Styles and Service-Oriented Software Architectures
- 4.4.1. Communication Models in Distributed Software Systems
- 4.4.2. Middleware
- 4.4.3. Architecture Patterns for Distributed Systems
- 4.4.4. General Software Service Design Process
- 4.4.5. Design Aspects of Software Services
- 4.4.6. Composition of Services
- 4.4.7. Web Services Architecture
- 4.4.8. Infrastructure and SOA Components

4.5. Introduction to Model Driven Software Development
- 4.5.1. The Model Concept
- 4.5.2. Model-Driven Software Development
- 4.5.3. MDA Model-Driven Development Framework
- 4.5.4. Elements of a Transformation Model

4.6. Graphical User Interface Design
- 4.6.1. Principles of User Interface Design
- 4.6.2. Architectural Design Patterns for Interactive Systems: Model View Controller (MVC)
- 4.6.3. UX User Experience
- 4.6.4. User-Centered Design
- 4.6.5. Graphical User Interface Analysis and Design Process
- 4.6.6. Usability of User Interfaces
- 4.6.7. Accessibility in User Interfaces

4.7. Web Application Design
- 4.7.1. Characteristics of Web Applications
- 4.7.2. Web Application User Interface
- 4.7.3. Navigation Design
- 4.7.4. Basic Interaction Protocol for Web Applications
- 4.7.5. Architecture Styles for Web Applications

4.8. Software Testing Strategies and Techniques and Software Quality Factors
- 4.8.1. Testing Strategies
- 4.8.2. Test Case Designs
- 4.8.3. Value for Money
- 4.8.4. Quality Models
- 4.8.5. ISO/IEC 25000 Family of Standards (SQuaRE)
- 4.8.6. Product Quality Model (ISO 2501n)
- 4.8.7. Data Quality Models (ISO 2501n)
- 4.8.8. Software Quality Management

4.9. Introduction to Software Engineering Metrics
- 4.9.1. Basic Concepts: Measurements, Metrics and Indicators
- 4.9.2. Types of Metrics in Software Engineering
- 4.9.3. The Measurement Process
- 4.9.4. ISO 25024. External and Quality Metrics in Use
- 4.9.5. Object-Oriented Metrics

4.10. Software Maintenance and Re-Engineering
- 4.10.1. Maintenance Process
- 4.10.2. Standard Maintenance Process Framework. ISO/EIEC 14764
- 4.10.3. Software Re-Engineering Process Model
- 4.10.4. Inverse Engineering

Module 5. Quality and Auditing of Information Systems

5.1. Introduction to Information Security Management Systems
- 5.1.1. Fundamental Principles of ISMS
- 5.1.2. ISMS Golden Rules
- 5.1.3. Role of IT Audit in ISMSs

5.2. Safety Management Planning
- 5.2.1. Concepts Related to Safety Management
- 5.2.2. Classification of Information: Objectives, Concepts and Roles
- 5.2.3. Implementation of Security Policies: Security Policies, Standards and Procedures
- 5.2.4. Risk Management: Information Assets Risk Principles and Analysis

5.3. Main Mechanisms for the Protection of Information Assets I
- 5.3.1. Summary of the Main Cryptographic Tools for the Protection of the CID Triad of the CID Triad
- 5.3.2. Consideration of Privacy, Anonymity and Adequate Management of User Traceability Requirements

5.4. Main Mechanisms for the Protection of Information Assets II
- 5.4.1. Communications Security: Protocols, Devices and Security Architectures
- 5.4.2. Operating System Security

5.5. ISMS Internal Controls
- 5.5.1. ISMS Controls Taxonomy: Administrative, Logical and Physical Controls
- 5.5.2. Classification of Controls According to How Threats Are Addressed: Controls for Threat Prevention, Detection and Correction
- 5.5.3. Implementation of Internal Control Systems in ISMSs

5.6. Types of Audits
- 5.6.1. Difference between Audit and Internal Control
- 5.6.2. Internal vs. External Audit
- 5.6.3. Audit Classification according to the Objective and Type of Analysis

5.7. Screenwriter and Screenplay: Subject Matter and Object Protected by Intellectual Property
- 5.7.1. Introduction to Penetration Testing and Forensic Analysis
- 5.7.2. Definition and Relevance of Fingerprinting and Footprinting Concepts

5.8. Vulnerability Scanning and Network Traffic Monitoring
- 5.8.1. Tools for Vulnerability Analysis in Systems
- 5.8.2. Main Vulnerabilities in the Context of Web Applications
- 5.8.3. Analysis of Communications Protocols

5.9. The IT Audit Process
- 5.9.1. Life Cycle Concept in Systems Development
- 5.9.2. Activity and Process Monitoring: Collection and Treatment of Evidence
- 5.9.3. IT Audit Methodology
- 5.9.4. IT Audit Process
- 5.9.5. Identification of the Main Crimes and Misdemeanors in the Context of Information Technologies
- 5.9.6. Computer Crime Investigation: Introduction to Forensic Analysis and its relation to Computer Auditing

5.10. Business Continuity and Disaster Recovery Plans
- 5.10.1. Definition of Business Continuity Plan and the Business Interruption Concept
- 5.10.2. NIST Recommendation on Business Continuity Plans
- 5.10.3. Disaster Recovery Plan
- 5.10.4. Disaster Recovery Plan Process

Module 6. Integration Systems

6.1. Introduction to Information Systems in the Enterprise

6.1.1. The Role of Information Systems
6.1.2. What is an Information System?
6.1.3. Dimensions of Information Systems
6.1.4. Business Processes and Information Systems
6.1.5. The IS/IT Department

6.2. Opportunities and Needs of Information Systems in the Enterprise

6.2.1. Organizations and Information Systems
6.2.2. Features of Organizations
6.2.3. Impact of Information Systems in the Enterprise
6.2.4. Information Systems to Achieve a Competitive Advantage
6.2.5. Use of Systems in the Administration and Management of the Enterprise

6.3. Basic Concepts of Information Systems and Technologies

6.3.1. Data, Information and Knowledge
6.3.2. Technology and Information Systems
6.3.3. Technology Components
6.3.4. Classification and Types of Information Systems
6.3.5. Service and Business Process Based Architectures
6.3.6. Forms of Systems Integration

6.4. Systems for the Integrated Enterprise Resource Planning

6.4.1. Business Needs
6.4.2. An Integrated Enterprise Resource Planning
6.4.3. Acquisition vs. Development
6.4.4. ERP Implementation
6.4.5. Implications for Management
6.4.6. Leading ERP Vendors

6.5. Supply Chain and Customer Relationship Management Information Systems

6.5.1. Definition of Supply Chain
6.5.2. Effective Supply Chain Management
6.5.3. The Role of Information Systems
6.5.4. Supply Chain Management Solutions
6.5.5. Customer Relationship Management
6.5.6. The Role of Information Systems
6.5.7. Implementation of a CRM System
6.5.8. Critical Success Factors in CRM Implementation
6.5.9. CRM, e-CRM and Other Trends

6.6. ICT Investment Decision-Making and Information Systems Planning

6.6.1. Criteria for ICT Investment Decisions
6.6.2. Linking the Project to the Management and Business Plan
6.6.3. Management Implications
6.6.4. Redesign of Business Processes
6.6.5. Management's Decision on Implementation Methodologies
6.6.6. Need for Information Systems Planning
6.6.7. Objectives, Participants and Moments
6.6.8. Structure and Development of the Systems Planning
6.6.9. Follow-up and Updating

6.7. Security Considerations in the Use of ICTs

6.7.1. Risk Analysis
6.7.2. Security in Information Systems
6.7.3. Practical Advice

6.8. Feasibility of ICT Project Implementation and Financial Aspects in Information Systems Projects

6.8.1. Description and Objectives
6.8.2. EVS Participants
6.8.3. Techniques and Procedures
6.8.4. Cost Structure
6.8.5. Financial Projection
6.8.6. Budgets

- 6.9. Business Intelligence
    - 6.9.1. What is Business Intelligence?
    - 6.9.2. BI Implementation Strategy
    - 6.9.3. Present and Future in BI
- 6.10. ISO/IEC 12207
    - 6.10.1. What is "ISO/IEC 12207"?
    - 6.10.2. Analysis of Information Systems
    - 6.10.3. Information System Design
    - 6.10.4. Implementation and Acceptance of the Information System

Module 7. Software Reuse

- 7.1. General Overview of the Software Reuse
    - 7.1.1. What is Software Reuse?
    - 7.1.2. Advantages and Disadvantages of Software Reuse
    - 7.1.3. Main Techniques of Software Reuse
- 7.2. Introduction to Design Patterns
    - 7.2.1. What is a Design Patterns?
    - 7.2.2. Catalog of the Main Design Patterns
    - 7.2.3. How to Use Patterns to Solve Design Problems
    - 7.2.4. How to Select the Best Design Pattern
- 7.3. Creation Patterns I
    - 7.3.1. Creation Patterns
    - 7.3.2. Abstract Factory Pattern
    - 7.3.3. Example of Abstract Factory Pattern implementation
    - 7.3.4. Builder Pattern
    - 7.3.5. Builder Implementation Example
    - 7.3.6. Abstract Factory Pattern vs. Builder
- 7.4. Creation Patterns II
    - 7.4.1. Factory Method Pattern
    - 7.4.2. Factory Method vs. Abstract Factory
    - 7.4.3. Singleton Pattern
- 7.5. Structural Patterns I
    - 7.5.1. Structural Patterns
    - 7.5.2. Adapter Pattern
    - 7.5.3. Bridge Pattern
- 7.6. Structural Patterns II
    - 7.6.1. Composite Pattern
    - 7.6.2. Decorator Pattern
- 7.7. Structural Patterns III
    - 7.7.1. Facade Pattern
    - 7.7.2. Proxy Pattern
- 7.8. Behavioral Patterns I
    - 7.8.1. Concept of Behavioral Patterns
    - 7.8.2. Behavior Pattern: Chain of Responsibility
    - 7.8.3. Behavior Pattern Order
- 7.9. Behavioral Patterns II
    - 7.9.1. Interpreter Pattern
    - 7.9.2. Iterator Pattern
    - 7.9.3. Observer Pattern
    - 7.9.4. Strategy Pattern
- 7.10. *Frameworks*
    - 7.10.1. Framework Concept
    - 7.10.2. Development Using Frameworks
    - 7.10.3. Model View Controller Pattern
    - 7.10.4. Framework for Graphical User Interface Design
    - 7.10.5. Frameworks for Web Application Development
    - 7.10.6. Frameworks for Managing Object Persistence in Databases

Module 8. Information Technology Services

8.1. Digital Transformation I
- 8.1.1. Business Innovation
- 8.1.2. Production Management
- 8.1.3. Financial Management

8.2. Digital Transformation II
- 8.2.1. Marketing
- 8.2.2. HR Management
- 8.2.3. The Integrated Information System

8.3. Case Study
- 8.3.1. Company Presentation
- 8.3.2. Methodologies to Analyze the Acquisition of IT
- 8.3.3. Determining the Costs, Benefits and Risks
- 8.3.4. Economic Evaluation of Investment

8.4. ICT Governance and Management
- 8.4.1. Definition of IT and Information Systems Governance
- 8.4.2. Difference Between IT Systems Governance and Management
- 8.4.3. Framework for IT Systems Governance and Management
- 8.4.4. Regulations and IT Systems Governance and Management

8.5. ICT Corporate Governance
- 8.5.1. What is Good Corporate Governance?
- 8.5.2. ICT Governance Background
- 8.5.3. The ISO/IEC 38500:2008 Standard
- 8.5.4. Implementation of Good ICT Governance
- 8.5.5. ICT Governance and Best Practices
- 8.5.6. Corporate Governance. Summary and Trends

8.6. Control Objectives for Information and Related Technologies (COBIT)
- 8.6.1. Application Framework
- 8.6.2. Domain: Planning and Organization
- 8.6.3. Domain: Acquisition and Implementation
- 8.6.4. Domain: Delivery and Support
- 8.6.5. Domain: Supervision and Evaluation
- 8.6.6. Application of the COBIT Guide

8.7. The Information Technology Infrastructure Library (ITIL)
- 8.7.1. Introduction to ITIL
- 8.7.2. Service Strategies
- 8.7.3. Service Design
- 8.7.4. Transition Between Services
- 8.7.5. Service Operation
- 8.7.6. Improving the Service

8.8. The Service Management System
- 8.8.1. Basic Principles of UNE-ISO/IEC 20000-1
- 8.8.2. The Structure of the ISO/IEC 20000 Regulations
- 8.8.3. Service Management System (SMS) Requirements
- 8.8.4. Design and Transition of New or Modified Services
- 8.8.5. Service Provision Processes
- 8.8.6. Groups of Processes

8.9. The Software Asset Management System
- 8.9.1. Justification of Needs
- 8.9.2. Background
- 8.9.3. Presentation of the 19770 Regulation
- 8.9.4. Management Implementation

8.10. Business Continuity Management
- 8.10.1. Business Continuity Plan
- 8.10.2. Implementation of a BCP

Module 9. Information Systems Security

9.1. A global Perspective on Security, Cryptography and Classical Cryptanalysis
   9.1.1. Computer Security: Historical Perspective
   9.1.2. But What Exactly is Meant by Security?
   9.1.3. History of Cryptography
   9.1.4. Substitution Ciphers
   9.1.5. Case Study: The Enigma Machine
9.2. Symmetric Cryptography
   9.2.1. Introduction and Basic Terminology.
   9.2.2. Symmetric Encryption
   9.2.3. Modes of Operation
   9.2.4. DES
   9.2.5. The New AES Standard
   9.2.6. Encryption in Flow
   9.2.7. Cryptanalysis
9.3. Asymmetric Cryptography
   9.3.1. Origins of Public Key Cryptography
   9.3.2. Basic Concepts and Operation
   9.3.3. The RSA Algorithm
   9.3.4. Digital Certificates
   9.3.5. Key Storage and Management
9.4. Network Attacks
   9.4.1. Network Threats and Attacks
   9.4.2. Enumeration
   9.4.3. Traffic Interception: *Sniffers*
   9.4.4. Denial of Service Attacks
   9.4.5. ARP Poisoning Attacks
9.5. Security Architectures
   9.5.1. Traditional Security Architectures
   9.5.2. Secure Socket Layer: SSL
   9.5.3. SSH Protocol
   9.5.4. Virtual Private Networks (VPNs)
   9.5.5. External Storage Unit Protection Mechanisms
   9.5.6. Hardware Protection Mechanisms
9.6. System Protection Techniques and Secure Code Development
   9.6.1. Operational Safety
   9.6.2. Resources and Controls
   9.6.3. Monitoring
   9.6.4. Intrusion Detection Systems
   9.6.5. Host IDS
   9.6.6. Network IDS
   9.6.7. Signature-based IDS
   9.6.8. Lure Systems
   9.6.9. Basic Security Principles in Code Development
   9.6.10. Failure Management
   9.6.11. Public Enemy Number 1: Buffer Overflows
   9.6.12. Cryptographic Botches
9.7. Botnets and Spam
   9.7.1. Origin of the Problem
   9.7.2. Spam Process
   9.7.3. Sending Spam
   9.7.4. Refinement of Mailing Lists
   9.7.5. Protection Techniques
   9.7.6. Anti-Spam Service offered by Third-Parties
   9.7.7. Study Cases
   9.7.8. Exotic Spam
9.8. Web Auditing and Attacks
   9.8.1. Information Gathering
   9.8.2. Attack Techniques
   9.8.3. Tools

- 9.9. Malware and Malicious Code
  - 9.9.1. What is Malware?
  - 9.9.2. Types of Malware
  - 9.9.3. Virus
  - 9.9.4. Criptovirus
  - 9.9.5. Worms
  - 9.9.6. Adware
  - 9.9.7. Spyware
  - 9.9.8. Hoaxes
  - 9.9.9. Phishing
  - 9.9.10. Trojans
  - 9.9.11. The Economy of Malware
  - 9.9.12. Possible Solutions
- 9.10. Forensic Analysis
  - 9.10.1. Evidence Collection
  - 9.10.2. Evidence Analysis
  - 9.10.3. Anti-Forensic Techniques
  - 9.10.4. Case Study

Module 10. Project Management

- 10.1. Fundamental Concepts of Project Management and the Project Management Lifecycle
  - 10.1.1. What is a Project?
  - 10.1.2. Common Methodology
  - 10.1.3. What is Project Management?
  - 10.1.4. What is a Project Plan?
  - 10.1.5. Benefits
  - 10.1.6. Project Life Cycle
  - 10.1.7. Process Groups or Project Management Life Cycle
  - 10.1.8. The Relationship between Process Groups and Knowledge Areas
  - 10.1.9. Relationships between Product and Project Life Cycle
- 10.2. Start-Up and Planning
  - 10.2.1. From the Idea to the Project
  - 10.2.2. Development of the Project Record
  - 10.2.3. Project Kick-Off Meeting
  - 10.2.4. Tasks, Knowledge and Skills in the Startup Process
  - 10.2.5. The Project Plan
  - 10.2.6. Development of the Basic Plan. Steps
  - 10.2.7. Tasks, Knowledge and Skills in the Planning Process
- 10.3. Stakeholders and Outreach Management
  - 10.3.1. Identify Stakeholders
  - 10.3.2. Develop Plan for Stakeholder Management
  - 10.3.3. Manage Stakeholder Engagement
  - 10.3.4. Control Stakeholder Engagement
  - 10.3.5. The Objective of the Project
  - 10.3.6. Scope Management and its Plan
  - 10.3.7. Gathering Requirements
  - 10.3.8. Define the Scope Statement
  - 10.3.9. Create the WBS
  - 10.3.10. Verify and Control the Scope
- 10.4. The Development of the Time-Schedule
  - 10.4.1. Time Management and its Plan
  - 10.4.2. Defining Activities
  - 10.4.3. Establishment of the Sequence of Activities
  - 10.4.4. Estimated Resources for Activities
  - 10.4.5. Estimated Duration of Activities
  - 10.4.6. Development of the Time-Schedule and Calculation of the Critical Path
  - 10.4.7. Schedule Control

10.5. Budget Development and Risk Response
- 10.5.1. Estimate Costs
- 10.5.2. Develop Budget and S-Curve
- 10.5.3. Cost Control and Earned Value Method
- 10.5.4. Risk Concepts
- 10.5.5. How to Perform a Risk Analysis
- 10.5.6. The Development of the Response Plan

10.6. Quality Management
- 10.6.1. Quality Planning
- 10.6.2. Assuring Quality
- 10.6.3. Quality Control
- 10.6.4. Basic Statistical Concepts
- 10.6.5. Quality Management Tools

10.7. Communication and Human Resources
- 10.7.1. Planning Communications Management
- 10.7.2. Communications Requirements Analysis
- 10.7.3. Communication Technology
- 10.7.4. Communication Models
- 10.7.5. Communication Methods
- 10.7.6. Communications Management Plan
- 10.7.7. Manage Communications
- 10.7.8. Management of Human Resources
- 10.7.9. Main Stakeholders and their Roles in the Projects
- 10.7.10. Types of Organization
- 10.7.11. Project Organization
- 10.7.12. The Work Team

10.8. Procurement
- 10.8.1. The Procurement Process
- 10.8.2. Planning
- 10.8.3. Search for Suppliers and Request for Quotations
- 10.8.4. Contract Allocation
- 10.8.5. Contract Administration
- 10.8.6. Contracts
- 10.8.7. Types of Contracts
- 10.8.8. Contract Negotiation

10.9. Execution, Monitoring and Control and Closure
- 10.9.1. Process Groups
- 10.9.2. Project Execution
- 10.9.3. Project Monitoring and Control
- 10.9.4. Project Closure

10.10. Professional Responsibility
- 10.10.1. Professional Responsibility
- 10.10.2. Characteristics of Social and Professional Responsibility
- 10.10.3. Project Leader Code of Ethics
- 10.10.4. Liability vs. PMP®
- 10.10.5. Examples of Liability
- 10.10.6. Benefits of Professionalization

06

Clinical Internship

After completing the online theoretical period, this Hybrid Professional Master's Degree in Advanced Software Engineering includes an Internship Program in a reference institution. During this period, graduates will have the support of a tutor who will help them to get the most out of this experience. Thanks to this, computer scientists will acquire advanced skills to experience a remarkable leap in quality in their career.

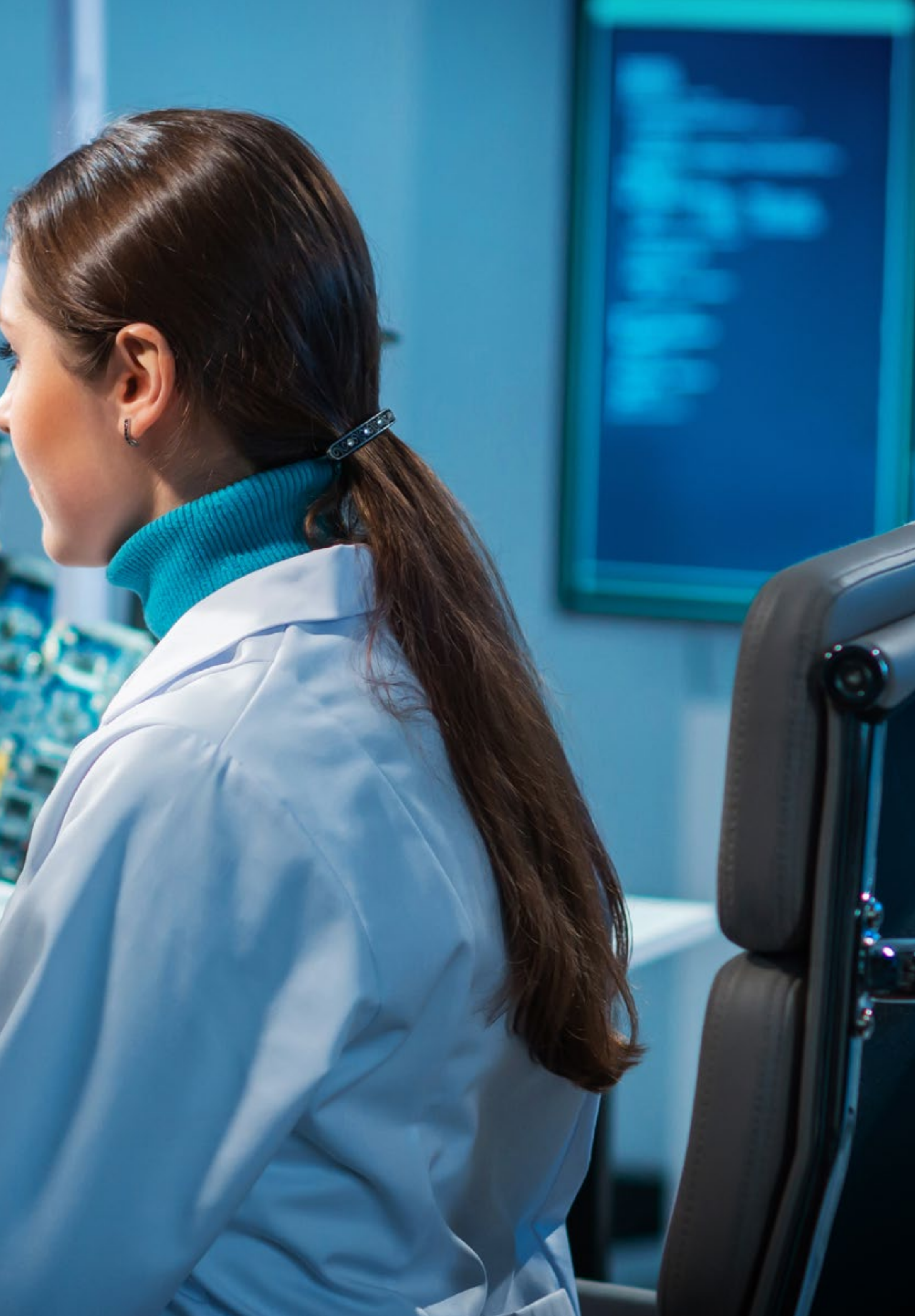
> You will carry out your practical stay in a distinguished company, where you will have the support of renowned professionals in Advanced Software Engineering”

The Internship Program of this program in Advanced Software Engineering consists of a practical internship in a distinguished company, lasting 3 weeks, from Monday to Friday, with 8 consecutive hours of practical training with an assistant specialist.

In this program proposal, completely practical in nature, the activities are aimed at developing and perfecting the skills necessary to provide Advanced Software Engineering IT services, as well as conditions that require a high level of qualification, oriented to the specific training for the exercise of the activity.

This is an excellent opportunity for computer scientists to immerse themselves in a real working environment, where they will be part of a team of professionals specialized in this field. Therefore, graduates will participate in projects related to subjects such as security in information systems, software reuse or web application design, among others.

The practical education will be carried out with the active participation of the student performing the activities and procedures of each area of competence (learning to learn and learning to do), with the accompaniment and guidance of teachers and other training partners that facilitate teamwork and multidisciplinary integration as transversal competencies for the praxis of Computer Science (learning to be and learning to relate).



The procedures described below will be the basis of the practical part of the program, and their implementation will be subject to the center's own availability and workload, the proposed activities being the following:

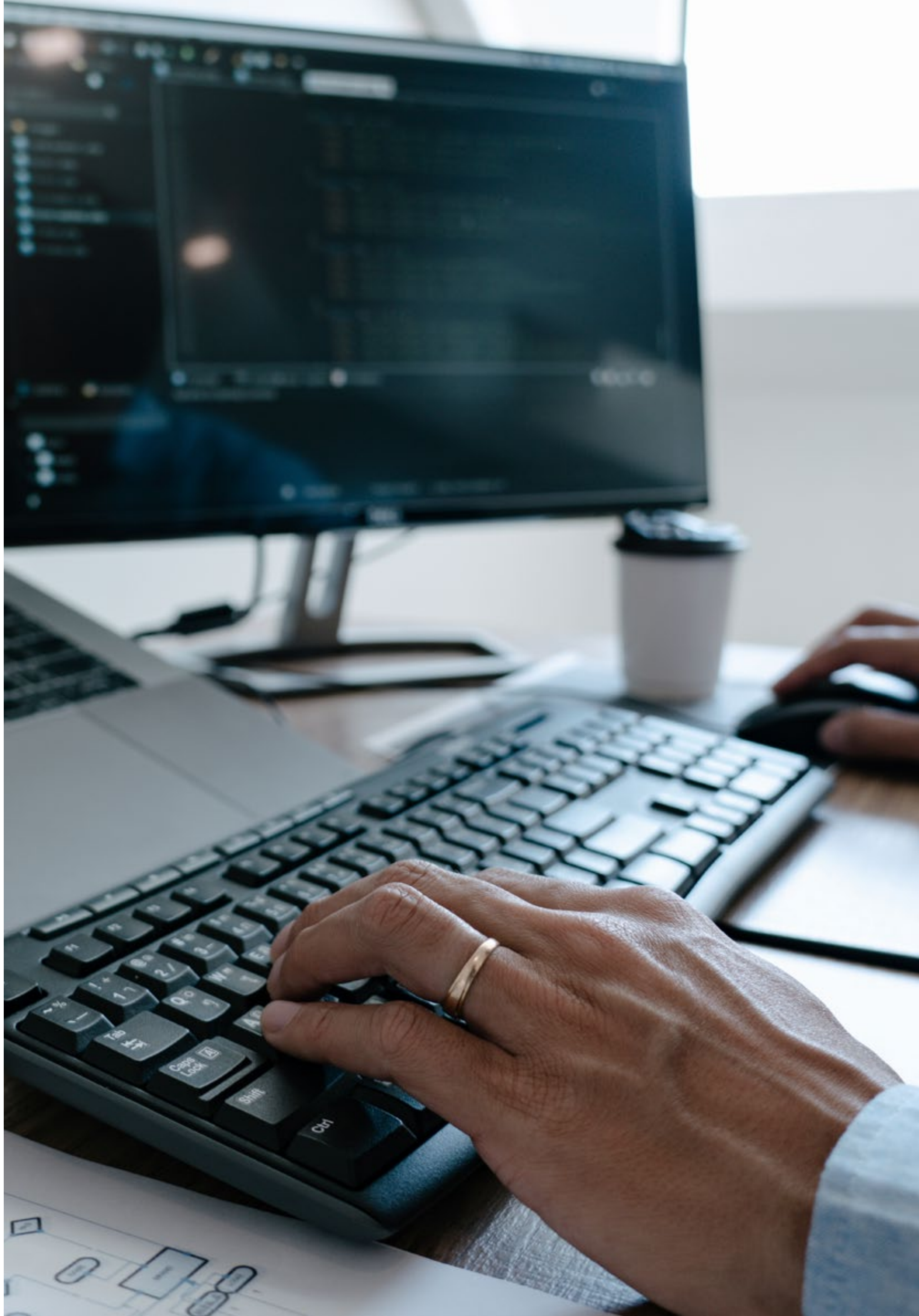
Module	Practical Activity
Advanced Software Development	Design software architectures that are scalable, robust, and easy to maintain
	Use modeling techniques such as Unified Modeling Language to represent both system structure and behavior prior to implementation
	Write efficient and clean code in a variety of programming languages
	Use agile methodologies to plan, execute and track software projects
Requirement Management	Analyze the user's environment and study the application domain to identify problems that the software must solve
	Use modeling techniques (such as UML diagrams or use cases and scenarios) to represent requirements in a structured way
	Draft requirements specification documents with stakeholders to ensure that the captured requirements are correct
	Develop prototypes and simulations of the system in order to validate requirements with users
Requirement Management	Establish quality standards for information systems maintenance, based on frameworks
	Conduct functional, performance, security and usability testing
	Identify and assess risks associated with information systems
	Implement continuous integration and deployment pipelines to ensure that new versions of software are developed, tested and deployed efficiently
Connecting Systems	Delve into systems to identify potential integration challenges and plan for effective solutions
	Create data models that define how data will be structured and handled between integrated systems
	Design the integration architecture, selecting the most appropriate patterns and methods (e.g., service-based integration, messaging, APIs)
	Configure interfaces to enable communication and data transfer between the integrated systems

Civil Liability Insurance

This institution's main concern is to guarantee the safety of the students and other collaborating agents involved in the internship process at the company. Among the measures dedicated to achieve this is the response to any incident that may occur during the entire teaching-learning process.

To this end, this educational entity undertakes to take out civil liability insurance to cover any eventuality that may arise during the internship during the stay at the internship center.

This liability policy for interns will have broad coverage and will be taken out prior to the start of the practical training period. That way professionals will not have to worry in case of having to face an unexpected situation and will be covered until the end of the internship program at the center.



General Conditions of the Internship Program

The general terms and conditions of the internship agreement for the program are as follows:

1. TUTOR: During the Hybrid Professional Master's Degree, students will be assigned with two tutors who will accompany them throughout the process, answering any doubts and questions that may arise. On the one hand, there will be a professional tutor belonging to the internship center who will have the purpose of guiding and supporting the student at all times. On the other hand, they will also be assigned with an academic tutor whose mission will be to coordinate and help the students during the whole process, solving doubts and facilitating everything they may need. In this way, the student will be accompanied and will be able to discuss any doubts that may arise, both clinical and academic.

2. DURATION: The internship program will have a duration of three continuous weeks, in 8-hour days, 5 days a week. The days of attendance and the schedule will be the responsibility of the center and the professional will be informed well in advance so that they can make the appropriate arrangements.

3. ABSENCE: If the students does not show up on the start date of the Hybrid Professional Master's Degree, they will lose the right to it, without the possibility of reimbursement or change of dates. Absence for more than two days from the internship, without justification or a medical reason, will result in the professional's withdrawal from the internship, therefore, automatic termination of the internship. Any problems that may arise during the course of the internship must be urgently reported to the academic tutor.

4. CERTIFICATION: Professionals who pass the Hybrid Professional Master's Degree will receive a certificate accrediting their stay at the center.

5. EMPLOYMENT RELATIONSHIP: the Hybrid Professional Master's Degree shall not constitute an employment relationship of any kind.

6. PRIOR EDUCATION: Some centers may require a certificate of prior education for the Hybrid Professional Master's Degree. In these cases, it will be necessary to submit it to the TECH internship department so that the assignment of the chosen center can be confirmed.

7. DOES NOT INCLUDE: The Hybrid Professional Master's Degree will not include any element not described in the present conditions. Therefore, it does not include accommodation, transportation to the city where the internship takes place, visas or any other items not listed.

However, students may consult with their academic tutor for any questions or recommendations in this regard. The academic tutor will provide the student with all the necessary information to facilitate the procedures in any case.

07

Where Can I Do the Internship?

Loyal to its philosophy of providing high quality university degrees, TECH expands the academic opportunities for students and enables them to carry out their practical training in different entities of international prestige. In this way, computer scientists have an ideal opportunity to improve their professional quality by working with the best specialists in the field of Advanced Software Engineering.

You will carry out your practical stay in a reference entity in Advanced Software Engineering"

The student will be able to complete the practical part of this Hybrid Professional Master's Degree at the following centers:

Computer Science

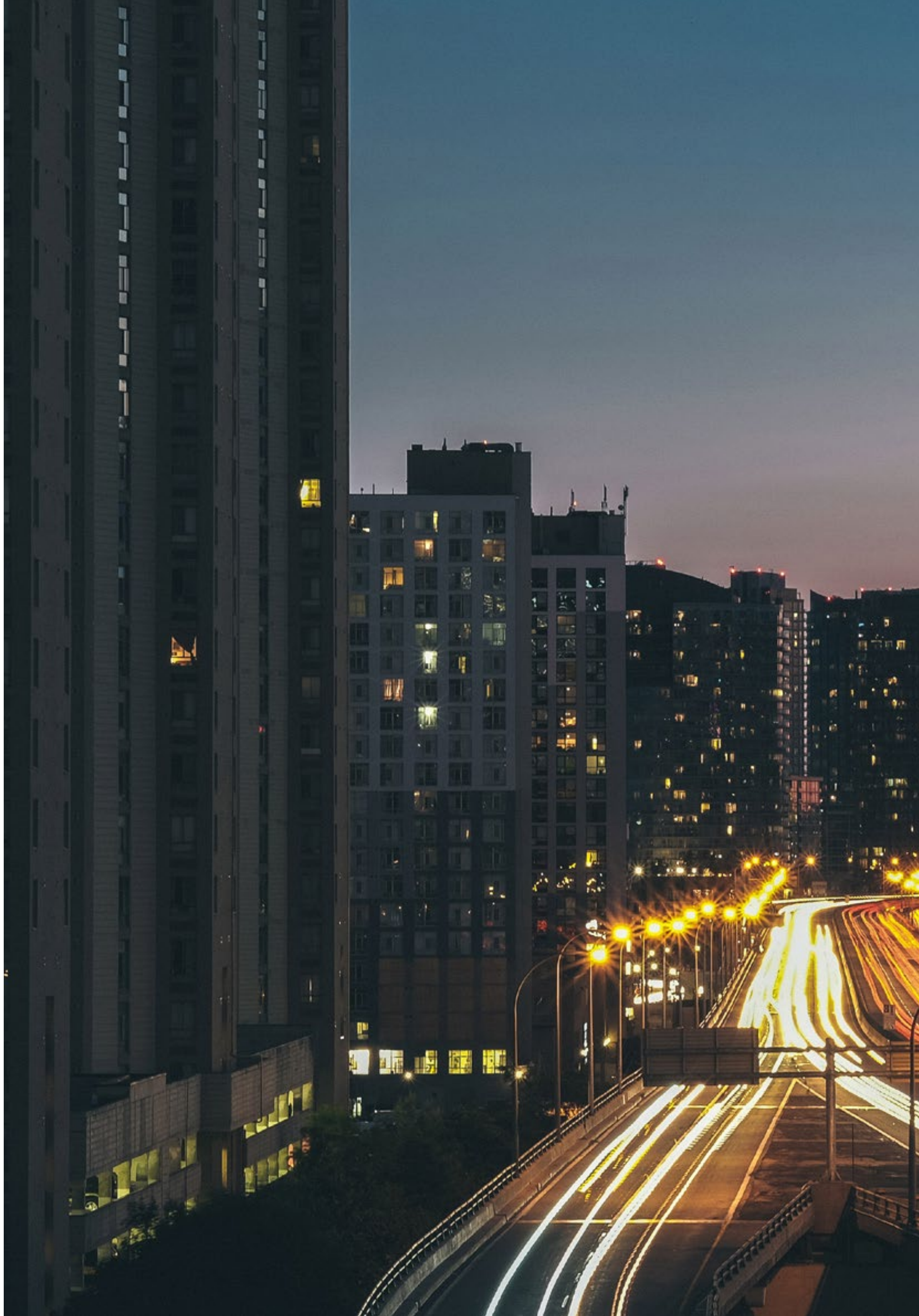
Captia Ingeniería

Country	City
Spain	Madrid

Address: Av. de las Nieves, 37, Bloque A Planta 1
Oficina E, 28935, Móstoles, Madrid

IT company dedicated to providing advanced technological solutions to industries.

Related internship programs:
- Visual Analytics and Big Data
- Software Development



Delve into the most relevant theory in this field, subsequently applying it in a real work environment"

08

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning.**

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

> *At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"*

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

> *Our program prepares you to face new challenges in uncertain environments and achieve success in your career"*

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

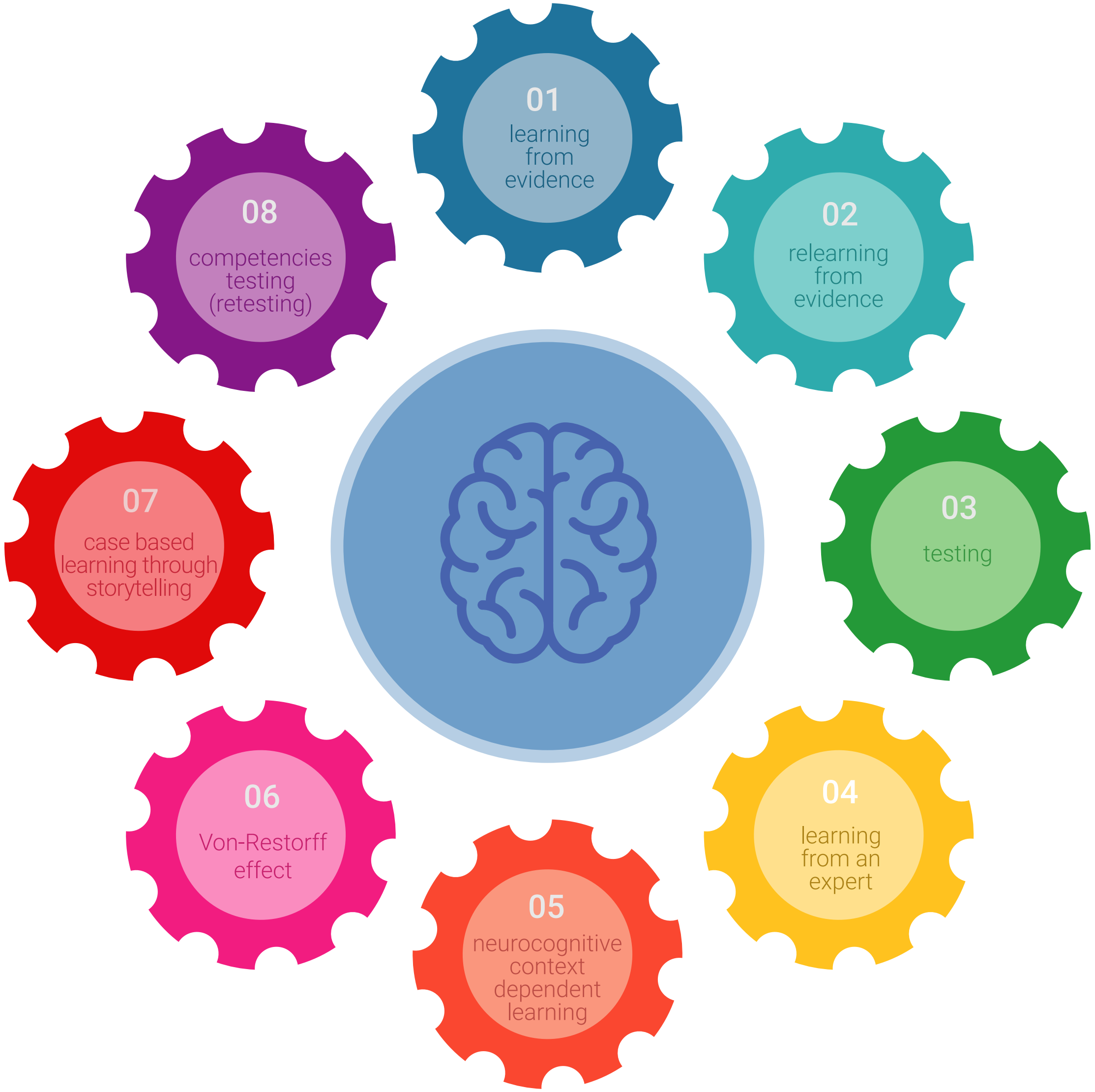
What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

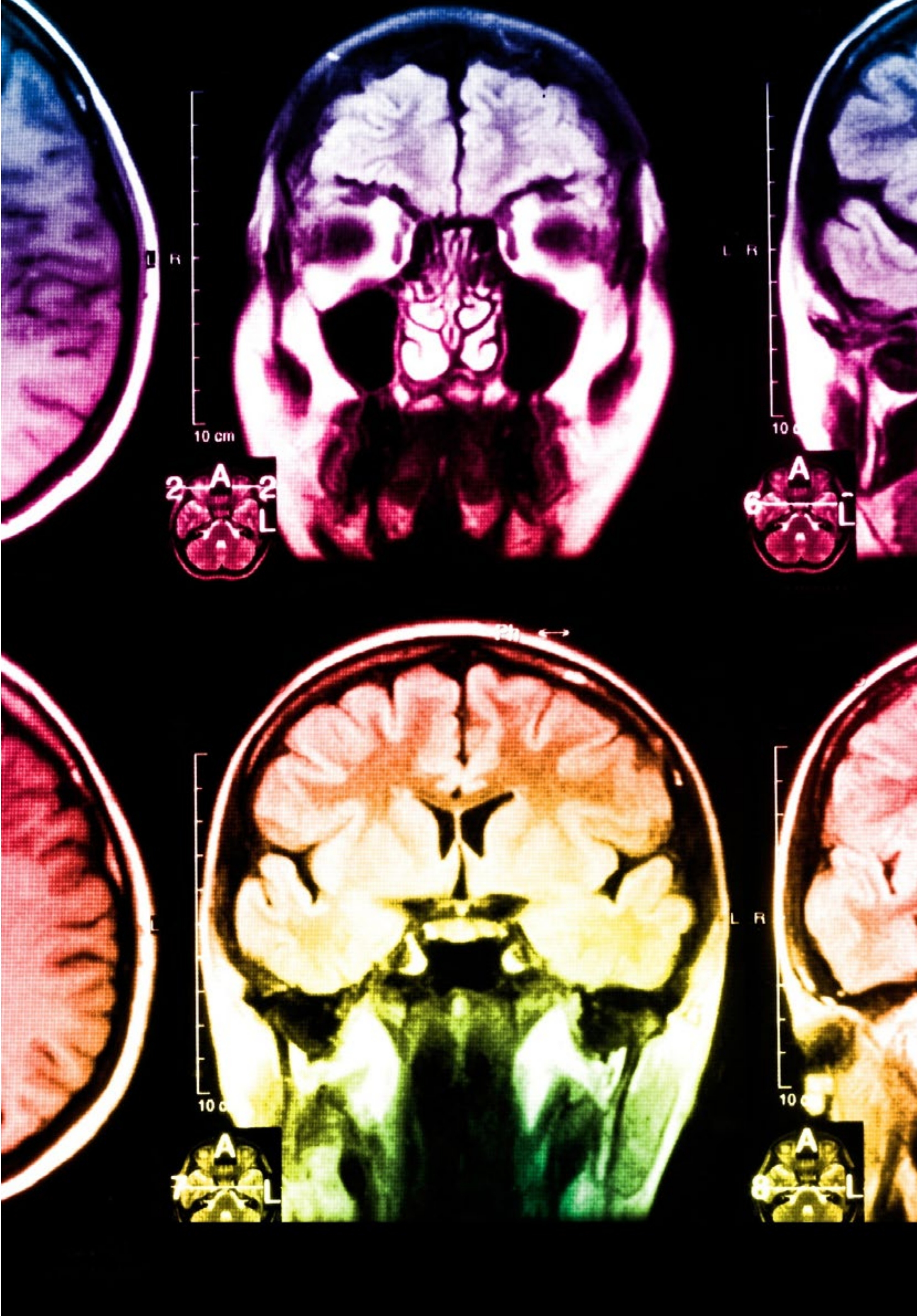
We enhance the Case Study with the best 100% online teaching method: Relearning.

> *In 2019, we obtained the best learning results of all online universities in the world.*


01
learning from evidence
02
relearning from evidence
03
testing
04
learning from an expert
05
neurocognitive context dependent learning
06
Von-Restorff effect
07
case based learning through storytelling
08
competencies testing (retesting)

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

Relearning will allow you to learn with less effort and better performance, involving you more in your training, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation for success.

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.

Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

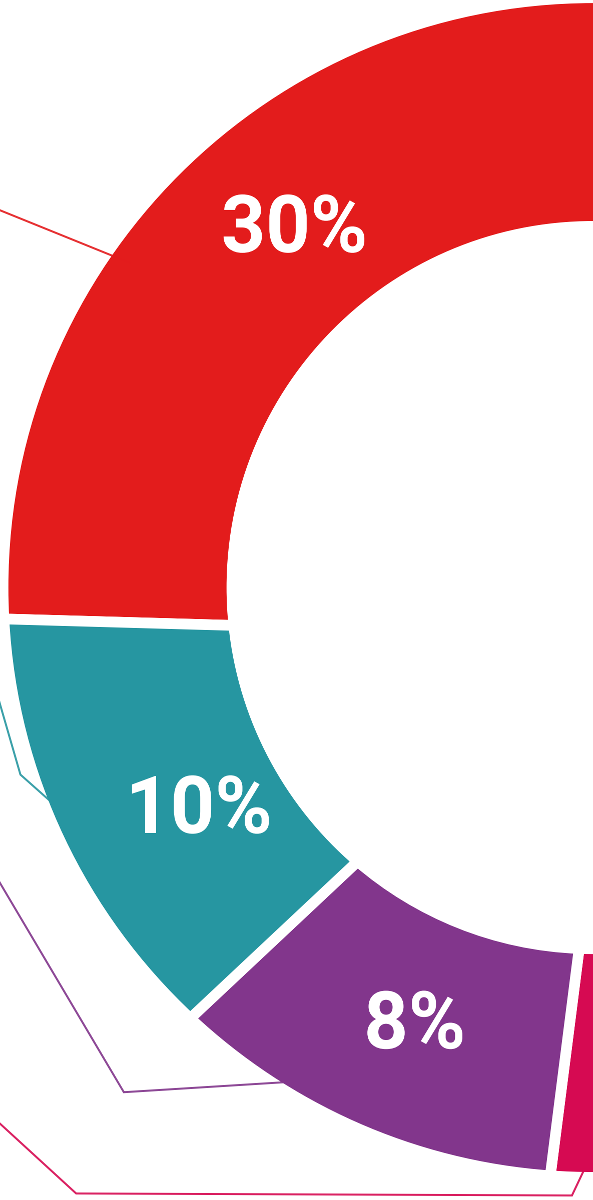
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

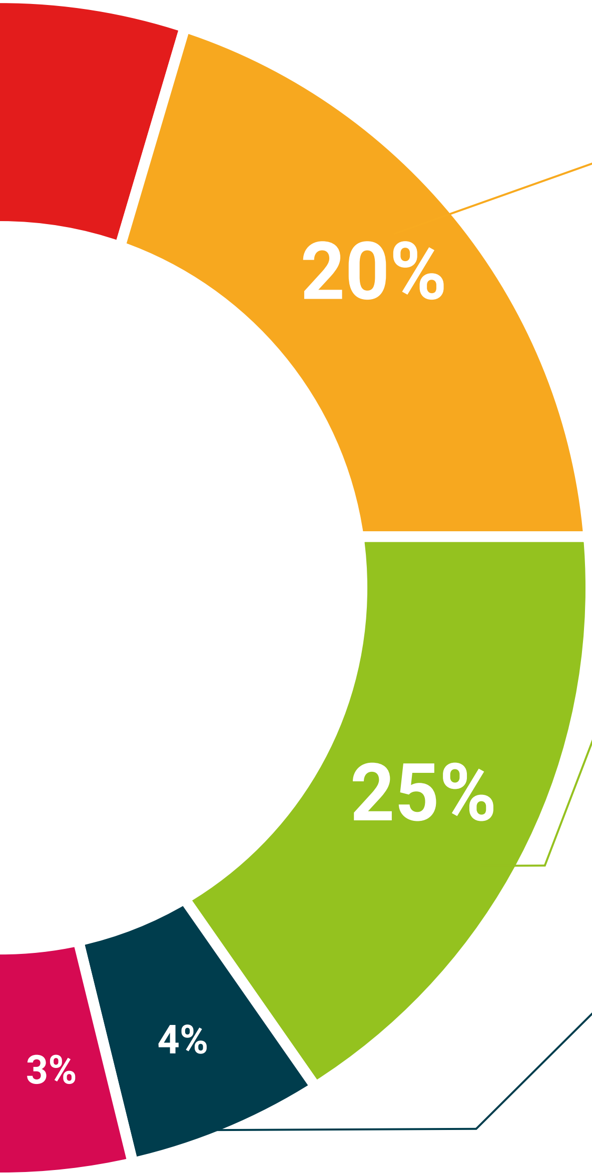
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

09

Certificate

The Hybrid Professional Master's Degree in Advanced Software Engineering guarantees students, in addition to the most rigorous and up-to-date education, access to a Hybrid Professional Master's Degree issued by TECH Global University.

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This private qualification will allow you to obtain a **Hybrid Professional Master’s Degree in Advanced Software Engineering** endorsed by **TECH Global University**, the world’s largest online university.

TECH Global University is an official European University publicly recognized by the Government of Andorra (***official bulletin***). Andorra is part of the European Higher Education Area (EHEA) since 2003. The EHEA is an initiative promoted by the European Union that aims to organize the international training framework and harmonize the higher education systems of the member countries of this space. The project promotes common values, the implementation of collaborative tools and strengthening its quality assurance mechanisms to enhance collaboration and mobility among students, researchers and academics.

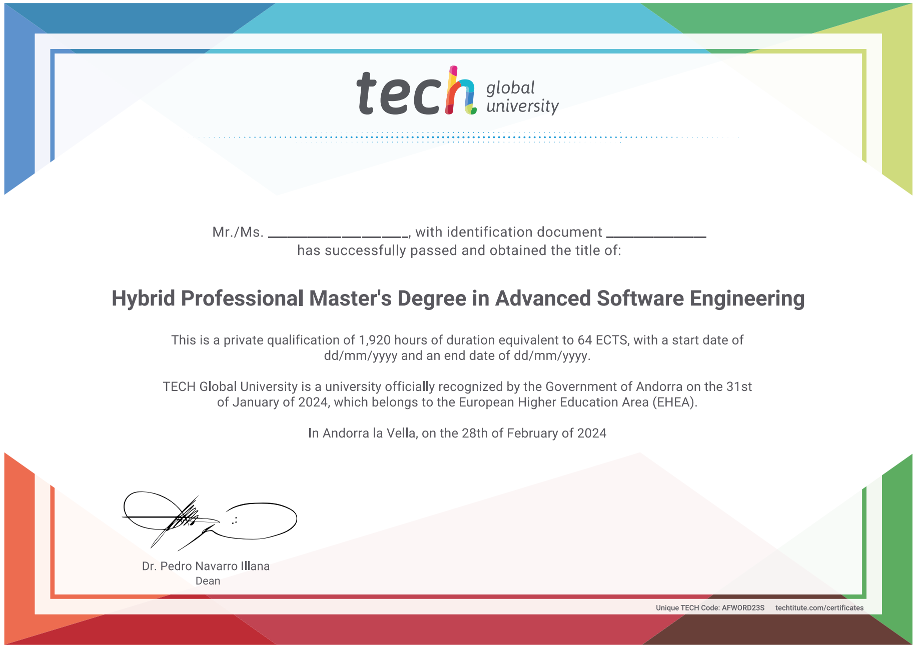
This **TECH Global University** private qualification is a European program of continuing education and professional updating that guarantees the acquisition of competencies in its area of knowledge, providing a high curricular value to the student who completes the program.

Title: **Hybrid Professional Master's Degree in Advanced Software Engineering**

Modality: **Hybrid (Online + Internship)**

Duration: **12 months**

Accreditation: **60 + 4 ECTS**



Mr./Ms. _________________, with identification document ____________ has successfully passed and obtained the title of:

Hybrid Professional Master's Degree in Advanced Software Engineering

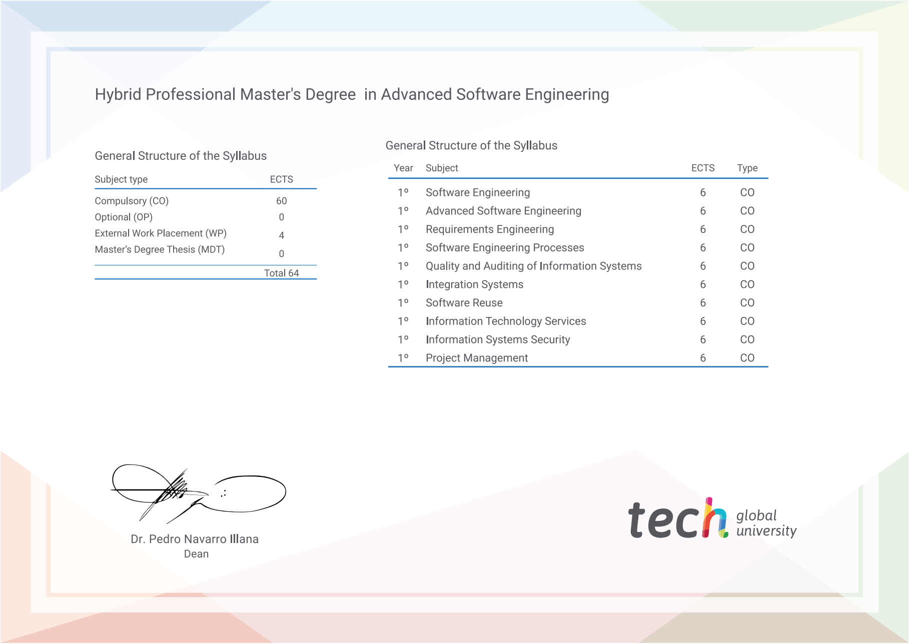
This is a private qualification of 1,920 hours of duration equivalent to 64 ECTS, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH Global University is a university officially recognized by the Government of Andorra on the 31st of January of 2024, which belongs to the European Higher Education Area (EHEA).

In Andorra la Vella, on the 28th of February of 2024

Dr. Pedro Navarro Illana
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates



Hybrid Professional Master's Degree in Advanced Software Engineering

General Structure of the Syllabus

Subject type	ECTS
Compulsory (CO)	60
Optional (OP)	0
External Work Placement (WP)	4
Master’s Degree Thesis (MDT)	0
	Total 64

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Software Engineering	6	CO
1º	Advanced Software Engineering	6	CO
1º	Requirements Engineering	6	CO
1º	Software Engineering Processes	6	CO
1º	Quality and Auditing of Information Systems	6	CO
1º	Integration Systems	6	CO
1º	Software Reuse	6	CO
1º	Information Technology Services	6	CO
1º	Information Systems Security	6	CO
1º	Project Management	6	CO

Dr. Pedro Navarro Illana
Dean

*Apostille Convention. In the event that the student wishes to have their paper diploma issued with an apostille, TECH Global University will make the necessary arrangements to obtain it, at an additional cost.

tech global university

Hybrid Professional Master's Degree

Advanced Software Engineering

Modality: **Hybrid (Online + Internship)**

Duration: **12 months**

Certificate: **TECH Global University**

Accreditation: **60 + 4 ECTS**

Hybrid Professional Master's Degree
Advanced Software Engineering

tech global university