Advanced Master's Degree
Software Engineering

tech global university

tech global university

Advanced Master's Degree
Software Engineering

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Global University**
- Accreditation: **120 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

Website: www.techtitute.com/us/information-technology/advanced-master-degree/advanced-master-degree-software-engineering

Index

01

Introduction

The demand for software has grown exponentially in recent years. The emergence of new digital platforms, advanced hardware and the increasing virtualization of everyday processes have posed constant challenges for software engineers. With an audience increasingly accustomed to emerging technologies and with higher expectations, developers must adapt and create solutions that respond to market demands. This requires technical proficiency in multiple areas of IT knowledge. For this reason, TECH has designed this program, which addresses essential competencies to meet today's challenges, including software planning and design, implementation of security measures, software reuse methods and web server administration.

Play a key role in the technological future. Specialize in Software Engineering and create transformative systems"

Software Engineering is positioned as one of the most relevant disciplines in the current technological era, as virtually all people interact daily with digital devices, driving a significant impact on global economies. In this context, the role of software engineers acquires great relevance, since the combination of technical knowledge and innovative methodologies allows the creation of effective solutions adapted to market needs. High specialization in this field is essential to ensure the success of technological projects, which is why TECH has designed this comprehensive academic program, designed to provide professionals with the most advanced and updated tools.

With this approach, the program addresses the key concepts of Software Engineering, delving into fundamental aspects such as application development, technology platforms and the most advanced methodologies for project management. The technical part acquires a relevant role, guiding the student from the conception of ideas to their implementation in functional systems. A unique opportunity to understand software architecture and the creation of innovative solutions, while exploring the latest trends in programming, mobile development and cloud applications. In addition, to strengthen leadership skills, the content includes the most current concepts in project management and agile methodologies, an added value for experienced professionals looking to update their skills or for those who wish to reach strategic positions for the first time.

One of the main advantages of this program is its 100% online modality, which eliminates the need to travel or adapt to rigid schedules. This allows students to manage their learning autonomously, organizing their time and pace of study according to their own needs, which is ideal for combining it with other daily responsibilities.

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

- The development of case studies presented by experts in Software Engineering
- The graphic, schematic, and practical contents with which they are created, provide scientific and practical information on the disciplines that are essential for professional practice
- Practical exercises where the self-assessment process can be carried out to improve learning
- Its special emphasis on innovative methodologies in Software Engineering Management
- 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

Drive technological development with efficient management in Software Engineering, key to organizational success"

Study at your own pace with a 100% online program, available at any time and from anywhere in the world"

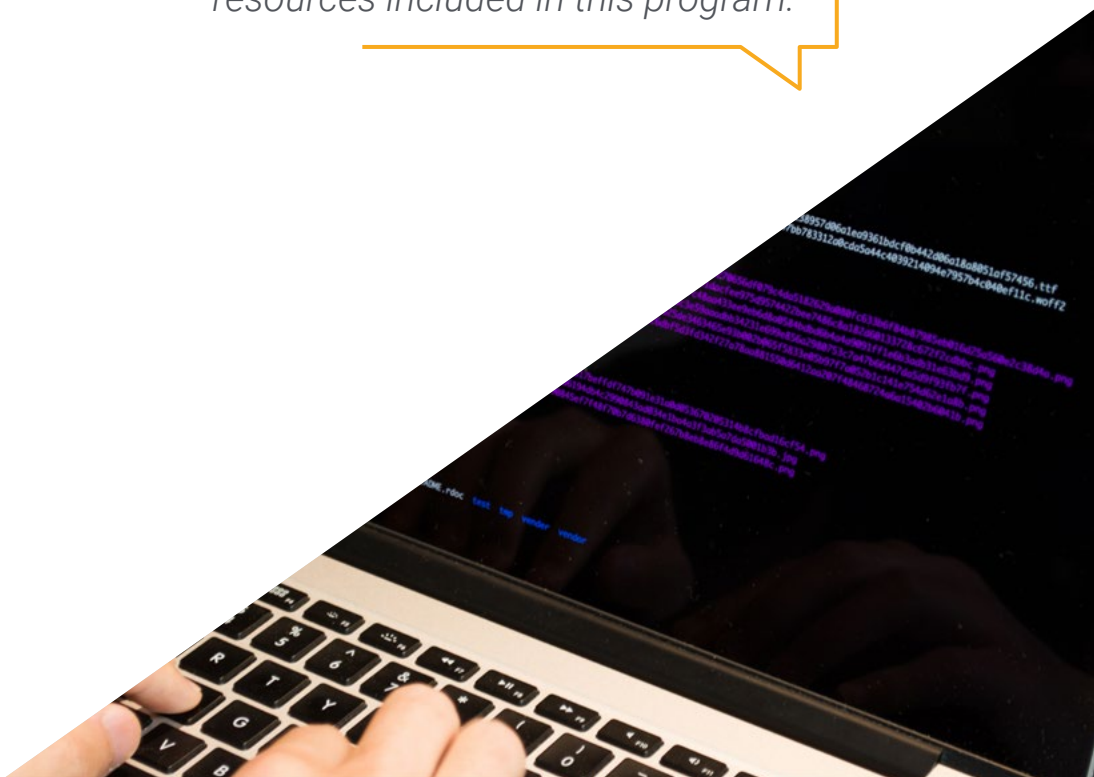
Access the most innovative and current methodology for learning in Software Engineering, designed by TECH.

Consolidate technical knowledge through a wide variety of practical resources included in this program.

It includes in its teaching staff professionals belonging to the field of Sofrware Engineering, who pour into this program the experience of their work, in addition to recognized specialists from reference societies and prestigious universities.

The multimedia content, developed with the latest educational technology, will provide the professional with situated and contextual learning, i.e., a simulated environment that will provide an immersive learning experience designed to prepare for real-life situations.

This program is designed around Problem-Based Learning, whereby the student must try to solve the different professional practice situations that arise throughout the program. For this purpose, the professional will be assisted by an innovative interactive video system created by renowned and experienced experts.



02

Why Study at TECH?

TECH is the world's largest online university. With an impressive catalog of more than 14,000 university programs, available in 11 languages, it is positioned as a leader in employability, with a 99% job placement rate. In addition, it has a huge faculty of more than 6,000 professors of the highest international prestige.

Study at the largest online university in the world and ensure your professional success. The future begins at TECH"

The world's best online university, according to FORBES

The prestigious Forbes magazine, specialized in business and finance, has highlighted TECH as "the best online university in the world" This is what they have recently stated in an article in their digital edition in which they echo the success story of this institution, "thanks to the academic offer it provides, the selection of its teaching staff, and an innovative learning method oriented to form the professionals of the future.

The best top international faculty

TECH's faculty is made up of more than 6,000 professors of the highest international prestige. Professors, researchers and top executives of multinational companies, including Isaiah Covington, performance coach of the Boston Celtics; Magda Romanska, principal investigator at Harvard MetaLAB; Ignacio Wistumba, chairman of the department of translational molecular pathology at MD Anderson Cancer Center; and D.W. Pine, creative director of TIME magazine, among others.

The world's largest online university

TECH is the world's largest online university. We are the largest educational institution, with the best and widest digital educational catalog, one hundred percent online and covering most areas of knowledge. We offer the largest selection of our own degrees and accredited online undergraduate and postgraduate degrees. In total, more than 14,000 university programs, in eleven different languages, making us the largest educational institution in the world.

The most complete syllabuses on the university scene

TECH offers the most complete syllabuses on the university scene, with programs that cover fundamental concepts and, at the same time, the main scientific advances in their specific scientific areas. In addition, these programs are continuously updated to guarantee students the academic vanguard and the most demanded professional skills. and the most in-demand professional competencies. In this way, the university's qualifications provide its graduates with a significant advantage to propel their careers to success.

A unique learning method

TECH is the first university to use Relearning in all its programs. This is the best online learning methodology, accredited with international teaching quality certifications, provided by prestigious educational agencies. In addition, this innovative academic model is complemented by the "Case Method", thereby configuring a unique online teaching strategy. Innovative teaching resources are also implemented, including detailed videos, infographics and interactive summaries.

The official online university of the NBA

TECH is the official online university of the NBA. Thanks to our agreement with the biggest league in basketball, we offer our students exclusive university programs, as well as a wide variety of educational resources focused on the business of the league and other areas of the sports industry. Each program is made up of a uniquely designed syllabus and features exceptional guest hosts: professionals with a distinguished sports background who will offer their expertise on the most relevant topics.

Leaders in employability

TECH has become the leading university in employability. Ninety-nine percent of its students obtain jobs in the academic field they have studied within one year of completing any of the university's programs. A similar number achieve immediate career enhancement. All this thanks to a study methodology that bases its effectiveness on the acquisition of practical skills, which are absolutely necessary for professional development.

Google Premier Partner

The American technology giant has awarded TECH the Google Premier Partner badge. This award, which is only available to 3% of the world's companies, highlights the efficient, flexible and tailored experience that this university provides to students. The recognition not only accredits the maximum rigor, performance and investment in TECH's digital infrastructures, but also places this university as one of the world's leading technology companies.

The top-rated university by its students

Students have positioned TECH as the world's top-rated university on the main review websites, with a highest rating of 4.9 out of 5, obtained from more than 1,000 reviews. These results consolidate TECH as the benchmark university institution at an international level, reflecting the excellence and positive impact of its educational model.

03

Syllabus

The curriculum of this program in Software Engineering has been carefully designed by a team of experts in computer science and technological development. In addition, throughout the program, the fundamental aspects of Software Engineering are comprehensively addressed, covering from planning, design and development of applications to advanced computer security and efficient server administration. Students will acquire specialized technical skills and learn to lead complex projects that respond to the demands of a constantly evolving market.

Contribute to technological advancement by developing innovative and effective solutions in Software Engineering"

Module 1. Methodologies, Development and Quality in Software Engineering

1.1. Introduction to Software Engineering
- 1.1.1. Introduction
- 1.1.2. The Software Crisis
- 1.1.3. Differences between Software Engineering and Computer Science
- 1.1.4. Ethics and Professional Responsibility in Software Engineering
- 1.1.5. Software Factories

1.2. The Software Development Process
- 1.2.1. Definition
- 1.2.2. Software Process Model
- 1.2.3. The Unified Software Development Process

1.3. Object-Oriented Software Development
- 1.3.1. Introduction
- 1.3.2. Principles of Object Orientation
- 1.3.3. Objectives Definition
- 1.3.4. Class Definition
- 1.3.5. Object-Oriented Analysis vs. Object-Oriented Design

1.4. Model-Based Software Development
- 1.4.1. The Need to Model
- 1.4.2. Software Systems Modeling
- 1.4.3. Object Modeling
- 1.4.4. UML
- 1.4.5. CASE Tools

1.5. Application Modeling and Design Patterns with UML
- 1.5.1. Advanced Requirements Modeling
- 1.5.2. Advanced Static Modeling
- 1.5.3. Advanced Dynamic Modeling
- 1.5.4. Component Modeling
- 1.5.5. Introduction to Design Patterns with UML
- 1.5.6. Adapter
- 1.5.7. Factory
- 1.5.8. Singleton
- 1.5.9. Strategy
- 1.5.10 Composite
- 1.5.11 Facade
- 1.5.12 Observer

1.6. Model-Driven Engineering
- 1.6.1. Introduction
- 1.6.2. Metamodeling of Systems
- 1.6.3. MDA
- 1.6.4. DSL
- 1.6.5. Model Refinements with OCL
- 1.6.6. Model Transformations

1.7. Ontologies in Software Engineering
- 1.7.1. Introduction
- 1.7.2. Ontology Engineering
- 1.7.3. Application of Ontologies in Software Engineering

1.8. Agile Methodologies for Software Development, Scrum
- 1.8.1. What Is Software Agility?
- 1.8.2. The Agile Manifesto
- 1.8.3. The Roadmap of an Agile Project
- 1.8.4. The Product Owner
- 1.8.5. User Stories
- 1.8.6. Agile Planning and Estimating
- 1.8.7. Measurements in Agile Development
- 1.8.8. Introduction to Scrum
- 1.8.9. The Roles
- 1.8.10 Product Backlog
- 1.8.11 The Sprint
- 1.8.12 Meetings

1.9. Lean Software Development Methodology
- 1.9.1. Introduction
- 1.9.2. Kanban

1.10. Quality and Software Process Improvement
- 1.10.1. Introduction
- 1.10.2. Software Measurement
- 1.10.3. Software Testing
- 1.10.4. Software Process Quality Model: CMMI

Module 2. Software Project Management

2.1. Fundamental Concepts of Project Management and the Project Management Lifecycle
- 2.1.1. What Is a Project?
- 2.1.2. Common Methodology
- 2.1.3. What Is Project Management?
- 2.1.4. What Is a Project Plan?
- 2.1.5. Benefits
- 2.1.6. Project Life Cycle
- 2.1.7. Process Groups or Project Management Life Cycle
- 2.1.8. The Relationship between Process Groups and Knowledge Areas
- 2.1.9. Relationships between Product and Project Life Cycle

2.2. Start-Up and Planning
- 2.2.1. From the Idea to the Project
- 2.2.2. Development of the Project Record
- 2.2.3. Project Kick-Off Meeting
- 2.2.4. Tasks, Knowledge and Skills in the Startup Process
- 2.2.5. The Project Plan
- 2.2.6. Development of the Basic Plan. Steps
- 2.2.7. Tasks, Knowledge and Skills in the Planning Process

2.3. Stakeholders and Outreach Management
- 2.3.1. Identify Stakeholders
- 2.3.2. Develop Plan for Stakeholder Management
- 2.3.3. Manage Stakeholder Engagement
- 2.3.4. Control Stakeholder Engagement
- 2.3.5. The Objective of the Project
- 2.3.6. Scope Management and Its Plan
- 2.3.7. Gathering Requirements
- 2.3.8. Define the Scope Statement
- 2.3.9. Create the WBS
- 2.3.10 Verify and Control the Scope

2.4. The Development of the Time-Schedule
- 2.4.1. Time Management and Its Plan
- 2.4.2. Defining Activities
- 2.4.3. Establishment of the Sequence of Activities
- 2.4.4. Estimated Resources for Activities
- 2.4.5. Estimated Duration of Activities
- 2.4.6. Development of the Time-Schedule and Calculation of the Critical Path
- 2.4.7. Schedule Control

2.5. Budget Development and Risk Response
- 2.5.1. Estimate Costs
- 2.5.2. Develop Budget and S-Curve
- 2.5.3. Cost Control and Earned Value Method
- 2.5.4. Risk Concepts
- 2.5.5. How to Perform a Risk Analysis
- 2.5.6. The Development of the Response Plan

2.6. Quality Management
- 2.6.1. Quality Planning
- 2.6.2. Assuring Quality
- 2.6.3. Quality Control
- 2.6.4. Basic Statistical Concepts
- 2.6.5. Quality Management Tools

2.7. Communication and Human Resources
- 2.7.1. Planning Communications Management
- 2.7.2. Communications Requirements Analysis
- 2.7.3. Communication Technology
- 2.7.4. Communication Models

2.7.5. Communication Methods
2.7.6. Communications Management Plan
2.7.7. Managing Communications
2.7.8. Management of Human Resources
2.7.9. Main Stakeholders and their Roles in the Projects
2.7.10 Types of Organization
2.7.11 Project Organization
2.7.12. The Work Equipment

2.8. Procurement
2.8.1. The Procurement Process
2.8.2. Planning
2.8.3. Search for Suppliers and Request for Quotations
2.8.4. Contract Allocation
2.8.5. Contract Administration
2.8.6. Contracts
2.8.7. Types of Contracts
2.8.8. Contract Negotiation

2.9. Execution, Monitoring and Control and Closure
2.9.1. Process Groups
2.9.2. Project Execution
2.9.3. Project Monitoring and Control
2.9.4. Project Closure

2.10. Professional Responsibility
2.10.1. Professional Responsibility
2.10.2. Characteristics of Social and Professional Responsibility
2.10.3. Project Leader Code of Ethics
2.10.4. Liability vs. PMP®
2.10.5. Examples of Liability
2.10.6. Benefits of Professionalization

Module 3. Software Development Platforms

3.1. Introduction to Application Development
3.1.1. Desktop Applications
3.1.2. Programming Language
3.1.3. Integrated Development Environments
3.1.4. Web Applications
3.1.5. Mobile Applications
3.1.6. Cloud Applications

3.2. Application Development and Graphical User Interface in Java
3.2.1. Integrated Development Environments for Java
3.2.2. Main IDE for Java
3.2.3. Introduction to the Eclipse Development Platform
3.2.4. Introduction to the NetBeans Development Platform
3.2.5. Controller View Model for Graphical User Interfaces
3.2.6. Design a Graphical Interface in Eclipse
3.2.7. Design a Graphical Interface in NetBeans

3.3. Debugging and Testing in Java
3.3.1. Testing and Debugging of Java Programs
3.3.2. Debugging in Eclipse
3.3.3. Debugging in NetBeans

3.4. Application Development and Graphical User Interface in .NET
3.4.1. Net Framework
3.4.2. Components of the .NET Development Platform
3.4.3. Visual Studio .NET
3.4.4. .NET tools for GUI
3.4.5. The GUI with Windows Presentation Foundation
3.4.6. Debugging and Compiling a WPF Application

3.5. Programming for .NET Networks
- 3.5.1. Introduction to .NET Network Programming
- 3.5.2. Requests and Responses in .NET
- 3.5.3. Use of Application Protocols in .NET
- 3.5.4. Security in .NET Network Programming

3.6. Mobile Application Development Environments
- 3.6.1. Mobile Applications
- 3.6.2. Android Mobile Applications
- 3.6.3. Steps for Development in Android
- 3.6.4. The IDE Android Studio

3.7. Development of Applications in the Environment Android Studio
- 3.7.1. Installing and Starting Android Studio
- 3.7.2. Running an Android Application
- 3.7.3. Development of the Graphic Interface in Android Studio
- 3.7.4. Starting Activities in Android Studio

3.8. Debugging and Publishing of Android Applications
- 3.8.1. Debugging an Application in Android Studio
- 3.8.2. Memorizing Applications in Android Studio
- 3.8.3. Publishing an Application on Google Play

3.9. Cloud Application Development
- 3.9.1. Cloud Computing
- 3.9.2. Cloud Levels: SaaS, PaaS, IaaS
- 3.9.3. Main Development Platforms in the Cloud
- 3.9.4. Bibliographical References

3.10. Introduction to Google Cloud Platform
- 3.10.1. Basic Concepts of Google Cloud Platform
- 3.10.2. Google Cloud Platform Services
- 3.10.3. Tools in Google Cloud Platform

Module 4. Web-Client Computing

4.1. Introduction to HTML
- 4.1.1. Structure of the Document
- 4.1.2. Color
- 4.1.3. Text
- 4.1.4. Hypertext Links
- 4.1.5. Images
- 4.1.6. Lists
- 4.1.7. Tables
- 4.1.8. Frames
- 4.1.9. Forms
- 4.1.10 Specific Elements for Mobile Technologies
- 4.1.11. Obsolete Elements

4.2. Cascading Style Sheets (CSS)
- 4.2.1. Elements and Structure of a Cascading Style Sheet
  - 4.2.1.1. Creation of Style Sheets
  - 4.2.1.2. Application of Styles Selectors
  - 4.2.1.3. Style Inheritance and Cascading
  - 4.2.1.4. Page Formatting Using Styles
  - 4.2.1.5. Page Structuring Using Styles. The Box Model
- 4.2.2. Style Design for different Devices
- 4.2.3. Types of Style Sheets: Static and Dynamic. The Pseudo-Classes
- 4.2.4. Best Practices in the Use of Style Sheets

4.3. Introduction and History of JavaScript
- 4.3.1. Introduction
- 4.3.2. History of JavaScript
- 4.3.3. Development Environment to Be Used

4.4. Basic Notions of Web Programming
- 4.4.1. Basic JavaScript Syntax
- 4.4.2. Primitive Data Types and Operators
- 4.4.3. Variables and Areas
- 4.4.4. Text Strings and Template Literals
- 4.4.5. Numbers and Booleans
- 4.4.6. Comparisons

4.5. Complex JavaScript Structures
- 4.5.1. Vectors or Arrays and Objects
- 4.5.2. Sets
- 4.5.3. Maps
- 4.5.4. Disjunctive
- 4.5.5. Loops

4.6. Functions and Objects
- 4.6.1. Function Definition and Invocation
- 4.6.2. Arguments
- 4.6.3. Arrow Functions
- 4.6.4. Callback Functions
- 4.6.5. Higher Order Functions
- 4.6.6. Literal Objects
- 4.6.7. The This Object
- 4.6.8. Objects as Namespaces: the Math and Date Objects

4.7. The Document Object Model (DOM)
- 4.7.1. What Is DOM?
- 4.7.2. A Bit of History
- 4.7.3. Navigation and Element Retrieval
- 4.7.4. A Virtual DOM with JSDOM
- 4.7.5. Query Selectors
- 4.7.6. Navigation using Properties
- 4.7.7. Assigning Attributes to Elements
- 4.7.8. Creation and Modification of Nodes
- 4.7.9. Updated Styling of the DOM Elements

4.8. Modern Web Development
- 4.8.1. Event-Driven Flow and Listeners
- 4.8.2. Modern Web Toolkits and Alignment Systems
- 4.8.3. Strict JavaScript Mode
- 4.8.4. More about Functions
- 4.8.5. Asynchronous Promises and Functions
- 4.8.6. Closures
- 4.8.7. Functional Programming
- 4.8.8. POO in JavaScript

4.9. Web Usability
- 4.9.1. Introduction to Usability
- 4.9.2. Definition of Usability
- 4.9.3. Importance of User-Centered Web Design
- 4.9.4. Differences Between Accessibility and Usability
- 4.9.5. Advantages and Problems in Combining Accessibility and Usability
- 4.9.6. Advantages and Difficulties in the Implementation of Usable Websites
- 4.9.7. Usability Methods
- 4.9.8. User Requirements Analysis
- 4.9.9. Conceptual Design Principles. User-Oriented Prototyping
- 4.9.10 Guidelines for the Creation of Usable Web Sites
  - 4.9.10.1. Usability Guidelines of Jakob Nielsen
  - 4.9.10.2. Usability Guidelines of Bruce Tognazzini
- 4.9.11. Usability Evaluation

4.10. Web Accessibility
- 4.10.1. Introduction
- 4.10.2. Definition of Web-Accessibility
- 4.10.3. Types of Disabilities
  - 4.10.3.1. Temporary or Permanent Disabilities
  - 4.10.3.2. Visual Impairment
  - 4.10.3.3. Hearing Impairment
  - 4.10.3.4. Motor Impairment
  - 4.10.3.5. Neurological or Cognitive Disabilities

4.10.3.6. Difficulties Arising from Aging
4.10.3.7. Limitations Arising from the Environment
4.10.3.8. Barriers Preventing Access to the Web
4.10.4. Technical Aids and Support Products to Overcome Barriers
4.10.4.1. Aids for the Blind
4.10.4.2. Aids for Those with Low Vision
4.10.4.3. Aids for Those with Color Blindness
4.10.4.4. Aids for the Hearing Impaired
4.10.4.5. Aids for the Motor Impaired
4.10.4.6. Aids for Those with Cognitive and Neurological Disabilities
4.10.5. Advantages and Difficulties in the Implementation of Web Accessibility
4.10.6. Web Accessibility Regulations and Standards
4.10.7. Web Accessibility Regulatory Bodies
4.10.8. Comparison of Standards and Regulations
4.10.9. Guidelines for Compliance with Regulations and Standards
4.10.9.1. Description of the Main Guidelines (Images, links, videos, etc.)
4.10.9.2. Guidelines for Accessible Navigation
4.10.9.2.1. Perceptibility
4.10.9.2.2. Operability
4.10.9.2.3. Comprehensibility
4.10.9.2.4. Robustness
4.10.10 Description of the Web Accessibility Compliance Process
4.10.11. Compliance Levels
4.10.12. Compliance Criteria
4.10.13. Compliance Requirements
4.10.14. Web Site Accessibility Evaluation Methodology

Module 5. Web Server Computing

5.1. Introduction to Server-Side Programming: PHP
5.1.1. Server-Side Programming Basics
5.1.2. Basic PHP Syntax
5.1.3. HTML Content Generation with PHP
5.1.4. Development and Testing Environments: XAMPP
5.2. Advanced PHP
5.2.1. Control Structures with PHP
5.2.2. PHP Functions
5.2.3. Array Handling in PHP
5.2.4. String Handling with PHP
5.2.5. Object Orientation in PHP
5.3. Data Models
5.3.1. Concept of Data. Life Cycle of Data
5.3.2. Types of Data
5.3.2.1. Basic
5.3.2.2. Records
5.3.2.3. Dynamics
5.4. Relational Model
5.4.1. Description
5.4.2. Entities and Types of Entities
5.4.3. Data Elements. Attributes
5.4.4. Relationships: Types, Subtypes, Cardinality
5.4.5. Keys. Types of Keys
5.4.6. Normalization. Normal Shapes
5.5. Construction of the Logical Data Model
5.5.1. Specification of Tables
5.5.2. Definition of Columns
5.5.3. Key Specification
5.5.4. Conversion to Normal Shapes. Dependency

5.6. The Physical Data Model. Data Files
- 5.6.1. Description of Data Files
- 5.6.2. Types of Files
- 5.6.3. Access Modes
- 5.6.4. File Organization

5.7. Database Access from PHP
- 5.7.1. Introduction to MariaDB
- 5.7.2. Working with a MariaDB Database: the SQL Language
- 5.7.3. Accessing the MariaDB Database from PHP
- 5.7.4. Introduction to MySql
- 5.7.5. Working with a MySql Database: The SQL Language
- 5.7.6. Accessing MySql Database from PHP

5.8. Client Interaction from PHP
- 5.8.1. PHP Forms
- 5.8.2. Cookies
- 5.8.3. Session Management

5.9. Web Application Architecture
- 5.9.1. The Controller View Model Pattern
- 5.9.2. Controller
- 5.9.3. Models
- 5.9.4. View

5.10. Introduction to Web Services
- 5.10.1. Introduction to XML
- 5.10.2. Service-Oriented Architecture (SOA): Web Services
- 5.10.3. Creation of SOAP and REST Web Services
- 5.10.4. The SOAP Protocol
- 5.10.5. The REST Protocol

Module 6. Security Management

6.1. Information Security
- 6.1.1. Introduction
- 6.1.2. Information Security Involves Confidentiality, Integrity and Availability
- 6.1.3. Security Is an Economic Issue
- 6.1.4. Security Is a Process
- 6.1.5. Classification of Information
- 6.1.6. Information Security Involves Risk Management
- 6.1.7. Security is Articulated with Security Controls
- 6.1.8. Security Is Both Physical and Logical
- 6.1.9. Security Involves People

6.2. The Information Security Professional
- 6.2.1. Introduction
- 6.2.2. Information Security as a Profession
- 6.2.3. Certifications (ISC)2
- 6.2.4. The ISO 27001 Standard
- 6.2.5. Best Security Practices in IT Service Management
- 6.2.6. Information Security Maturity Models
- 6.2.7. Other Certifications, Standards and Professional Resources

6.3. Access Control
- 6.3.1. Introduction
- 6.3.2. Access Control Requirements
- 6.3.3. Authentication Mechanisms
- 6.3.4. Authorization Methods
- 6.3.5. Access Accounting and Auditing
- 6.3.6. Triple A Technologies

6.4. Information Security Programs, Processes and Policies
- 6.4.1. Introduction
- 6.4.2. Security Management Programs
- 6.4.3. Risk Management
- 6.4.4. Design of Security Policies

6.5. Business Continuity Plans
- 6.5.1. Introduction to BCPs
- 6.5.2. Phase I and II
- 6.5.3. Phase III and IV
- 6.5.4. Maintenance of the BCP

6.6. Procedures for the Correct Protection of the Company
- 6.6.1. DMZ Networks
- 6.6.2. Intrusion Detection Systems
- 6.6.3. Access Control Lists
- 6.6.4. Learning from the Attacker: Honeypot

6.7. Security Architecture Prevention
- 6.7.1. Overview. Activities and Layer Model
- 6.7.2. Perimeter Defense (Firewalls, WAFs, WAFs, IPS, etc.)
- 6.7.3. Endpoint Defense (Equipment, Servers and Services)

6.8. Security Architecture. Detection
- 6.8.1. Overview Detection and Monitoring
- 6.8.2. Logs, Encrypted Traffic Breaking, Recording and Siems
- 6.8.3. Alerts and Intelligence

6.9. Security Architecture. Reaction
- 6.9.1. Reaction. Products, Services and Resources
- 6.9.2. Incident Management
- 6.9.3. CERTS and CSIRTs

6.10. Security Architecture. Recovery
- 6.10.1. Resilience, Concepts, Business Requirements and Regulations
- 6.10.2. IT Resilience Solutions
- 6.10.3. Crisis Management and Governance

Module 7. Information Systems Security

7.1. A Global Perspective on Security, Cryptography and Classical Cryptanalysis
- 7.1.1. IT Security: Historical Perspective
- 7.1.2. But What Exactly Is Meant by Security?
- 7.1.3. History of Cryptography
- 7.1.4. Substitution Ciphers
- 7.1.5. Study Case: The Enigma Machine

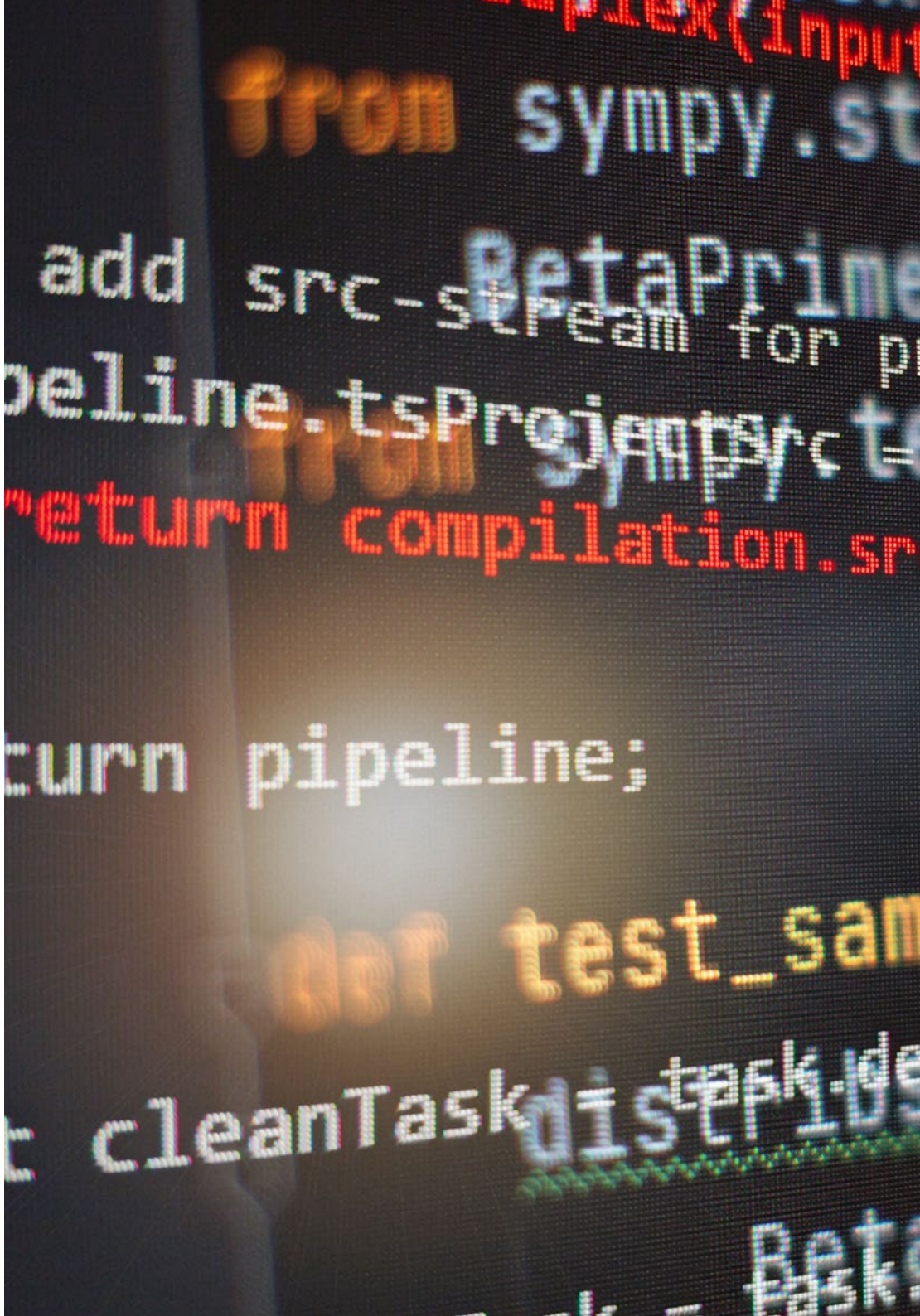
7.2. Symmetric Cryptography
- 7.2.1. Introduction and Basic Terminology.
- 7.2.2. Symmetric Encryption
- 7.2.3. Modes of Operation
- 7.2.4. DES
- 7.2.5. The New AES Standard
- 7.2.6. Encryption in Flow
- 7.2.7. Cryptanalysis

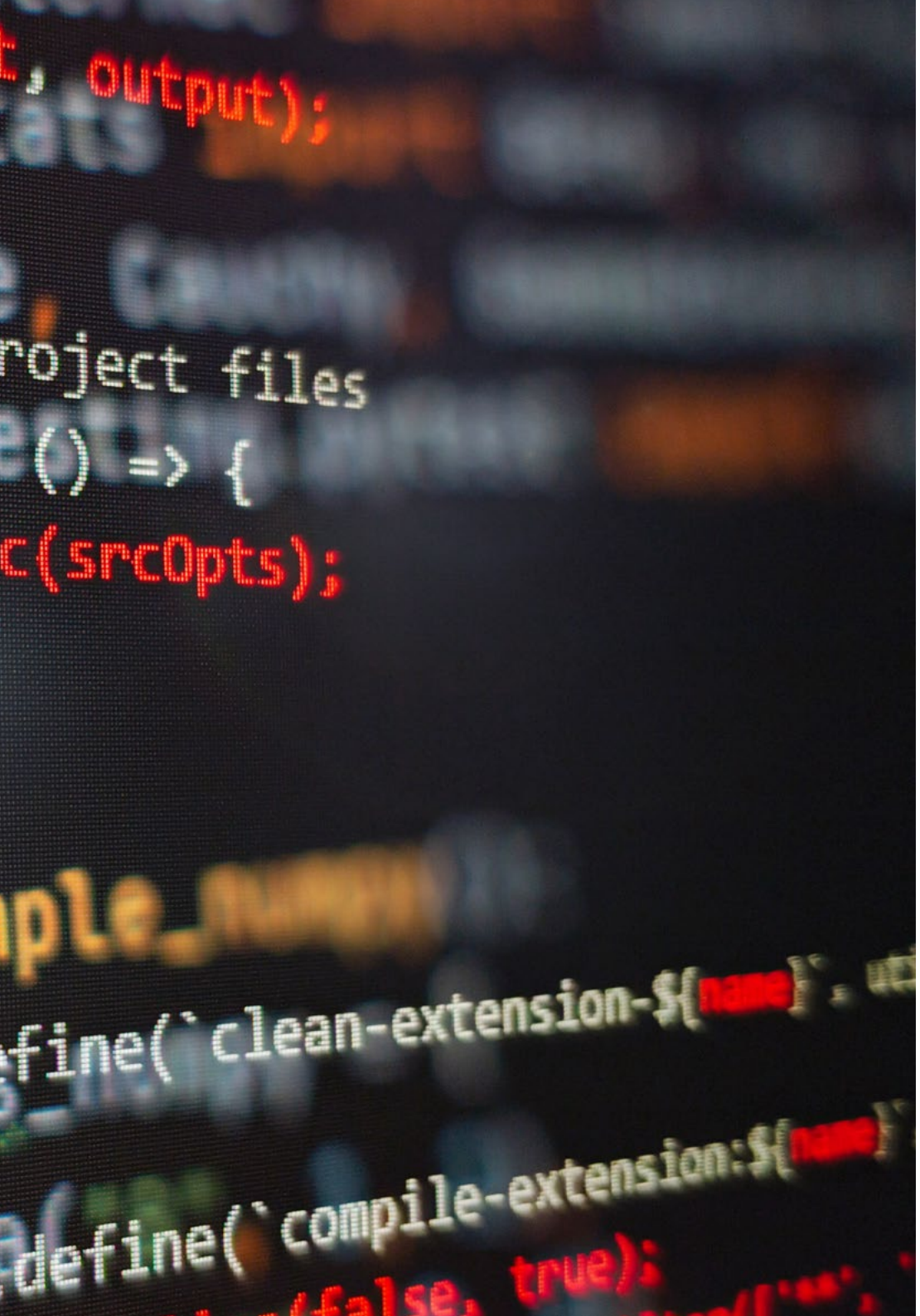
7.3. Asymmetric Cryptography
- 7.3.1. Origins of Public Key Cryptography
- 7.3.2. Basic Concepts and Operation
- 7.3.3. The RSA Algorithm
- 7.3.4. Digital Certificates
- 7.3.5. Key Storage and Management

7.4. Network Attacks
- 7.4.1. Network Threats and Attacks
- 7.4.2. Enumeration
- 7.4.3. Traffic Interception: Sniffers
- 7.4.4. Denial of Service Attacks
- 7.4.5. ARP Poisoning Attacks

7.5. Security Architectures
- 7.5.1. Traditional Security Architectures
- 7.5.2. Secure Socket Layer: SSL
- 7.5.3. SSH Protocol
- 7.5.4. Virtual Private Networks (VPNs)
- 7.5.5. External Storage Unit Protection Mechanisms
- 7.5.6. Hardware Protection Mechanisms

7.6. System Protection Techniques and Secure Code Development
- 7.6.1. Operational Safety
- 7.6.2. Resources and Controls
- 7.6.3. Monitoring
- 7.6.4. Intrusion Detection Systems
- 7.6.5. Host IDS
- 7.6.6. Network IDS
- 7.6.7. Signature-Based IDS
- 7.6.8. Lure Systems
- 7.6.9. Basic Security Principles in Code Development
- 7.6.10. Failure Management
- 7.6.11. Public Enemy Number 1: Buffer Overflows
- 7.6.12. Cryptographic Botches





7.7. Botnets and Spam
- 7.7.1. Origin of the Problem
- 7.7.2. Spam Process
- 7.7.3. Sending Spam
- 7.7.4. Refinement of Mailing Lists
- 7.7.5. Protection Techniques
- 7.7.6. Anti-Spam Service Offered by Third-Parties
- 7.7.7. Study Cases
- 7.7.8. Exotic Spam

7.8. Web Auditing and Attacks
- 7.8.1. Information Gathering
- 7.8.2. Attack Techniques
- 7.8.3. Tools

7.9. Malware and Malicious Code
- 7.9.1. What Is Malware?
- 7.9.2. Types of Malware
- 7.9.3. Virus
- 7.9.4. Criptovirus
- 7.9.5. Worms
- 7.9.6. Adware
- 7.9.7. Spyware
- 7.9.8. Hoaxes
- 7.9.9. Phishing
- 7.9.10. Trojans
- 7.9.11. The Economy of Malware
- 7.9.12. Possible Solutions

7.10. Forensic Analysis
- 7.10.1. Evidence Collection
- 7.10.2. Evidence Analysis
- 7.10.3. Anti-Forensic Techniques
- 7.10.4. Case Study

Module 8. Software Security

8.1. Software Security Problems
- 8.1.1. Introduction to the Problem of Software Security
- 8.1.2. Vulnerabilities and their Classification
- 8.1.3. Secure Software Properties
- 8.1.4. References

8.2. Software Safety Design Principles
- 8.2.1. Introduction
- 8.2.2. Software Safety Design Principles
- 8.2.3. Types of S-SDLC
- 8.2.4. Software Safety in S-SDLC Phases
- 8.2.5. Methodologies and Standards
- 8.2.6. References

8.3. Software Lifecycle Safety in the Requirements and Design Phases
- 8.3.1. Introduction
- 8.3.2. Attack Modeling
- 8.3.3. Cases of Abuse
- 8.3.4. Safety Requirements Engineering
- 8.3.5. Risk Analysis. Architectural
- 8.3.6. Design Patterns
- 8.3.7. References

8.4. Software Lifecycle Safety in the Coding, Testing and Operation Phases
- 8.4.1. Introduction
- 8.4.2. Risk-Based Safety Testing
- 8.4.3. Code Review
- 8.4.4. Penetration Test
- 8.4.5. Security Operations
- 8.4.6. External Review
- 8.4.7. References

8.5. Secure Coding Applications I
- 8.5.1. Introduction
- 8.5.2. Secure Coding Practices
- 8.5.3. Manipulation and Validation of Inputs
- 8.5.4. Memory Overflow
- 8.5.5. References

8.6. Secure Coding Applications II
- 8.6.1. Introduction
- 8.6.2. Integers Overflows, Truncation Errors and Problems with Type Conversions between Integers
- 8.6.3. Errors and Exceptions
- 8.6.4. Privacy and Confidentiality
- 8.6.5. Privileged Programs
- 8.6.6. References

8.7. Development and Cloud Security
- 8.7.1. Safety in Development; Methodology and Practice
- 8.7.2. PaaS, IaaS, CaaS and SaaS Models
- 8.7.3. Security in the Cloud and for Cloud Services

8.8. Security Automation and Orchestration (SOAR)
- 8.8.1. Complexity of Manual Processing; Need to Automate Tasks
- 8.8.2. Products and Services
- 8.8.3. SOAR Architecture

8.9. Telework Safety
- 8.9.1. Need and Scenarios
- 8.9.2. Products and Services
- 8.9.3. Telework Safety

Module 9. Quality and Auditing of Information Systems

9.1. Introduction to Information Security Management Systems
- 9.1.1. Fundamental Principles of ISMS
- 9.1.2. ISMS Golden Rules
- 9.1.3. Role of IT Audit in ISMSs

9.2. Safety Management Planning
- 9.2.1. Concepts Related to Safety Management
- 9.2.2. Classification of Information: Objectives, Concepts and Roles
- 9.2.3. Implementation of Security Policy Practices: Security Policies, Standards and Procedures
- 9.2.4. Risk Management: Information Asset Risk Principles and Analysis

9.3. Main Mechanisms for the Protection of Information Assets (I)
- 9.3.1. Summary of the Main Cryptographic Tools for the Protection of the CIA Triad
- 9.3.2. Consideration of Privacy, Anonymity and Adequate Management of User Traceability Requirements

9.4. Main Mechanisms for the Protection of Information Assets (II)
- 9.4.1. Communications Security: Protocols, Devices and Security Architectures
- 9.4.2. Operating System Security

9.5. ISMS Internal Controls
- 9.5.1. ISMS Controls Taxonomy: Administrative, Logical and Physical Controls
- 9.5.2. Classification of Controls according to How the Threat is Addressed: Controls for Threat Prevention, Detection and Correction
- 9.5.3. Implementation of Internal Control Systems in ISMSs

9.6. Types of Audits
- 9.6.1. Difference between Audit and Internal Control
- 9.6.2. Internal vs. External Audit
- 9.6.3. Audit Classification according to the Objective and Type of Analysis

9.7. Scriptwriter and Script: Subject and Object Protected by the Intellectual Property
- 9.7.1. Introduction to Penetration Testing and Forensic Analysis
- 9.7.2. Definition and Relevance of Fingerprinting and Footprinting Concepts

9.8. Vulnerability Scanning and Network Traffic Monitoring
- 9.8.1. Tools for Vulnerability Analysis in Systems
- 9.8.2. Main Vulnerabilities in the Context of Web Applications
- 9.8.3. Analysis of Communications Protocols

9.9. The IT Audit Process
- 9.9.1. Life Cycle Concept in Systems Development
- 9.9.2. Activity and Process Monitoring: Collection and Treatment of Evidence
- 9.9.3. IT Audit Methodology
- 9.9.4. IT Audit Process
- 9.9.5. Identification of the Main Crimes and Misdemeanors in the Context of Information Technologies
- 9.9.6. Computer Crime Investigation: Introduction to Forensic Analysis and Its Relation to Computer Auditing

9.10. Business Continuity and Disaster Recovery Plans
- 9.10.1. Definition of Business Continuity Plan and the Business Interruption Concept
- 9.10.2. NIST Recommendation on Business Continuity Plans
- 9.10.3. Disaster Recovery Plan
- 9.10.4. Disaster Recovery Plan Process

Module 10. Web Server Administration

10.1. Introduction to Web Servers
- 10.1.1. What Is a Web Server?
- 10.1.2. Architecture and Operation of a Web Server
- 10.1.3. Resources and Contents on a Web Server
- 10.1.4. Application Servers
- 10.1.5. Proxy Servers
- 10.1.6. Main Web Servers on the Market
- 10.1.7. Web Server Usage Statistics
- 10.1.8. Web Server Security
- 10.1.9. Load Balancing on Web Servers
- 10.1.10. References

10.2. HTTP Protocol Handling
- 10.2.1. Operation and Structure
- 10.2.2. Request Methods
- 10.2.3. Status Codes
- 10.2.4. Headers
- 10.2.5. Content Coding Code Pages
- 10.2.6. Performing HTTP Requests on the Internet using a Proxy, Livehttpheadersor Similar Method, Analyzing the Protocol Used

10.3. Description of Distributed Multi-Server Architectures
- 10.3.1. 3-Layer Model
- 10.3.2. Fault Tolerance
- 10.3.3. Load Sharing
- 10.3.4. Session State Stores
- 10.3.5. Cache Stores

10.4. Internet Information Services (IIS)
- 10.4.1. What Is IIS?
- 10.4.2. History and Evolution of IIS
- 10.4.3. Main Advantages and Features of IIS7 and Later Versions
- 10.4.4. IIS7 Architecture and Later Versions

10.5. IIS Installation, Administration and Configuration
- 10.5.1. Preamble
- 10.5.2. Internet Information Services (IIS) Installation
- 10.5.3. IIS Administration Tools
- 10.5.4. Web Site Creation, Configuration and Administration
- 10.5.5. Installation and Management of IIS Extensions

10.6. Advanced Security in IIS
- 10.6.1. Preamble
- 10.6.2. Authentication, Authorization and Access Control in IIS
- 10.6.3. Configuring a Secure Website on IIS with SSL
- 10.6.4. Security Policies Implemented in IIS 8.x

10.7. Introduction to Apache
- 10.7.1. What Is Apache?
- 10.7.2. Main Advantages of Apache
- 10.7.3. Main Features of Apache
- 10.7.4. Architecture

10.8. Apache Installation and Configuration
- 10.8.1. Initial Installation of Apache
- 10.8.2. Apache Configuration

10.9. Installation and Configuration of the Different Apache Modules
- 10.9.1. Apache Module Installation
- 10.9.2. Types of Modules
- 10.9.3. Secure Apache Configuration

10.10. Advanced Security
- 10.10.1. Authentication, Authorization and Access Control
- 10.10.2. Authentication Methods
- 10.10.3. Secure Apache Configuration with SSL

Module 11. Online Application Security

11.1. Vulnerabilities and Security Issues in Online Applications
- 11.1.1. Introduction to Online Application Security
- 11.1.2. Security Vulnerabilities in the Design of Web Applications
- 11.1.3. Security Vulnerabilities in the Implementation of Web Applications
- 11.1.4. Security Vulnerabilities in the Deployment of Web Applications
- 11.1.5. Official Lists of Security Vulnerabilities

11.2. Policies and Standards for Online Application Security
- 11.2.1. Pillars for the Security of Online Applications
- 11.2.2. Security Policy
- 11.2.3. Information Security Management System
- 11.2.4. Secure Software Development Life Cycle
- 11.2.5. Standards for Application Security

11.3. Security in the Design of Web Applications
- 11.3.1. Introduction to Web Application Security
- 11.3.2. Security in the Design of Web Applications

11.4. Testing the Online Safety and Security of Web Applications
- 11.4.1. Web Application Security Testing and Analysis
- 11.4.2. Web Application Deployment and Production Security

11.5. Web Services Security
- 11.5.1. Introduction to Web Services Security
- 11.5.2. Web Services Security Functions and Technologies

11.6. Testing the Online Safety and Security of Web Services
- 11.6.1. Evaluation of Web Services Security
- 11.6.2. Online Protection. Firewalls and XML Gateways

11.7. Ethical Hacking, Malware and Forensics
- 11.7.1. Ethical Hacking
- 11.7.2. Malware Analysis
- 11.7.3. Forensic Analysis

11.8. Incident Resolution on Web Services
- 11.8.1. Monitoring
- 11.8.2. Performance Measurement Tools
- 11.8.3. Containment Measures
- 11.8.4. Root Cause Analysis
- 11.8.5. Proactive Problem Management

11.9. Best Practices to Ensure Application Security
- 11.9.1. Handbook of Best Practices in the Development of Online Applications
- 11.9.2. Handbook of Good Practices in the Implementation of Online Applications

11.10. Common Errors that Undermine Application Security
- 11.10.1. Common Errors in Development
- 11.10.2. Common Errors in Hosting
- 11.10.3. Common Production Errors

Module 12. Software Engineering

12.1. Introduction to Software Engineering and Modeling
- 12.1.1. The Nature of Software
- 12.1.2. The Unique Nature of Webapps
- 12.1.3. Software Engineering
- 12.1.4. Software Process
- 12.1.5. Software Engineering Practice
- 12.1.6. Software Myths
- 12.1.7. How It All Begins
- 12.1.8. Object-Oriented Concepts
- 12.1.9. Introduction to UML

12.2. Software Process
- 12.2.1. A General Process Model
- 12.2.2. Prescriptive Process Models
- 12.2.3. Specialized Process Models
- 12.2.4. The Unified Process
- 12.2.5. Personal and Team Process Models
- 12.2.6. What Is Agility?
- 12.2.7. What Is an Agile Process?
- 12.2.8. Scrum
- 12.2.9. Agile Process Toolkit

12.3. Software Engineering Guiding Principles
- 12.3.1. Principles Guiding the Process
- 12.3.2. Principles Guiding the Practice
- 12.3.3. Principles of Communication
- 12.3.4. Planning Principles
- 12.3.5. Modeling Principles
- 12.3.6. Construction Principles
- 12.3.7. Deployment Principles

12.4. Understanding the Requirements
- 12.4.1. Requirements Engineering
- 12.4.2. Establish the Basis
- 12.4.3. Inquiry of Requirements
- 12.4.4. Development of Cases Studies
- 12.4.5. Elaboration of the Requirements Model
- 12.4.6. Negotiation of Requirements
- 12.4.7. Validation of Requirements

12.5. Requirements Modeling: Scenarios, Information and Types of Analysis
- 12.5.1. Analysis of Requirements
- 12.5.2. Scenario-Based Modeling
- 12.5.3. UML Models that Provide the Case Study
- 12.5.4. Data Modeling Concepts
- 12.5.5. Class-Based Modeling
- 12.5.6. Class Diagrams

12.6. Requirements Modeling: Flow, Behavior and Patterns
- 12.6.1. Requirements that Shape Strategies
- 12.6.2. Flow-Oriented Modeling
- 12.6.3. Status Diagrams
- 12.6.4. Creation of a Behavioral Model
- 12.6.5. Sequence Diagrams
- 12.6.6. Communication Diagrams
- 12.6.7. Patterns for Requirements Modeling

12.7. Design Concepts
- 12.7.1. Design in Software Engineering
- 12.7.2. The Design Process
- 12.7.3. Design Concepts
- 12.7.4. Object-Oriented Design Concepts
- 12.7.5. Model of the Design

12.8. Designing the Architecture
- 12.8.1. Software Design
- 12.8.2. Architectural Genres
- 12.8.3. Architectural Styles
- 12.8.4. Architectural Design
- 12.8.5. Evolution of Alternative Designs for Architecture
- 12.8.6. Mapping the Architecture Using the Data Flow

12.9. Component-Level and Pattern-Based Design
- 12.9.1. What Is a Component?
- 12.9.2. Class-Based Component Design
- 12.9.3. Realization of the Design at the Component Level
- 12.9.4. Design of Traditional Components
- 12.9.5. Component-Based Development
- 12.9.6. Design Patterns
- 12.9.7. Pattern-Based Software Design
- 12.9.8. Architectural Patterns
- 12.9.9. Design Patterns at the Component Level
- 12.9.10. User Interface Design Patterns

12.10. Software Quality and Project Management
- 12.10.1. Quality
- 12.10.1. Software Quality
- 12.10.2. The Software Quality Dilemma
- 12.10.3. Achieving Software Quality
- 12.10.4. Ensuring Software Quality
- 12.10.5. The Administrative Spectrum
- 12.10.6. The Personnel
- 12.10.7. The Product
- 12.10.8. The Process
- 12.10.9. The Project
- 12.10.10. Principles and Practices

Module 13. Advanced Software Engineering

13.1. Extreme Programming
- 13.1.1. Justification and Overview of XP
- 13.1.2. The XP Life Cycle
- 13.1.3. The Five Core Values
- 13.1.4. The Twelve Basic Practices in XP
- 13.1.5. Roles of Participants
- 13.1.6. XP Industrial
- 13.1.7. Critical Assessment of XP

13.2. Software Development Based on Reusability
- 13.2.1. Software Reuse
- 13.2.2. Code Reuse Levels
- 13.2.3. Specific Reuse Techniques
- 13.2.4. Component-Based Development
- 13.2.5. Benefits and Problems of Reuse
- 13.2.6. Reuse Planning

13.3. System Architecture and Software Design Patterns
- 13.3.1. Architectural Design
- 13.3.2. General Architectural Patterns
- 13.3.3. Fault Tolerant Architectures
- 13.3.4. Distributed Systems Architectures
- 13.3.5. Design Patterns
- 13.3.6. Gamma Patterns
- 13.3.7. Interaction Design Patterns

13.4. Cloud Application Architecture
- 13.4.1. Cloud Computing Fundamentals
- 13.4.2. Cloud Application Quality
- 13.4.3. Architectural Styles
- 13.4.4. Design Patterns

13.5. Software Testing: TDD, ATDD and BDD
- 13.5.1. Software Verification and Validation
- 13.5.2. Software Testing
- 13.5.3. Test Driven Development (TDD)
- 13.5.4. Acceptance Test Driven Development (ATDD)
- 13.5.5. Behavior Driven Development (BDD)
- 13.5.6. BDD and Cucumber

13.6. Software Process Improvement
- 13.6.1. Software Process Improvement
- 13.6.2. The Process Improvement Approach
- 13.6.3. Maturity Models
- 13.6.4. The CMMI Model
- 13.6.5. CMMI V2.0
- 13.6.6. CMMI and Agile

13.7. The Quality of the Software Product: SQuaRE
- 13.7.1. Software Quality
- 13.7.2. Software Product Quality Models
- 13.7.3. ISO/IEC 25000 Family
- 13.7.4. ISO/IEC 25010: Quality Model and Quality Characteristics
- 13.7.5. ISO/IEC 25012: the Quality of the Data
- 13.7.6. ISO/IEC 25020 Software Quality Measurement
- 13.7.7. ISO/IEC 25022, 25023 and 25024: Software and Data Quality Metrics
- 13.7.8. ISO/IEC 25040 Software Assessment
- 13.7.9. Accreditation Process

13.8. Introduction to DevOps
- 13.8.1. DevOps Concept
- 13.8.2. Core Practices

Module 14. Requirements Engineering

14.1. Introduction to Requirements Engineering
- 14.1.1. The Importance of Requirements
- 14.1.2. Concept of Requirement
- 14.1.3. Dimensions of Requirements
- 14.1.4. Levels and Types of Requirements
- 14.1.5. Requirements Characteristics
- 14.1.6. Requirements Engineering
- 14.1.7. The Requirements Engineering Process
- 14.1.8. Frameworks for Requirements Engineering
- 14.1.9. Best Practices in Requirements Engineering
- 14.1.10. The Business Analyst

14.2. Sources of Requirements
- 14.2.1. The Requirements Network
- 14.2.2. The Stakeholders
- 14.2.3. Business Requirements
- 14.2.4. Vision and Scope Document

14.3. Requirements Elicitation Techniques
- 14.3.1. Elicitation of Requirements
- 14.3.2. Problems of Requirements Elicitation
- 14.3.3. Contexts of Discovery
- 14.3.4. Interviews
- 14.3.5. Observation and Learning
- 14.3.6. Ethnography
- 14.3.7. Workshops
- 14.3.8. Focus Groups
- 14.3.9. Questionnaires
- 14.3.10. Brainstorming and Creative Techniques
- 14.3.11. Group Media
- 14.3.12. Analysis of System Interfaces
- 14.3.13. Document Analysis and "Archeology"
- 14.3.14. Case Studies and Scenarios
- 14.3.15. Prototypes
- 14.3.16. Reverse Engineering
- 14.3.17. Reuse of Requirements
- 14.3.18. Good Elicitation Practices

14.4. User Requirements
- 14.4.1. People
- 14.4.2. Case Studies and User Stories
- 14.4.3. Scenarios
- 14.4.5. Types of Scenarios
- 14.4.6. How to Discover Scenarios

14.5. Prototyping Techniques
- 14.5.1. Prototyping
- 14.5.2. Prototypes According to Their Scope
- 14.5.3. Prototypes According to Their Seasonality
- 14.5.4. The Fidelity of a Prototype
- 14.5.5. User Interface Prototypes
- 14.5.6. Evaluation of Prototypes

14.6. Requirements Analysis
- 14.6.1. Requirements Analysis
- 14.6.2. Requirements Analysis Best Practices
- 14.6.3. The Data Dictionary
- 14.6.4. Prioritization of Requirements

14.7. Documentation of Requirements
- 14.7.1. The Requirements Specification Document
- 14.7.2. Structure and Contents of an SRS
- 14.7.3. Natural Language Documentation
- 14.7.4. EARS: Easy Approach to Requirements Syntax
- 14.7.5. Non-Functional Requirements
- 14.7.6. Attributes and Templates in Table Form
- 14.7.7. Good Specifications Practices

14.8. Validation and Negotiation of Requirements
- 14.8.1. Validation of Requirements
- 14.8.2. Requirements Validation Techniques
- 14.8.3. Negotiation of Requirements

14.9. Modeling and Requirements Management
- 14.9.1. Requirements Modeling
- 14.9.2. The User's Perspective
- 14.9.3. The Data Perspective
- 14.9.4. The Functional or Flow-Oriented Perspective
- 14.9.5. The Behavioral Perspective
- 14.9.6. Volatility of Requirements
- 14.9.7. Requirements Management Process
- 14.9.8. Tools for Requirements Management
- 14.9.9. Best Practices in Requirements Management

14.10. Critical Systems and Formal Specification
- 14.10.1. Critical Systems
- 14.10.2. Risk-Driven Specification
- 14.10.3. Formal Specification

Module 15. Software Engineering Processes

15.1. Software Engineering Framework
- 15.1.1. Software Features
- 15.1.2. The Main Processes in Software Engineering
- 15.1.3. Software Development Process Models
- 15.1.4. Standard Reference Framework for the Software Development Process: ISO/IEC 12207 Regulations

15.2. Unified Software Development Process
- 15.2.1. The Unified Process
- 15.2.2. Dimensions of the Unified Process
- 15.2.3. Case Studies Driven Development Process
- 15.2.4. Fundamental Unified Process Workflows

15.3. Planning in the Context of Agile Software Development
- 15.3.1. Characteristics of Agile Software Development
- 15.3.2. Different Planning Time Horizons in Agile Development
- 15.3.3. Scrum Agile Development Framework and Planning Time Horizons
- 15.3.4. User Stories as a Planning and Estimating Unit
- 15.3.5. Common Techniques for Deriving an Estimate
- 15.3.6. Scales for Interpreting Estimates
- 15.3.7. Planning Poker
- 15.3.8. Common Scheduling Types: Delivery Scheduling and Iteration Scheduling

15.4. Distributed Software Design Styles and Service-Oriented Software Architectures
- 15.4.1. Communication Models in Distributed Software Systems
- 15.4.2. Middleware
- 15.4.3. Architecture Patterns for Distributed Systems
- 15.4.4. General Software Service Design Process
- 15.4.5. Design Aspects of Software Services
- 15.4.6. Composition of Services
- 15.4.7. Web Services Architecture
- 15.4.8. Infrastructure and SOA Components

15.5. Introduction to Model Driven Software Development
- 15.5.1. The Model Concept
- 15.5.2. Model-Driven Software Development
- 15.5.3. MDA Model-Driven Development Framework
- 15.5.4. Elements of a Transformation Model

15.6. Graphical User Interface Design
- 15.6.1. Principles of User Interface Design
- 15.6.2. Architectural Design Patterns for Interactive Systems: Model View Controller (MVC)
- 15.6.3. UX User Experience
- 15.6.4. User-Centered Design
- 15.6.5. Graphical User Interface Analysis and Design Process
- 15.6.6. Usability of User Interfaces
- 15.6.7. Accessibility in User Interfaces

15.7. Web Application Design
- 15.7.1. Characteristics of Web Applications
- 15.7.2. Web Application User Interface
- 15.7.3. Navigation Design
- 15.7.4. Basic Interaction Protocol for Web Applications
- 15.7.5. Architecture Styles for Web Applications

15.8. Software Testing Strategies and Techniques and Software Quality Factors
- 15.8.1. Testing Strategies
- 15.8.2. Test Case Designs
- 15.8.3. Value for Money
- 15.8.4. Quality Models
- 15.8.5. ISO/IEC 25000 Family of Standards (SQuaRE)
- 15.8.6. Product Quality Model (ISO 2501n)
- 15.8.7. Data Quality Models (ISO 2501n)
- 15.8.8. Software Quality Management

15.9. Introduction to Software Engineering Metrics
- 15.9.1. Basic Concepts: Measures, Metrics and Indicators
- 15.9.2. Types of Metrics in Software Engineering
- 15.9.3. The Measurement Process
- 15.9.4. ISO 25024. External and Quality Metrics in Use
- 15.9.5. Object-Oriented Metrics

15.10. Software Maintenance and Re-Engineering
- 15.10.1. Maintenance Process
- 15.10.2. Standard Maintenance Process Framework. ISO/EIEC 14764
- 15.10.3. Software Reengineering Process Model
- 15.10.4. Reverse Engineering

Module 16. Integration Systems

16.1. Introduction to Information Systems in the Enterprise
- 16.1.1. The Role of Information Systems
- 16.1.2. What is an Information System?
- 16.1.3. Dimensions of Information Systems
- 16.1.4. Business Processes and Information Systems
- 16.1.5. The IS/IT Department

16.2. Opportunities and Needs of Information Systems in the Enterprise
- 16.2.1. Organizations and Information Systems
- 16.2.2. Features of Organizations
- 16.2.3. Impact of Information Systems in the Enterprise
- 16.2.4. Information Systems to Achieve a Competitive Advantage
- 16.2.5. Use of Systems in the Administration and Management of the Enterprise

16.3. Basic Concepts of Information Systems and Technologies
- 16.3.1. Data, Information and Knowledge
- 16.3.2. Technology and Information Systems
- 16.3.3. Technology Components
- 16.3.4. Classification and Types of Information Systems
- 16.3.5. Service and Business Process Based Architectures
- 16.3.6. Forms of Systems Integration

16.4. Systems for the Integrated Enterprise Resource Planning
- 16.4.1. Business Needs
- 16.4.2. An Integrated Enterprise Resource Planning
- 16.4.3. Acquisition vs. Development
- 16.4.4. ERP Implementation
- 16.4.5. Implications for Management
- 16.4.6. Leading ERP Vendors

16.5. Supply Chain and Customer Relationship Management Information Systems
- 16.5.1. Definition of Supply Chain
- 16.5.2. Effective Supply Chain Management
- 16.5.3. The Role of Information Systems
- 16.5.4. Supply Chain Management Solutions
- 16.5.5. Customer Relationship Management
- 16.5.6. The Role of Information Systems
- 16.5.7. Implementation of a CRM System
- 16.5.8. Critical Success Factors in CRM Implementation
- 16.5.9. CRM, e-CRM and Other Trends

16.6. ICT Investment Decision-Making and Information Systems Planning
- 16.6.1. Criteria for ICT Investment Decisions
- 16.6.2. Linking the Project to the Management and Business Plan
- 16.6.3. Management Implications
- 16.6.4. Redesign of Business Processes
- 16.6.5. Management's Decision on Implementation Methodologies
- 16.6.6. Need for Information Systems Planning
- 16.6.7. Objectives, Participants and Moments
- 16.6.8. Structure and Development of the Systems Planning
- 16.6.9. Follow-Up and Updating

16.7. Security Considerations in the Use of ICTs
- 16.7.1. Risk Analysis
- 16.7.2. Security in Information Systems
- 16.7.3. Practical Advice

16.8. Feasibility of ICT Project Implementation and Financial Aspects in Information Systems Projects
- 16.8.1. Description and Objectives
- 16.8.2. EVS Participants
- 16.8.3. Techniques and Procedures
- 16.8.4. Cost Structure
- 16.8.5. Financial Projection
- 16.8.6. Budgets

16.9. Business Intelligence
- 16.9.1. What Is Business Intelligence?
- 16.9.2. BI Implementation Strategy
- 16.9.3. Present and Future in BI

16.10. ISO/IEC 12207
- 16.10.1. What is "ISO/IEC 12207"?
- 16.10.2. Analysis of Information Systems
- 16.10.3. Information System Design
- 16.10.4. Implementation and Acceptance of the Information System

Module 17. Software Reuse

17.1. General Overview of the Software Reuse
- 17.1.1. What Is Software Reuse?
- 17.1.2. Advantages and Disadvantages of Software Reuse
- 17.1.3. Main Techniques of Software Reuse

17.2. Introduction to Design Patterns
- 17.2.1. What is a Design Patterns?
- 17.2.2. Catalog of the Main Design Patterns
- 17.2.3. How to Use Patterns to Solve Design Problems
- 17.2.4. How to Select the Best Design Pattern

17.3. Creation Patterns
- 17.3.1. Creation Patterns
- 17.3.2. Abstract Factory Pattern
- 17.3.3. Example of Abstract Factory Pattern Implementation
- 17.3.4. Builder Pattern
- 17.3.5. Builder Implementation Example
- 17.3.6. Abstract Factory Pattern vs. Builder

17.4. Creation Patterns (II)
- 17.4.1. Factory Method Pattern
- 17.4.2. Factory Method vs. Abstract Factory
- 17.4.3. Singleton Pattern

17.5. Structural Patterns
- 17.5.1. Structural Patterns
- 17.5.2. Adapter Pattern
- 17.5.3. Bridge Pattern

17.6. Structural Patterns (II)
- 17.6.1. Composite Pattern
- 17.6.2. Decorator Pattern

17.7. Structural Patterns (III)
- 17.7.1. Facade Pattern
- 17.7.2. Proxy Pattern

17.8. Behavioral Patterns
- 17.8.1. Concept of Behavioral Patterns
- 17.8.2. Pattern of Behavior: Responsibility Chain
- 17.8.3. Behavior Pattern Order

17.9. Behavioral Patterns (II)
- 17.9.1. Interpreter Pattern
- 17.9.2. Iterator Pattern
- 17.9.3. Observer Pattern
- 17.9.4. Strategy Pattern

17.10. Frameworks
- 17.10.1. Framework Concept
- 17.10.2. Development Using Frameworks
- 17.10.3. Model View Controller Pattern
- 17.10.4. Frameworks for Graphical User Interface Design
- 17.10.5. Frameworks for Web Application Development
- 17.10.6. Frameworks for Managing Object Persistence in Databases

Module 18. Information Technology Services

18.1. Digital Transformation (I)
- 18.1.1. Business Innovation
- 18.1.2. Production Management
- 18.1.3. Financial Management

18.2. Digital Transformation (II)
- 18.2.1. Marketing
- 18.2.2. HR Management
- 18.2.3. The Integrated Information System

18.3. Case Study
- 18.3.1. Company Presentation
- 18.3.2. Methodologies to Analyze the Acquisition of IT
- 18.3.3. Determining the Costs, Benefits and Risks
- 18.3.4. Economic Evaluation of Investment

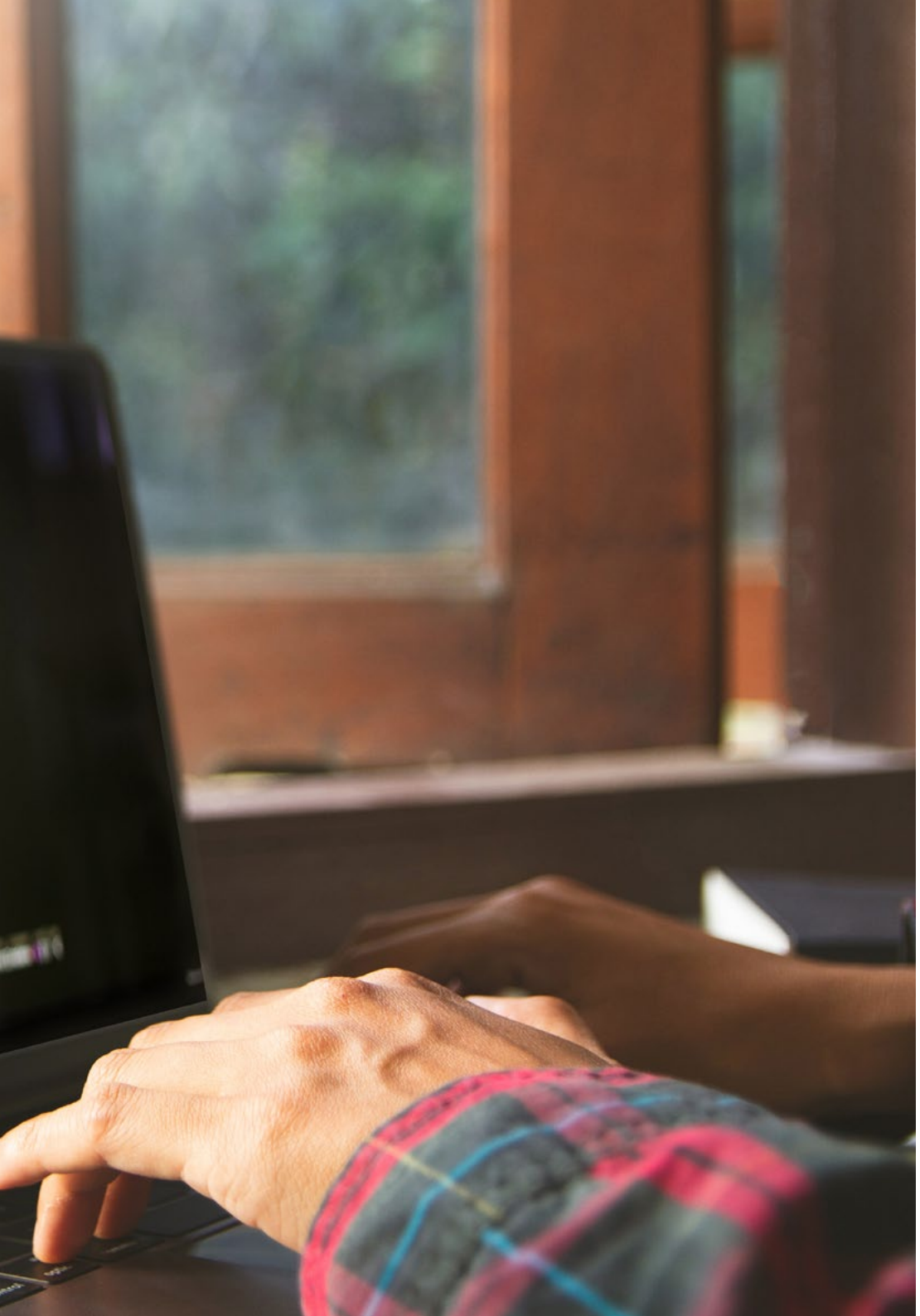
18.4. ICT Governance and Management
- 18.4.1. Definition of IT and Information Systems Governance
- 18.4.2. Difference Between IT Systems Governance and Management
- 18.4.3. Framework for IT Systems Governance and Management
- 18.4.4. Regulations and IT Systems Governance and Management

18.5. ICT Corporate Governance
- 18.5.1. What Is Good Corporate Governance?
- 18.5.2. ICT Governance Background
- 18.5.3. The ISO/IEC 38500:2008 Standard
- 18.5.4. Implementation of Good ICT Governance
- 18.5.5. ICT Governance and Best Practices
- 18.5.6. Corporate Governance. Summary and Trends

18.6. Control Objectives for Information and Related Technologies (COBIT)
- 18.6.1. Application Framework
- 18.6.2. Domain: Planning and Organization
- 18.6.3. Domain: Acquisition and Implementation
- 18.6.4. Domain: Delivery and Support
- 18.6.5. Domain: Supervision and Evaluation
- 18.6.6. Application of the COBIT Guide

18.7. The Information Technology Infrastructure Library (ITIL)
- 18.7.1. Introduction to ITIL
- 18.7.2. Service Strategies
- 18.7.3. Service Design
- 18.7.4. Transition Between Services
- 18.7.5. Service Operation
- 18.7.6. Improving the Service

18.8. The Service Management System
- 18.8.1. Basic Principles of UNE-ISO/IEC 20000-1
- 18.8.2. The Structure of the ISO/IEC 20000 Regulations
- 18.8.3. Service Management System (SMS) Requirements
- 18.8.4. Design and Transition of New or Modified Services
- 18.8.5. Service Provision Processes
- 18.8.6. Groups of Processes



18.9. The Software Asset Management System
- 18.9.1. Justification of Needs
- 18.9.2. Background
- 18.9.3. Presentation of the 19770 Regulation
- 18.9.4. Management Implementation

18.10. Business Continuity Management
- 18.10.1. Business Continuity Plan
- 18.10.2. Implementation of a BCP

Take advantage of your previous knowledge and apply for the validation of subjects already passed, optimizing your experience in this program"

04

Teaching Objectives

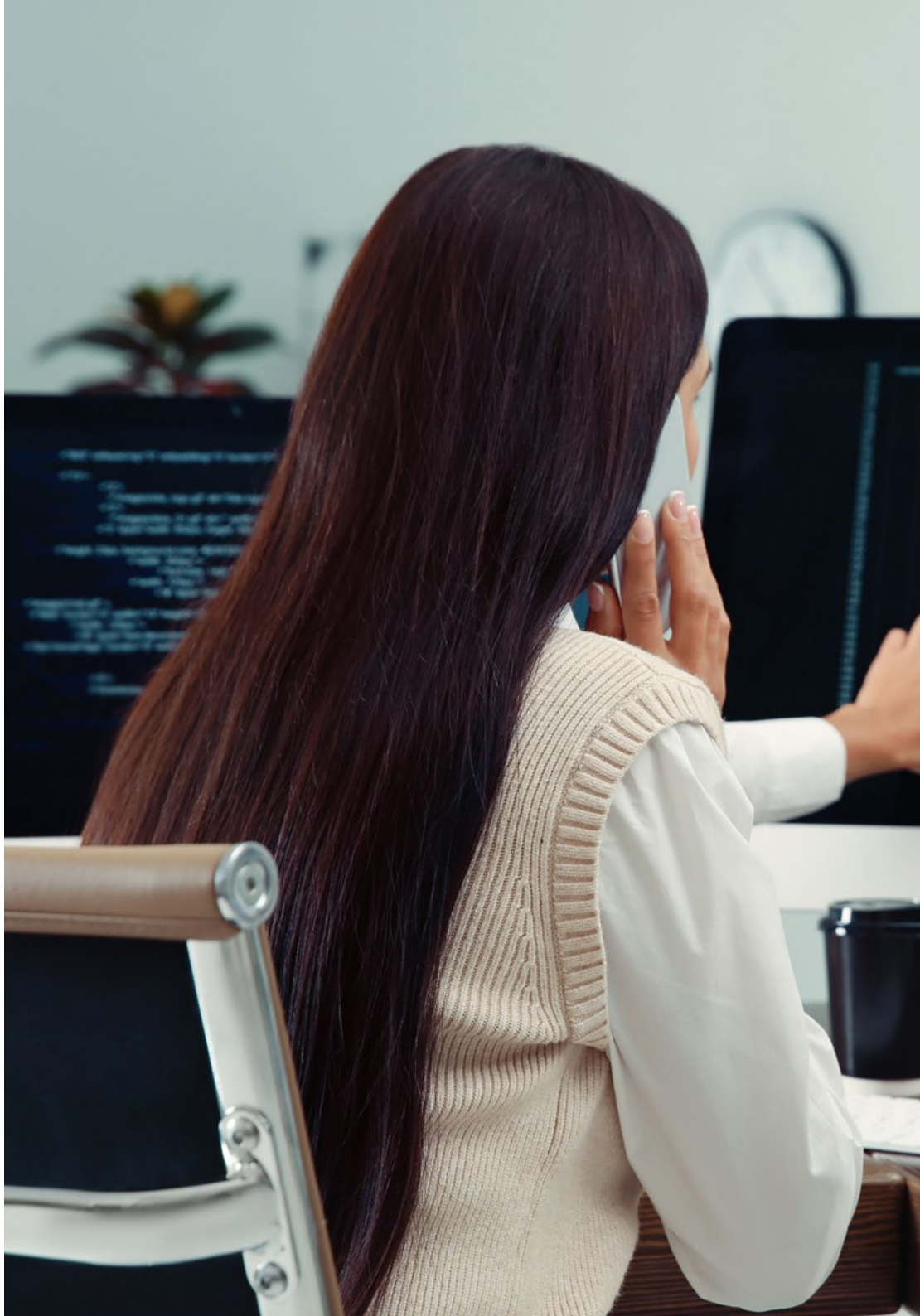
This program in Software Engineering has been designed to provide IT professionals with the advanced tools necessary to propel their careers into modern software development, adapted to the changing demands of the marketplace. Through specialized technical knowledge, students will significantly increase their opportunities for professional growth, opening doors to key roles in leading companies in the industry.

A program designed to become the ultimate boost towards a successful career in Software Engineering"

General Objectives

- Acquire new skills needed and demanded in terms of new technologies and latest developments in software
- Complement the acquired knowledge with skills in the field of computation and computer structure, including the mathematical, statistical and physical basis essential in Engineering
- Expand knowledge in Software Engineering and Computer Systems with the latest developments and most innovative methodology
- Tackle complex software projects and environments, knowing how to provide intelligent solutions to diverse problems

Boost your career with a specialization program designed to lead in the strategic direction of Software Engineering"



Specific Objectives

Module 1. Methodologies, Development and Quality in Software Engineering

- Train in the use of agile and traditional methodologies in software development
- Ensure the quality of the final product by applying standards and best practices in each phase of the software lifecycle

Module 2. Software Project Management

- Develop the skills necessary to plan, execute and manage software projects
- Apply project management methodologies and planning tools to ensure on-time and on-budget delivery

Module 3. Software Development Platforms

- Provide knowledge about the different software development platforms
- Address the use of development environments, frameworks and tools that optimize the creation of robust and scalable applications

Module 4. Web-Client Computing

- Train in the development of client-side web applications, using technologies such as HTML, CSS and JavaScript
- Create interactive interfaces optimized for user experience

Module 5. Web Server Computing

- Develop the necessary skills to manage and develop web server-side applications
- Handle technologies such as PHP, Node.js or Python, and managing databases and backend services

Module 6. Security Management
- Provide the knowledge and skills to manage security in software development environments
- Apply policies and strategies for data protection, access and security of the technological infrastructure

Module 7. Information Systems Security
- Provide an in-depth understanding of information systems security
- Analyze from network and server protection to the implementation of security protocols and standards in technological infrastructures

Module 8. Software Security
- Train in the development of secure software
- Identify, mitigate and prevent vulnerabilities in the code and application development processes

Module 9. Quality and Auditing of Information Systems
- Develop skills to carry out quality audits of information systems
- Ensure the effectiveness of security controls, compliance with regulations and optimization of technological processes

Module 10. Web Server Administration
- Train in the management and administration of web servers
- Delve into the maintenance of servers that host online applications and services

Module 11. Online Application Security
- Provide the necessary knowledge to secure online applications
- Address aspects such as authentication, data protection, access management and attack prevention in web environments

Module 12. Software Engineering
- Provide a comprehensive understanding of the principles and practices of software engineering
- Design, develop and maintain efficient, scalable and high quality software applications

Module 13. Advanced Software Engineering
- Delve into advanced software engineering techniques
- Delve into sophisticated development methodologies, design patterns, software architecture and process optimization for complex projects

Module 14. Requirements Engineering
- Develop skills in the identification, analysis and documentation of software requirements
- Ensure that the final products meet the needs and expectations of the users

Module 15. Software Engineering Processes
- Train in the fundamental processes of software engineering, from planning and design to implementation, testing and maintenance
- Improve the quality and efficiency of the software life cycle

Module 16. Integration Systems

- Integrate different software systems, databases and applications
- Create cohesive and efficient technology solutions to solve complex business problems

Module 17. Software Reuse

- Apply techniques and principles for the reuse of software components and modules in new projects
- Optimize development time and costs by creating reusable libraries or frameworks

Module 18. Information Technology Services

- Design, implement and manage information technology services
- Improve technology infrastructure to ensure business continuity and operational efficiency

05

Career Opportunities

Upon completion of this university program, professionals will have a solid understanding of the techniques, methodologies and tools necessary to address modern software development projects adapted to current market needs. Graduates will be able to design, implement and manage advanced technological solutions, ensuring quality, security and efficiency in all phases of development. In addition, they will be able to lead multidisciplinary teams, collaborate in agile environments and make strategic decisions based on international standards.

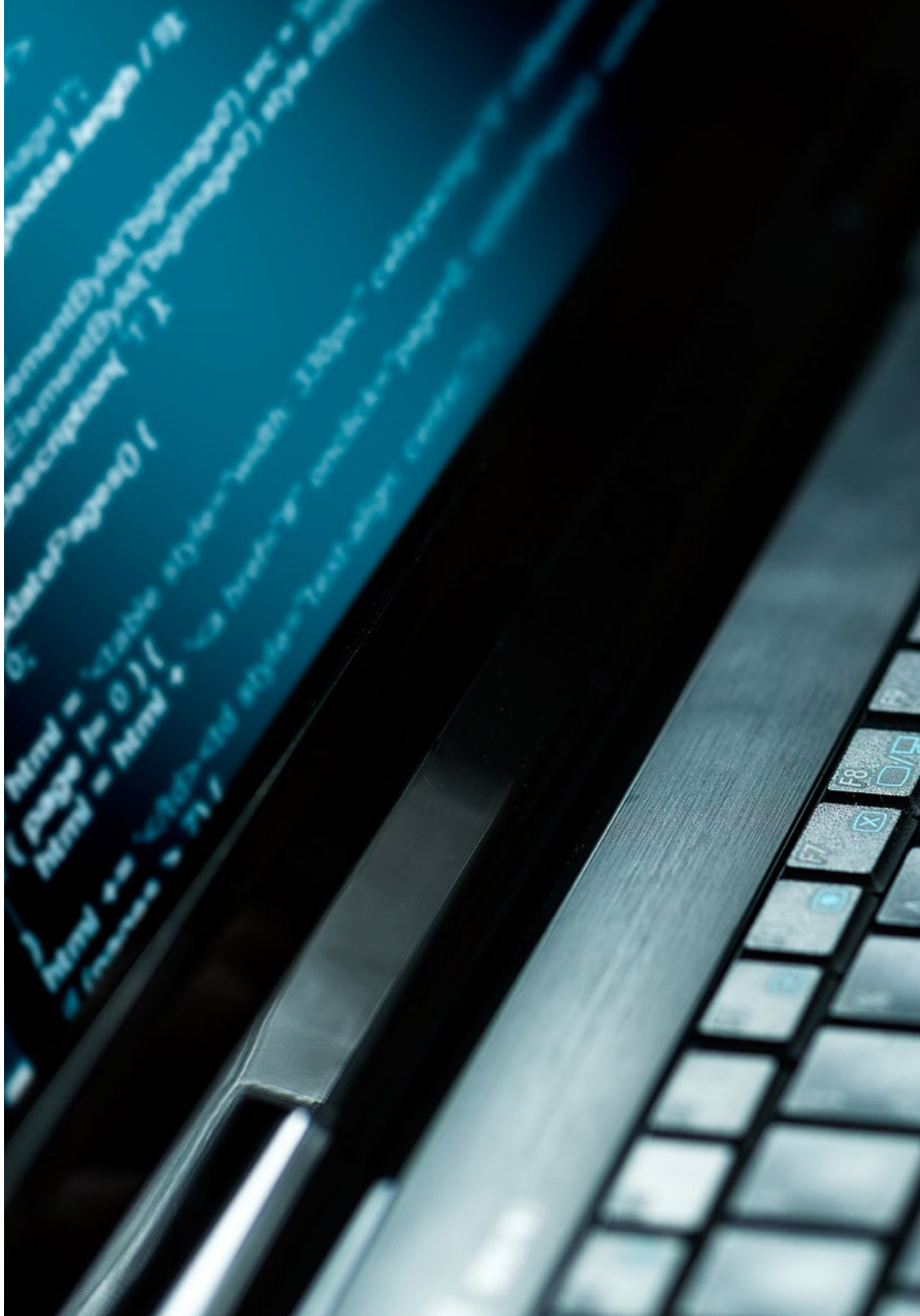
You will apply innovative approaches and effective solutions that optimize the development and management of software in complex technological projects"

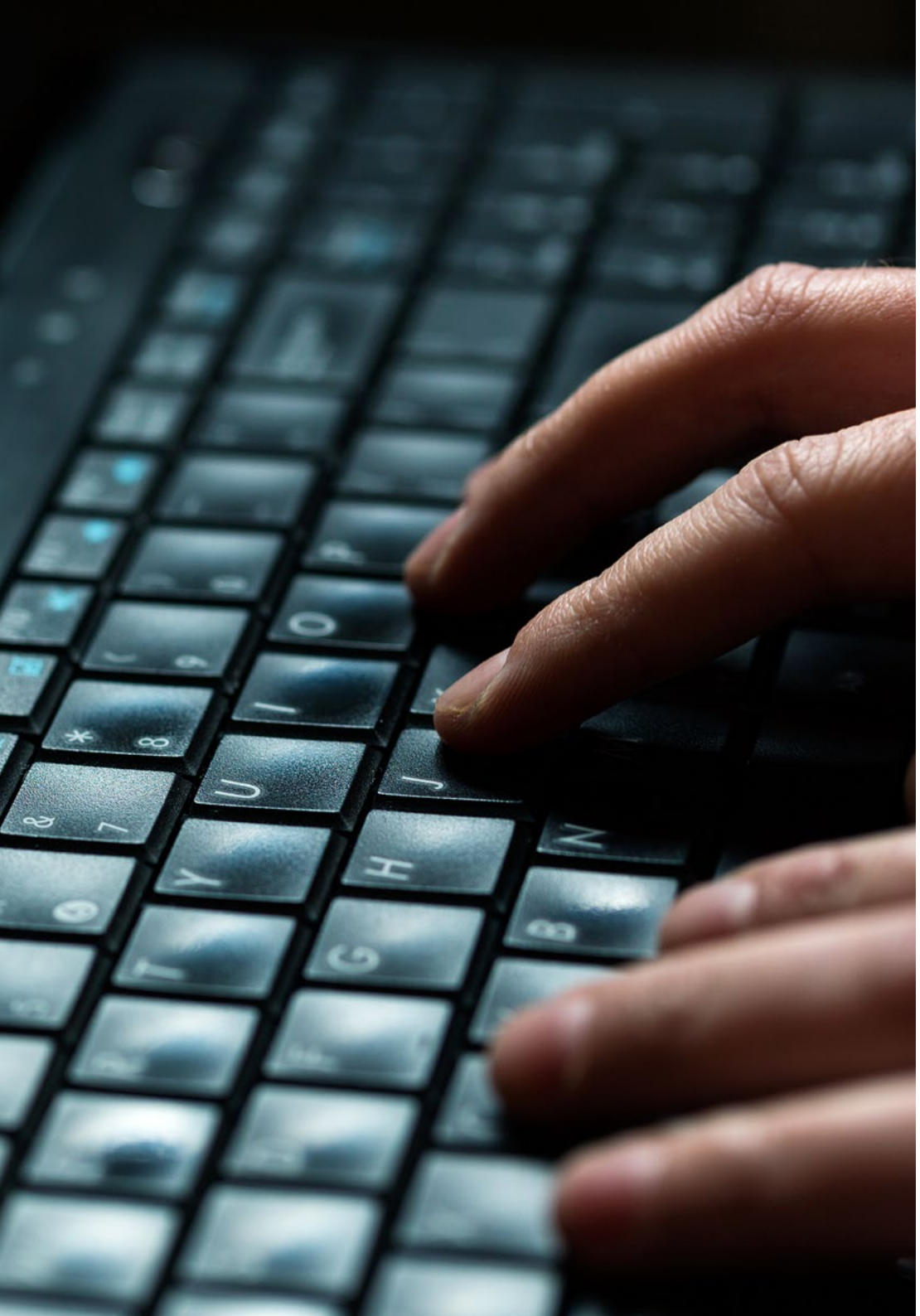
Graduate Profile

Graduates of the Software Engineering program are highly qualified to face the technological challenges of modern software development. Additionally, they have a deep knowledge of advanced methodologies, development tools and international standards, necessary to create innovative and effective solutions. Furthermore, you are prepared to design, implement and manage high-impact technological projects, collaborate with multidisciplinary teams and lead strategic initiatives in dynamic and competitive environments, ensuring the quality, security and scalability of the developed applications.

You will combine theoretical knowledge with practical skills in software design and advanced development methodologies.

- **Software Design and Development:** Ability to design and implement advanced applications, integrating design patterns, UML modeling and development platforms such as Java and .NET
- **Technological Project Management:** Ability to plan, supervise and lead technology projects using agile methodologies such as Scrum and applying international ISO/IEC standards
- **Security in Development:** In-depth understanding of best practices in IT security, risk management and data protection at all stages of the software lifecycle.
- **Implementation in Complex Environments:** Ability to develop mobile, web and cloud-based applications using technologies such as REST, SOA and MVC.
- **Problem Solving:** Application of critical thinking and innovative strategies to identify and solve technological challenges in software projects.
- **Digital Competence:** Proficiency in advanced technological tools and digital platforms for development, testing and management of systems





After completing the Advanced Master's Degree, you will be able to apply your knowledge and skills in the following positions:

1. **Software Project Manager:** In charge of planning and executing complex projects, ensuring that quality, time and budget objectives are met.
2. **Software Architect:** Specialist in the design and construction of complex and scalable systems.
3. **IT Security Engineer:** In charge of protecting systems, applications and data against vulnerabilities and attacks.
4. **Full-Stack Developer:** Professional capable of creating end-to-end solutions on both the client and server side.
5. **Technological Consultant:** Expert advisor in the implementation of advanced technologies and digital transformation strategies.
6. **Cloud Application Specialist:** Designer and implementer of SaaS, PaaS and IaaS based solutions.
7. **Systems Administrator:** Manager of servers, networks and technology platforms, optimizing their performance and security.
8. **Software Quality Analyst:** Evaluator of functionality, usability and performance of technological applications.
9. **Requirements Engineering Consultant:** Specialist in the analysis, documentation and validation of requirements in software projects.

Complete this program and gain access to key positions within the technology industry, standing out as a leader in Software Engineering"

06

Study Methodology

TECH is the world's first university to combine the **case study** methodology with **Relearning**, a 100% online learning system based on guided repetition.

This disruptive pedagogical strategy has been conceived to offer professionals the opportunity to update their knowledge and develop their skills in an intensive and rigorous way. A learning model that places students at the center of the educational process giving them the leading role, adapting to their needs and leaving aside more conventional methodologies.

TECH will prepare you to face new challenges in uncertain environments and achieve success in your career"

The student: the priority of all TECH programs

In TECH's study methodology, the student is the main protagonist.
The teaching tools of each program have been selected taking into account the demands of time, availability and academic rigor that, today, not only students demand but also the most competitive positions in the market.

With TECH's asynchronous educational model, it is students who choose the time they dedicate to study, how they decide to establish their routines, and all this from the comfort of the electronic device of their choice. The student will not have to participate in live classes, which in many cases they will not be able to attend. The learning activities will be done when it is convenient for them. They can always decide when and from where they want to study.

"At TECH you will NOT have live classes (which you might not be able to attend)"

The most comprehensive study plans at the international level

TECH is distinguished by offering the most complete academic itineraries on the university scene. This comprehensiveness is achieved through the creation of syllabi that not only cover the essential knowledge, but also the most recent innovations in each area.

By being constantly up to date, these programs allow students to keep up with market changes and acquire the skills most valued by employers. In this way, those who complete their studies at TECH receive a comprehensive education that provides them with a notable competitive advantage to further their careers.

And what's more, they will be able to do so from any device, pc, tablet or smartphone.

> *TECH's model is asynchronous, so it allows you to study with your pc, tablet or your smartphone wherever you want, whenever you want and for as long as you want"*

Case Studies and Case Method

The case method has been the learning system most used by the world's best business schools. Developed in 1912 so that law students would not only learn the law based on theoretical content, its function was also to present them with real complex situations. In this way, they could make informed decisions and value judgments about how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

With this teaching model, it is students themselves who build their professional competence through strategies such as Learning by Doing or Design Thinking, used by other renowned institutions such as Yale or Stanford.

This action-oriented method will be applied throughout the entire academic itinerary that the student undertakes with TECH. Students will be confronted with multiple real-life situations and will have to integrate knowledge, research, discuss and defend their ideas and decisions. All this with the premise of answering the question of how they would act when facing specific events of complexity in their daily work.

Relearning Methodology

At TECH, case studies are enhanced with the best 100% online teaching method: Relearning.

This method breaks with traditional teaching techniques to put the student at the center of the equation, providing the best content in different formats. In this way, it manages to review and reiterate the key concepts of each subject and learn to apply them in a real context.

In the same line, and according to multiple scientific researches, reiteration is the best way to learn. For this reason, TECH offers between 8 and 16 repetitions of each key concept within the same lesson, presented in a different way, with the objective of ensuring that the knowledge is completely consolidated during the study process.

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

A 100% online Virtual Campus with the best teaching resources

In order to apply its methodology effectively, TECH focuses on providing graduates with teaching materials in different formats: texts, interactive videos, illustrations and knowledge maps, among others. All of them are designed by qualified teachers who focus their work on combining real cases with the resolution of complex situations through simulation, the study of contexts applied to each professional career and learning based on repetition, through audios, presentations, animations, images, etc.

The latest scientific evidence in the field of Neuroscience points to the importance of taking into account the place and context where the content is accessed before starting a new learning process. Being able to adjust these variables in a personalized way helps people to remember and store knowledge in the hippocampus to retain it in the long term. This is a model called Neurocognitive context-dependent e-learning that is consciously applied in this university qualification.

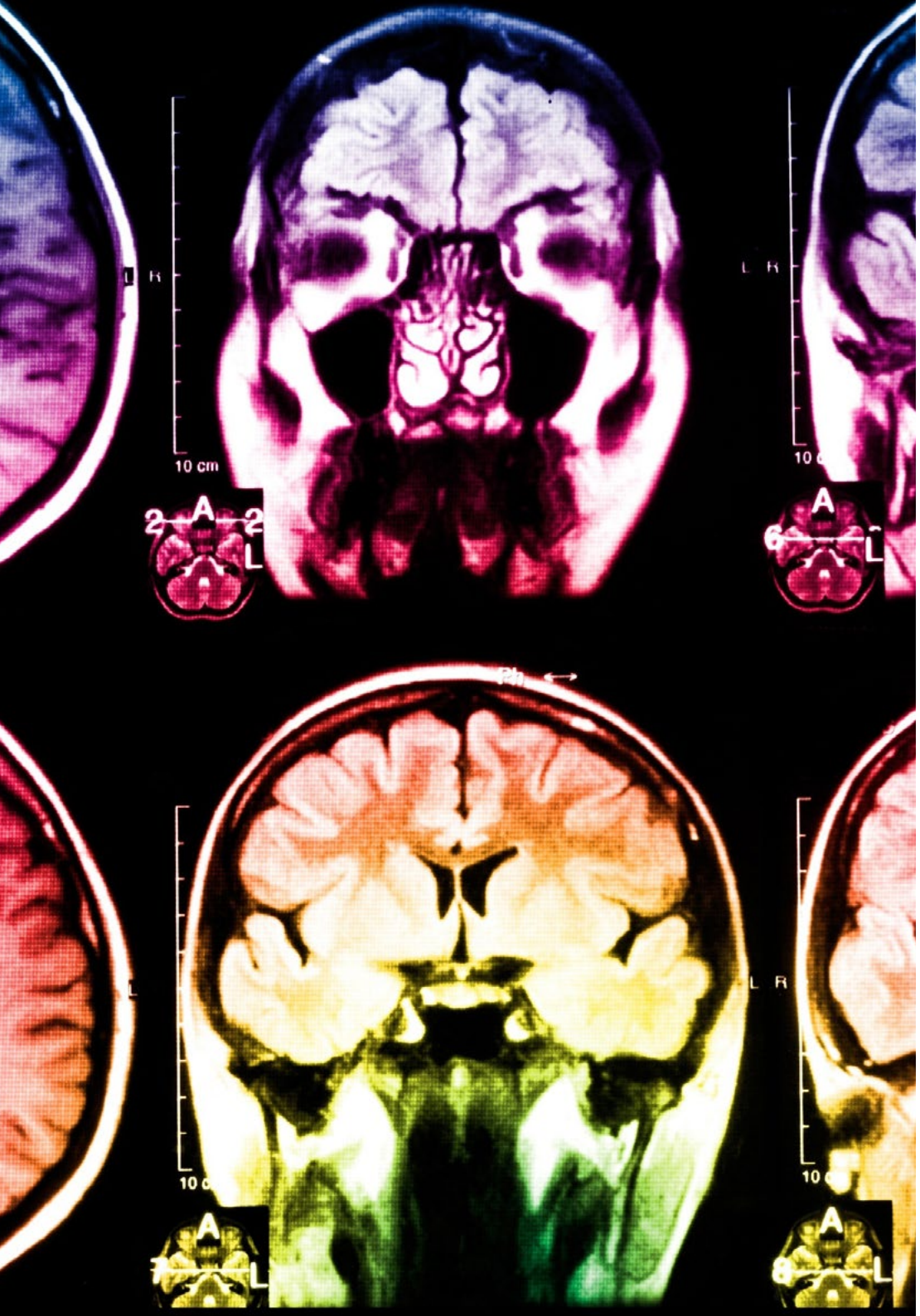
In order to facilitate tutor-student contact as much as possible, you will have a wide range of communication possibilities, both in real time and delayed (internal messaging, telephone answering service, email contact with the technical secretary, chat and videoconferences).

Likewise, this very complete Virtual Campus will allow TECH students to organize their study schedules according to their personal availability or work obligations. In this way, they will have global control of the academic content and teaching tools, based on their fast-paced professional update.

The online study mode of this program will allow you to organize your time and learning pace, adapting it to your schedule"

The effectiveness of the method is justified by four fundamental achievements:

1. Students who follow this method not only achieve the assimilation of concepts, but also a development of their mental capacity, through exercises that assess real situations and the application of knowledge.
2. Learning is solidly translated into practical skills that allow the student to better integrate into the real world.
3. Ideas and concepts are understood more efficiently, given that the example situations are based on real-life.
4. Students like to feel that the effort they put into their studies is worthwhile. This then translates into a greater interest in learning and more time dedicated to working on the course.



The university methodology top-rated by its students

The results of this innovative teaching model can be seen in the overall satisfaction levels of TECH graduates.

The students' assessment of the teaching quality, the quality of the materials, the structure of the program and its objectives is excellent. Not surprisingly, the institution became the top-rated university by its students according to the global score index, obtaining a 4.9 out of 5.

Access the study contents from any device with an Internet connection (computer, tablet, smartphone) thanks to the fact that TECH is at the forefront of technology and teaching.

You will be able to learn with the advantages that come with having access to simulated learning environments and the learning by observation approach, that is, Learning from an expert.

As such, the best educational materials, thoroughly prepared, will be available in this program:

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.

This content is then adapted in an audiovisual format that will create our way of working online, with the latest techniques that allow us to offer you high quality in all of the material that we provide you with.

Practicing Skills and Abilities

You will carry out activities to develop specific competencies and skills in each thematic field. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop within the framework of the globalization we live in.

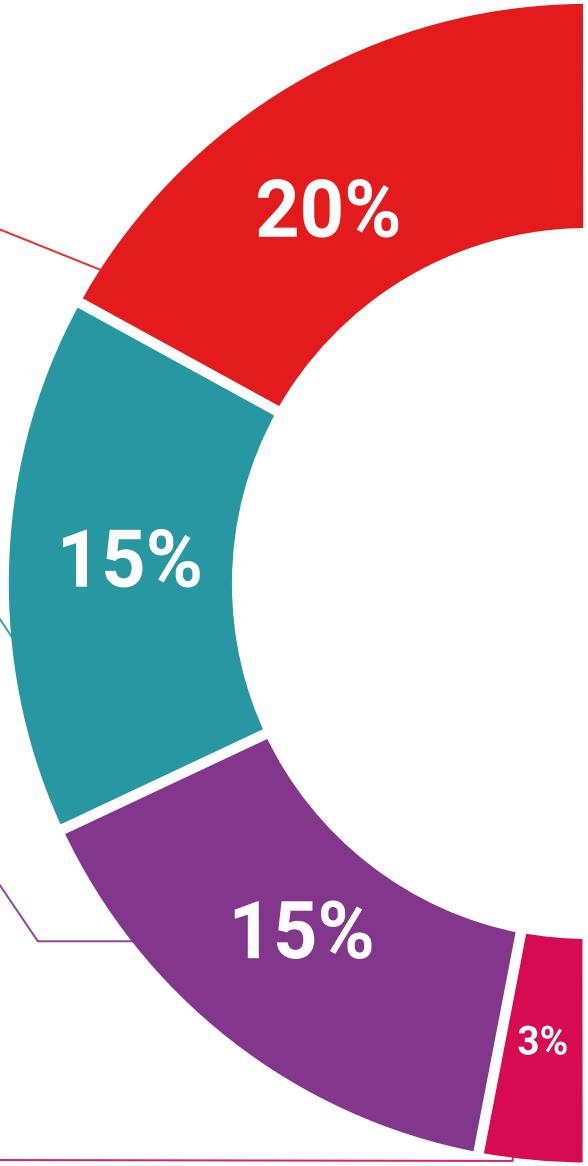
Interactive Summaries

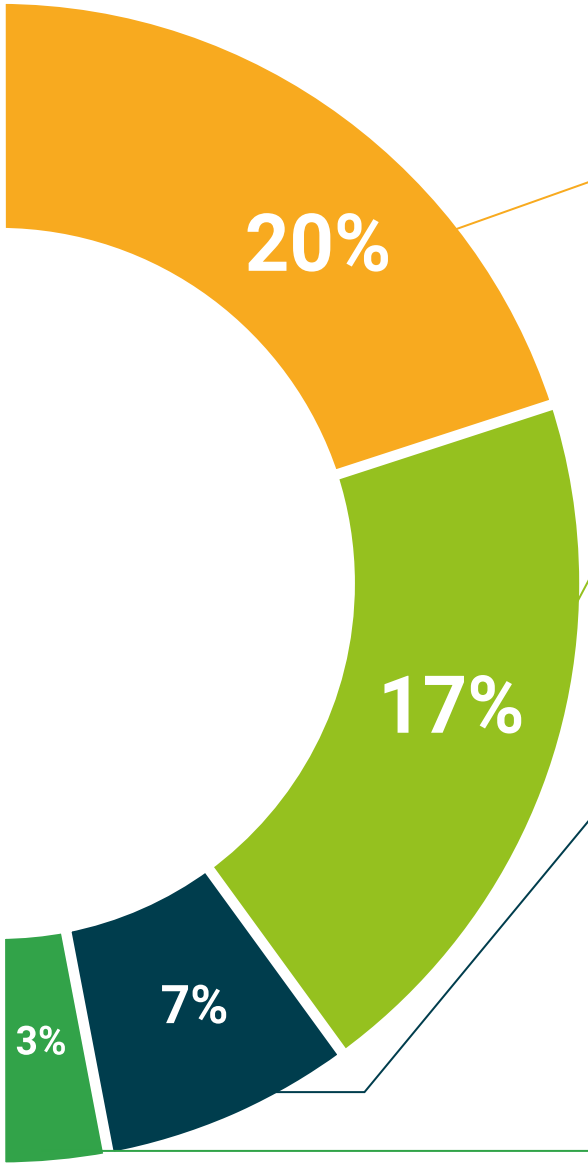
We present the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

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

Additional Reading

Recent articles, consensus documents, international guides... In our virtual library you will have access to everything you need to complete your education.





Case Studies

Students will complete a selection of the best case studies in the field. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Testing & Retesting

We periodically assess and re-assess your knowledge throughout the program. We do this on 3 of the 4 levels of Miller's Pyramid.

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 for future difficult decisions.

Quick Action Guides

TECH offers the most relevant contents of the course in the form of worksheets or quick action guides. A synthetic, practical and effective way to help students progress in their learning.

07

Teaching Staff

The professors of this program in Software Engineering stand out for their deep technical knowledge and practical experience in the software industry. In fact, they have led significant projects in leading companies in the sector, which allows them to share not only advanced theories, but also practical applications and real cases that will enrich the preparation of graduates. In addition, their commitment to innovation and research drives them to keep up to date with the latest trends and development methodologies.

Learn from teachers who are experts in Computer Science and Software Engineering, with experience and advanced knowledge"

International Guest Director

Darren Pulsipher is a highly experienced **software architect**, an innovator with an outstanding international track record in **oftware and firmware development**. **In fact, he possesses highly developed** communication, project management **and** business **skills, which have enabled him to lead major global initiatives.**

He has also held senior positions of great responsibility throughout his career, such as Chief Solution Architect for the Public Sector **at** Intel Corporation, **where he has promoted** modern business, processes and technologies **for customers, partners and users in the** public sector. **In addition, he founded** Yoly Inc. **where he has also served as** CEO, **working to develop a** social network aggregation and diagnostic tool **based on** Software as a Service (SaaS), **using *Big Data* and** Web 2.0 technologies.

Additionally, he has served in other companies, as Senior Director of Engineering, **at** Dell Technologies, **where he led the** Big Data **in the** Cloud Business Unit, **leading teams in the** United States **and** China **for the management of large projects and the restructuring of business divisions for their successful integration. He has also worked as *Chief Information Officer* at** XanGo, **where he managed projects such as** Help Desk ***support***, production support **and** solution development.

Among the many specialties in which he is an expert, *Edge to Cloud* technology, cybersecurity, Generative Artificial Intelligence, software development, networking technology, cloud-native development **and the** container ecosystem **stand out. Knowledge he has shared through the** "Embracing Digital Transformation" podcast **and** weekly newsletter, **which he produced and hosted, helping organizations successfully navigate** digital transformation **by leveraging** people, processes **and** technology.

Mr. Pulsipher, Darren

- Chief Solution Architect for Public Sector at Intel, California, United States
- Presenter and Producer of "Embracing Digital Transformation", California
- Founder and CEO at Yoly Inc., Arkansas
- Senior Director of Engineering at Dell Technologies, Arkansas
- Chief Information Technology Officer, XanGo, Utah
- Senior Architect at Cadence Design Systems, California
- Senior Project Process Manager at Lucent Technologies, California
- Software Engineer at Cemax-Icon, California
- Software Engineer at ISG Technologies, Canada
- MBA in Technology Management from the University of Phoenix
- B.S. in Computer Science and Electrical Engineering from Brigham Young University

Thanks to TECH, you will be able to learn with the best professionals in the world"

08

Certificate

The Advanced Master's Degree in Software Engineering guarantees students, in addition to the most rigorous and up-to-date education, access to an Advanced Master's Degree diploma 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 an **Advanced Master's Degree diploma in 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: **Advanced Master's Degree in Software Engineering**

Modality: **online**

Duration: **2 years**

Accreditation: **120 ECTS**

tech global university

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

Advanced Master's Degree in Software Engineering

This is a private qualification of 3,600 hours of duration equivalent to 120 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

Advanced Master's Degree in Software Engineering

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Methodologies, Development and Quality in Software Engineering	6	CO
1º	Software Project Management	6	CO
1º	Software Development Platforms	6	CO
1º	Web-Client Computing	6	CO
1º	Web Server Computing	6	CO
1º	Security Management	6	CO
1º	Information Systems Security	7	CO
1º	Software Security	7	CO
1º	Quality and Auditing of Information Systems	7	CO
2º	Web Server Administration	7	CO
2º	Online Application Security	7	CO
2º	Software Engineering	7	CO
2º	Advanced Software Engineering	7	CO
2º	Requirements Engineering	7	CO
2º	Software Engineering Processes	7	CO
2º	Integration Systems	7	CO
2º	Software Reuse	7	CO
2º	Information Technology Services	7	CO

Dr. Pedro Navarro Illana
Dean

tech global university

*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

Advanced Master's Degree
Software Engineering

- Modality: **online**
- Duration: **2 years**
- Certificate: **TECH Global University**
- Accreditation: **120 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

Advanced Master's Degree
Software Engineering

tech global university