

Hybrid Professional Master's Degree Software and Computer Systems Engineering



Hybrid Professional Master's Degree Software and Computer Systems Engineering

Modality: Hybrid (Online + Internship)

Duration: 12 months.

Certificate: TECH Technological University

Teaching Hours: 1,620 hours.

Website: www.techtitute.com/us/information-technology/master/software-computer-systems-engineering

Index

01

Introduction

p. 4

02

Why Study this Hybrid
Professional Master's
Degree?

p. 8

03

Objectives

p. 12

04

Competencies

p. 18

05

Educational Plan

p. 22

06

Practices

p. 36

07

Where Can I Do the
Internship?

p. 42

08

Methodology

p. 46

09

Certificate

p. 54

01

Introduction

Software engineering has undergone an enormous evolution in the last two decades. The development of increasingly complex and efficient computer systems has favored the creation of applications and programs adapted to the needs of society, using different languages and mechanisms, and making it possible to increase the security and stability of their structures exponentially. However, for many professionals in this sector, thoroughly mastering the latest strategies and keeping up with updates was a difficult task to accomplish until TECH decided to launch this degree. This is a multidisciplinary theoretical and practical program, which will allow you not only to broaden your knowledge in a specialized way, but also to perfect your professional skills through active work in a leading company in the IT industry.



“

A multidisciplinary program that will mark a before and after in your career through the improvement of your professional skills as a specialist in Software Engineering"

It could be said that the objective of software engineering is the creation of computer programs that meet the needs of society as a whole and of companies. However, it goes beyond this, since it allows, through database management, to save production costs, increase system security and raise the quality of the services offered. Thanks to the advance of technology and the development of programming, it is now possible to model applications with secure and stable structures through different languages and codes, which makes this branch of computer science a complex, but exciting and highly demanded profession in the digital society.

For this reason, and with the aim that the graduate finds in a single degree the content that allows him not only to catch up, but to specialize in this area in a multidisciplinary way, TECH has developed this blended Master in Software Engineering and Computer Systems. It is a program that combines 1,500 hours of the best theoretical content with 3 weeks of practical training in a leading software development company.

The degree includes a 100% online syllabus that delves into methodologies, development and quality in software engineering, project management, web computing, security management and server administration. All this, through information selected by a team of specialized engineers who have also developed additional material in different formats so that the graduate can get the most out of the content. of the content.

You will then have the opportunity to do practical training in a specialized software development center, where you will actively participate in the projects being developed at that time. During this period, a tutor will guide you through the experience, ensuring the completion of a plan of activities that will allow you not only to acquire a broad and specialized knowledge of the current profession, but also to improve your skills and abilities exponentially and based on the requirements of the current demand of the labor market.

This **Master's Degree in Software Engineering and Computer Systems** 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 computer engineering professionals with extensive experience in software project management
- ♦ Its graphic, schematic and practical contents with which they are conceived, gather CURRENT and assistance information on those disciplines that are essential for professional practice
- ♦ Assessment and detection of vulnerabilities in online applications, knowledge of security policies and standards, management of web services technologies, mitigation and containment of web services incidents, etc
- ♦ 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
- ♦ In addition, you will be able to do a 3-week internship in a prestigious software development company



You will be able to study all the theoretical content 100% online and with a fully customized schedule, so you can combine it with any other activity without any problem"

“

Web security has never been more important. With this Semipresential Master you will learn in detail the best tools to guarantee inviolable and stable applications”

This Master blended learning's program, which has a professionalizing nature and Communication Management blended learning modality, is aimed at updating computer engineering specialized in aesthetic units, and who require a high level of qualification. The contents are based on the latest scientific evidence, and oriented in an educational way to integrate theoretical knowledge into practice, and the theoretical-practical elements will facilitate knowledge update and decision-making in patient management.

Thanks to their multimedia content developed with the latest educational technology, they will allow the physiotherapy professional to obtain situated and contextual learning, i.e. a simulated environment that will provide immersive learning programmed to train 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, students will be assisted by an innovative interactive video system created by renowned and experienced experts.

You will have access to practical cases of risk assessment in IT auditing, so that you will know the most effective tests and the appropriate support tools for each type of project.

A multidisciplinary and intensive option to learn everything necessary to handle the HTTP protocol on multiple servers.



02

Why Study this Hybrid Professional Master's Degree?

Software Engineering has countless job opportunities: from the programming of navigation systems in ships and airplanes, to the design of software packages for word processing, among the thousands of options it offers. For this reason, and because of the technical and practical knowledge required by its activity, TECH has developed the ideal program to specialize in this area in a multidisciplinary way.

Through this blended Master, the graduate will be able to work on the knowledge of the main strategies for the management of computer systems, acquiring, in addition, a thorough mastery of its tools with a 3-week stay in a reference company on the international scene.

“

A multidisciplinary experience of the highest level with which you will undoubtedly reach your professional zenith in the management of the security of the main IT systems"

1. Updating from the Latest Technology Available

TECH is a pioneer in the entire online academic landscape for the use of state-of-the-art teaching tools, in order to make your experiences the best and most dynamic. For this reason, and in order to continue along the same line of innovation, it requires its practice centers to be up to date with the technology and strategies of their sector, so that graduates can update their practice in a new and intensive way based on the most significant advances in their profession.

2. Gaining In-depth Knowledge from the Experience of Top Specialists

The professionals who will accompany the graduate during the 12 months of theoretical and practical training have a wide and extensive experience in the management of information systems. Therefore, you will be able to contextualize the information included in the curriculum through the anecdotes of these experts, which will motivate you to understand the problems to be solved, as well as to make decisions about the strategies to follow in certain cases.

3. Combining the Best Theory with State-of-the-Art Practice

This blended master's degree ideally combines theoretical and practical information, offering the graduate the opportunity to access an unparalleled multidisciplinary academic experience. In this way, they are able to expand their knowledge in a specialized and updated way, with the motivation and security of culminating the course with a stay in which they will be able to apply it and verify its effectiveness.



4. Expanding the Boundaries of Knowledge

The syllabus of this program has been designed following the guidelines of the teaching team, which, being formed by professionals versed in the sector who are also active, know what aspects the graduate must master to succeed in the software engineering market. For this reason, TECH guarantees access to a limitless academic experience that will undoubtedly allow students to exceed even their most ambitious expectations in just 12 months.

5. Expanding the Boundaries of Knowledge

The practical stay of this blended Master's program can be carried out in companies of international scope. Thanks to this, the graduate can obtain a critical, exhaustive and diverse vision of the reality of Computer Engineering in different parts of the world, contributing to an integral and multidisciplinary specialization perfectly applicable in any country.

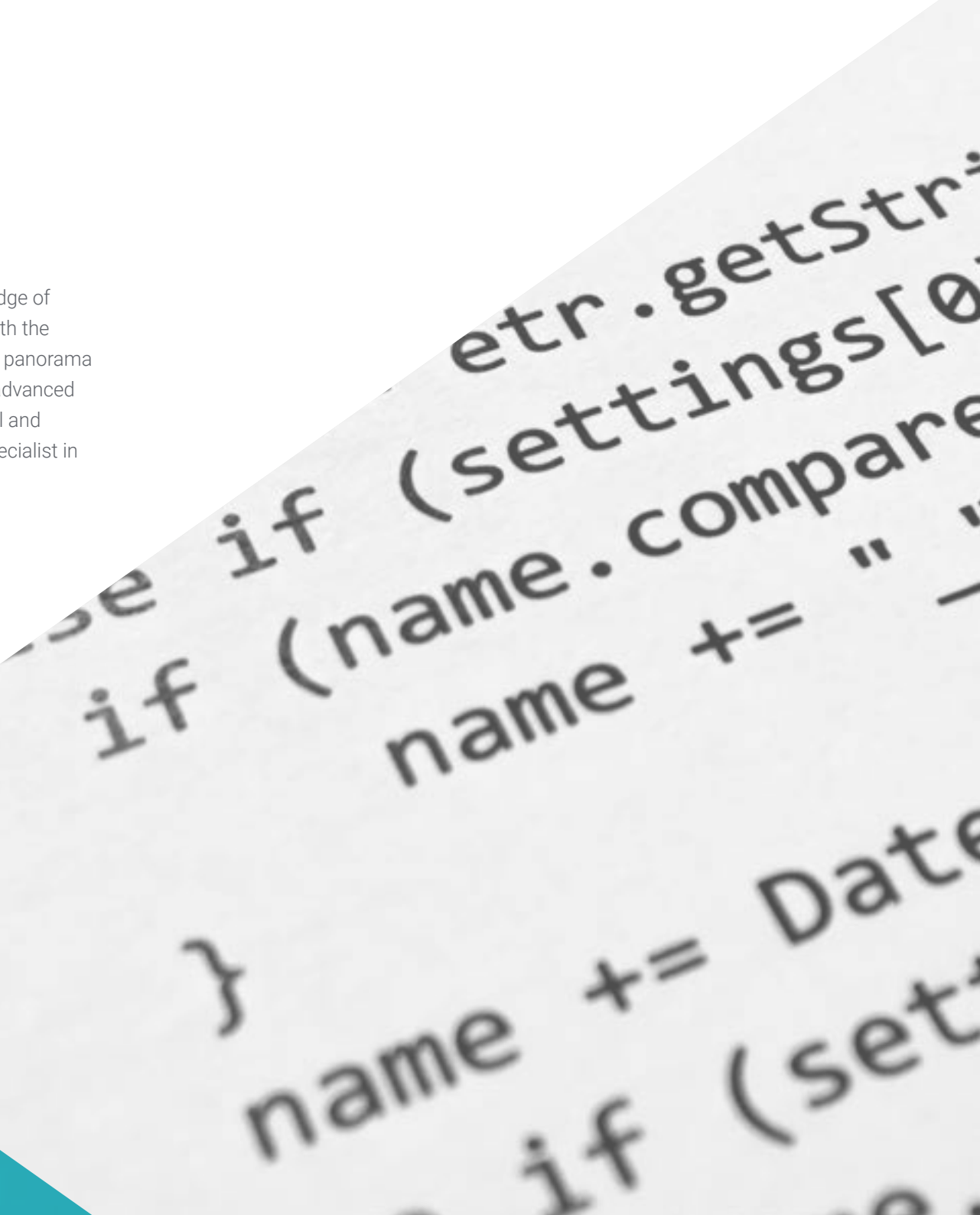


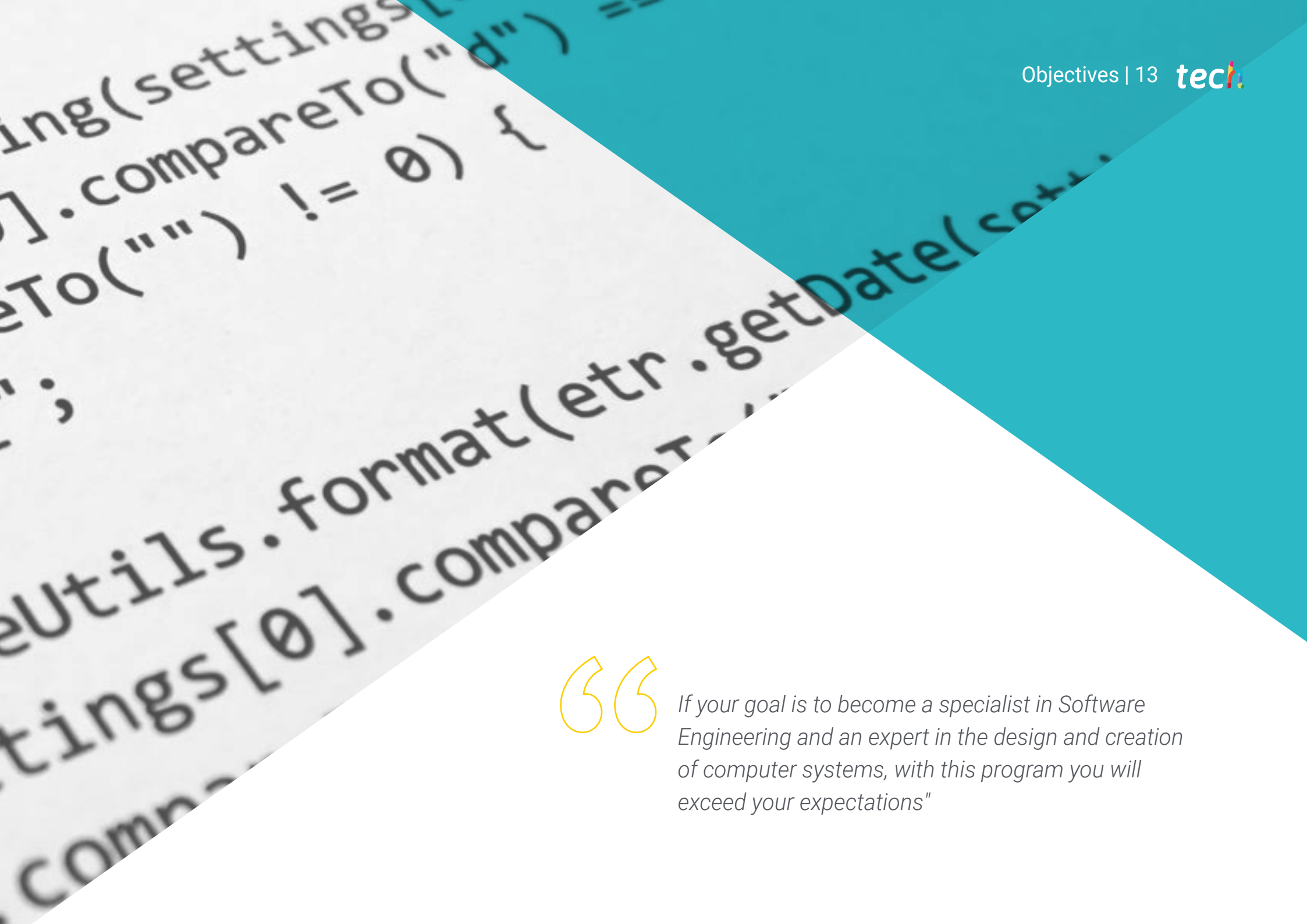
You will have full practical immersion at the center of your choice"

03

Objectives

The handling of computer systems requires a broad and exhaustive knowledge of them. For this reason, TECH has developed this Blended Master's Degree with the objective that the graduate has access to the best program in the academic panorama that includes not only a complete and dynamic syllabus, but also the most advanced pedagogical tools that allow him/her to develop his/her skills in a theoretical and practical way. In this way, and in less than 1 year, you will have become a specialist in Software Engineering.





“

If your goal is to become a specialist in Software Engineering and an expert in the design and creation of computer systems, with this program you will exceed your expectations"



General Objective

- This Practical Training has been developed so that the graduate not only acquires new knowledge related to the latest developments in the Software Engineering and Computer Systems sector, but also to improve his or her skills in the use of new technologies and software. In addition, you will have the necessary knowledge to process the data generated in your activity in a professional, optimal and effective manner



You will delve into advanced PHP language and syntax, using XAMPP to test the generation of HTML content"





Specific Objectives

Module 1. Methodologies, Development and Quality in Software Engineering

- ◆ Know the basics of Software Engineering, as well as the set of rules or principles of ethics and professional responsibility during and after development
- ◆ Understand the software development process, under the different programming models and the object-oriented programming paradigm
- ◆ Understand the different types of application modeling and design patterns in the Unified Modeling Language (UML)
- ◆ Acquire the knowledge for the correct application of agile methodologies in Software development such as Scrum, among others
- ◆ Know the *Lean* development methodology to identify the activities that do not add value to the process, in order to obtain a higher quality software.

Module 2. Software Project Management

- ◆ Know the fundamental concepts of project management and the project management life cycle
- ◆ Understand the different stages of project management such as initiation, planning, *stakeholder* management and scoping
- ◆ Learning schedule development for time management, budget development and risk response
- ◆ 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

Module 3. Software Development Platforms

- ◆ Understand the different software development platforms
- ◆ Acquire the necessary knowledge for the development of applications and graphical interfaces in Java and .NET languages
- ◆ Know the techniques required for the debugging and testing of the developments made
- ◆ Learn Android mobile application development environments and debugging and publishing processes
- ◆ Understand cloud-based application development and determine the correct procedures for its implementation
- ◆ Master the basic concepts, services and tools of the Google *Clouds* platform

Module 4. Web-Client Computing

- ◆ Understand the process of creating web content through HTML markup language
- ◆ Understand the procedures and techniques to improve the appearance of a document written in HTML
- ◆ Know the evolution of the JavaScript language
- ◆ Acquire the necessary knowledge for the development of web client-side applications
- ◆ Develop applications with complex structures, by using the different procedures, functions and objects that integrate JavaScript
- ◆ Learn how to use the DOM programming interface for HTML and XML documents to modify their structure, style and content
- ◆ Understand the use of event-based flow and *listeners*, as well as the use of modern toolkit and alignment systems
- ◆ Know the concept of web usability, its advantages, principles, methods and techniques to make a web site usable by the user
- ◆ Establish knowledge of web accessibility, its importance in current digital platforms, methodologies, norms, standards and determine compliance

Module 5. Web Server Computing

- ♦ Understand the basic, intermediate and advanced concepts of the PHP language for the implementation of server-side applications
- ♦ Acquire the necessary knowledge for data modeling, relationships, keys and normalizations
- ♦ Understand the construction of the logical data model, the specification of tables, columns, keys and dependencies, as well as the knowledge necessary for the physical handling of data, file types, access modes and file organization
- ♦ files, access modes and file organization
- ♦ Learn how to integrate applications developed in PHP with MariaDB and MySQL databases
- ♦ Master the process of customer interactions, using : forms, *Cookies* and session management
- ♦ Understand the Model View Controller View (MVC) Software architecture that separates an application's data, user interface, and control logic into three distinct components
- ♦ Acquire the skills for the use of web services using XML, SOA and REST

Module 6. Safety Management

- ♦ Know the information security process, its implications on confidentiality, integrity, availability and economic costs
- ♦ Learn the use of good security practices in the management of information technology services
- ♦ Acquire the knowledge for the correct certification of security processes
- ♦ Understand authentication mechanisms and methods for access control, as well as the access audit process
- ♦ Understand security management programs, risk management and security policy design
- ♦ Learn about business continuity plans, their phases and maintenance process
- ♦ Know the procedures for the correct protection of the company through DMZ networks, the use of intrusion detection systems and other methodologies

Module 7. Software Security

- ♦ Understand Software security issues, vulnerabilities and how they are classified
- ♦ Know the design principles, methodologies and standards in software security
- ♦ Understand the application of security in the different phases of the software life cycle
- ♦ Acquire the necessary knowledge for secure software coding and validation techniques
- ♦ Understand the methodologies and processes to guarantee security during the development and delivery of cloud services
- ♦ Understand the basics of cryptology and the different encryption techniques currently available

Module 8. Web Server Administration

- ♦ Know the concept, operation, architecture, resources and contents of a web server
- ♦ Understand the functioning, structure and HTTP protocol handling
- ♦ Understand the concept of distributed multi-server architectures
- ♦ Master the functioning of an application server and another proxy
- ♦ Analyze the different web servers that are trending in today's market
- ♦ Understand the process of usage statistics and load balancing on web servers
- ♦ Acquire the necessary knowledge for the installation, administration, configuration and security of the *Microsoft Internet Information Services* (IIS) web server as well as the free Apache web server

Module 9. Security Audit

- ♦ Acquire the knowledge required for the correct execution of the audit process and internal computer control
- ♦ Understand the processes to carry out for the security audit in systems and networks
- ♦ Understand the different support tools, methodologies and subsequent analysis during internet and mobile device security auditing
- ♦ Learn the properties and influencing factors that condition business risks and determine the correct implementation of appropriate risk management
- ♦ Know the risk mitigation measures, as well as , the methodologies for the implementation of an Information Security Management System and the norms and standards to be used
- ♦ Understand the procedures for conducting the security audit, its traceability and presentation of results

```
36
37
38   this.bindInput();
39   this.bindButtons();
40 }
41
42 Searchbar.prototype = {
43   bindInput: function() {
44     var input = this.input;
45     var hist = this.searchHistory;
46     var self = this;
47
48     input.on("keydown", function(e) {
49       //escape
50       if (e.keyCode == 27) {
51         // this is a new line
52         self.deactivate(true);
53         return; // this is a modified line
54       }
55       if (e.keyCode == 13) {
56         e.stopImmediatePropagation();
57         e.preventDefault();
58         self.search();
59         self.deactivate();
60         return;
61       }
62       //up/down
63       if (e.keyCode == 38 || e.keyCode == 40) {
64         e.preventDefault();
65         e.stopImmediatePropagation();
66         if (e.keyCode == 38) { // up
67           // show previous search query
68           if (hist.currentIndex == hist.history.length - 1) {
69             hist.currentIndex = 0;
70             hist.currentQuery = hist.history[0];
71           }
72           // skip previous search if we're already there
73           if (hist.currentQuery == hist.history[hist.currentIndex]) {
74             hist.currentIndex--;
75           }
76         }
77       }
78     });
79   }
80 }
```

Module 10. Online Applications Security

- ◆ Acquire the knowledge required to evaluate and detect the vulnerabilities of online applications
- ◆ Understand the security policies and standards to be applied to online applications
- ◆ Know the procedures to use during the development of web applications and their subsequent evaluation through analysis and security tests
- ◆ Learn the security measures for the deployment and production of web applications
- ◆ Understand the concepts, functions and technologies to be applied in the security of web services, as well as security tests and protective measures
- ◆ Assimilate the procedures for ethical hacking, malware analysis and forensics
- ◆ Know the mitigation and containment measures for incidents on web services
- ◆ Acquire the knowledge to implement best practice techniques for the development and implementation of online applications as well as the most common errors

04 Skills

With the completion of this Semipresential Master's Degree, the graduate will acquire a series of professional competencies that will make him/her stand out in the Software Engineering environment. Not only will you have a thorough knowledge of the management and creation of IT projects, but you will also be able to perfect your skills and aptitudes through practice, solving real complex problems and developing strategies for computing, programming and evaluating applications, which you can then include in your curriculum.



“

During the course of this degree you will develop specialized knowledge of the security policies and standards to be applied to online applications"



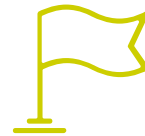
General Skills

- Design, manage and implement Software engineering and information systems projects.
- Understand the different types of application modeling and design patterns in the Unified Modeling Language (UML)



The course of this program will give you the keys to master software development to check the quality of the different programs and identify any errors they may present in their programmatic structure"





Specific Skills

- ◆ Understand how quality management works in projects, including planning, assurance, control, statistical concepts and available tools
- ◆ Use the necessary knowledge for the development of applications and graphical interfaces in Java and .NET languages
- ◆ Understand the procedures and techniques to improve the appearance of a document written in HTML
- ◆ Master the process of customer interactions, using forms, Cookies and session management
- ◆ Understand authentication mechanisms and methods for access control, as well as the access audit process
- ◆ Understand the application of security in the different phases of the software life cycle
- ◆ Know the concept, operation, architecture, resources and contents of a web server
- ◆ Understand the different support tools, methodologies and subsequent analysis during internet and mobile device security auditing
- ◆ Understand the security policies and standards to be applied to online applications

05

Educational Plan

For the development of the content of this program, TECH has taken into account two fundamental aspects: on the one hand, the requirements and specifications of current Computer Engineering, and on the other hand, the recommendation of a team of experts in the sector. All the information gathered has been adapted to the *Relearning* methodology, in whose application this center is a pioneer. Thanks to the use of this pedagogical strategy, together with the exhaustiveness of its syllabus and the quality and variety of the additional material that the graduate will find in the Virtual Campus, this degree will mark a before and after in his or her professional career.



“

A study plan designed to offer you the most advanced and updated knowledge in Hepatology in only 12 months"

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. Object 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. The *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. Define 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. Manage 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. Education
 - 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. Install and Start 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 Pseudoclasses
 - 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 *Maths* 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 Persons with Low Vision
 - 4.10.4.3. Aids for People 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 the and Neurological Impaired
 - 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

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. Safety Management

- 6.1. Information Security
 - 6.1.1. Introduction
 - 6.1.2. Information Security Involves Confidentiality, Integrity and Availability
 - 6.1.3. Safety is an Economic Issue
 - 6.1.4. Safety 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. Safety 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 Defence (*Firewalls, WAFs, WAFs, IPS,*
 - 6.7.3. Endpoint Defence (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. Software Security

- 7.1. Problems of the Software Security
 - 7.1.1. Introduction to the Problem of Software Safety
 - 7.1.2. Vulnerabilities and their Classification
 - 7.1.3. Secure Software Properties
 - 7.1.4. References
- 7.2. Software Safety Design Principles
 - 7.2.1. Introduction
 - 7.2.2. Software Safety Design Principles
 - 7.2.3. Types of S-SDLC
 - 7.2.4. Software Safety in S-SDLC Phases
 - 7.2.5. Methodologies and Standards
 - 7.2.6. References
- 7.3. Software Lifecycle Safety in the Requirements and Design Phases
 - 7.3.1. Introduction
 - 7.3.2. Attack Modeling
 - 7.3.3. Cases of Abuse
 - 7.3.4. Safety Requirements Engineering
 - 7.3.5. Risk Analysis Architectural
 - 7.3.6. Design Patterns
 - 7.3.7. References
- 7.4. Software Lifecycle Safety in the Coding, Testing and Operation Phases
 - 7.4.1. Introduction
 - 7.4.2. Risk-Based Safety Testing
 - 7.4.3. Code Review
 - 7.4.4. Penetration Test
 - 7.4.5. Security Operations
 - 7.4.6. External Review
 - 7.4.7. References
- 7.5. Secure Coding Applications I
 - 7.5.1. Introduction
 - 7.5.2. Secure Coding Practices
 - 7.5.3. Manipulation and Validation of Inputs
 - 7.5.4. Memory Overflow
 - 7.5.5. References
- 7.6. Secure Coding Applications II
 - 7.6.1. Introduction
 - 7.6.2. *Integers Overflows, Truncation Errors and Problems with Type Conversions between Integers*
 - 7.6.3. Errors and Exceptions

- 7.6.4. Privacy and Confidentiality
- 7.6.5. Privileged Programs
- 7.6.6. References
- 7.7. Development and Cloud Security
 - 7.7.1. Safety in Development; Methodology and Practice
 - 7.7.2. PaaS, IaaS, CaaS and SaaS Models
 - 7.7.3. Security in the Cloud and for Cloud Services
- 7.8. Encryption
 - 7.8.1. Fundamentals of Cryptology
 - 7.8.2. Symmetric and Asymmetric Encryption
 - 7.8.3. Encryption at Rest and in Transit
- 7.9. Security Automation and Orchestration (SOAR)
 - 7.9.1. Complexity of Manual Processing; Need to Automate Tasks
 - 7.9.2. Products and Services
 - 7.9.3. SOAR Architecture
- 7.10. Telework Safety
 - 7.10.1. Need and Scenarios
 - 7.10.2. Products and Services
 - 7.10.3. Telework Safety

Module 8. Web Server Administration

- 8.1. Introduction to Web Servers
 - 8.1.1. What is a Web Server?
 - 8.1.2. Architecture and Operation of a Web Server
 - 8.1.3. Resources and Contents on a Web Server
 - 8.1.4. Application Servers
 - 8.1.5. Proxy Servers
 - 8.1.6. Main Web Servers on the Market
 - 8.1.7. Web Server Usage Statistics
 - 8.1.8. Web Server Security
 - 8.1.9. Load Balancing on Web Servers
 - 8.1.10. References
- 8.2. HTTP Protocol Handling
 - 8.2.1. Operation and Structure
 - 8.2.2. Request Methods
 - 8.2.3. Status Codes
 - 8.2.4. Headers
 - 8.2.5. Content Coding Code Pages
 - 8.2.6. Performing HTTP Requests on the Internet using a Proxy, Livehttpheaders or Similar Method, Analyzing the Protocol Used.
- 8.3. Description of Distributed Multi-Server Architectures
 - 8.3.1. 3-Layer Model
 - 8.3.2. Fault Tolerance
 - 8.3.3. Load Sharing
 - 8.3.4. Session State Stores
 - 8.3.5. Cache Stores
- 8.4. Internet Information Services (IIS)
 - 8.4.1. What is IIS?
 - 8.4.2. History and Evolution of IIS
 - 8.4.3. Main Advantages and Features of IIS7 and Later Versions
 - 8.4.4. IIS7 Architecture and Later Versions
- 8.5. IIS Installation, Administration and Configuration
 - 8.5.1. Preamble
 - 8.5.2. Internet Information Services (IIS) Installation
 - 8.5.3. IIS Administration Tools
 - 8.5.4. Web Site Creation, Configuration and Administration
 - 8.5.5. Installation and Management of IIS Extensions
- 8.6. Advanced Security in IIS
 - 8.6.1. Preamble
 - 8.6.2. Authentication, Authorization, and Access Control in IIS
 - 8.6.3. Configuring a Secure Website on IIS with SSL
 - 8.6.4. Security Policies Implemented in IIS 8.x
- 8.7. Introduction to Apache
 - 8.7.1. What is Apache?
 - 8.7.2. Main Advantages of Apache
 - 8.7.3. Main Features of Apache
 - 8.7.4. Architecture

- 8.8. Apache Installation and Configuration
 - 8.8.1. Initial Installation of Apache
 - 8.8.2. Apache Configuration
- 8.9. Installation and Configuration of the Different Apache Modules
 - 8.9.1. Apache Module Installation
 - 8.9.2. Types of Modules
 - 8.9.3. Secure Apache Configuration
- 8.10. Advanced Security
 - 8.10.1. Authentication, Authorization and Access Control
 - 8.10.2. Authentication Methods
 - 8.10.3. Secure Apache Configuration with SSL

Module 9. Security Audit

- 9.1. Introduction to Information Systems in the Company
 - 9.1.1. Introduction to Information Systems in the Company and the Role of IT Auditing
 - 9.1.2. Definitions of "IT Audit" and "IT Internal Control"
 - 9.1.3. Functions and Objectives of IT Auditing
 - 9.1.4. Differences between Internal Control and IT Auditing
- 9.2. Internal Controls of Information Systems
 - 9.2.1. Functional Flowchart of a Data Processing Center
 - 9.2.2. Classification of Information Systems Controls
 - 9.2.3. The Golden Rule
- 9.3. The Process and Phases of the Information Systems Audit
 - 9.3.1. Risk Assessment and Other IT Auditing Methodologies
 - 9.3.2. Execution of an Information Systems Audit. Phases of the Audit
 - 9.3.3. Fundamental Skills of the Auditor of an IT System
- 9.4. Technical Audit of Security in Systems and Networks
 - 9.4.1. Technical Security Audits. Intrusion Test. Previous Concepts
 - 9.4.2. Security Audits in Systems. Support Tools
 - 9.4.3. Security Audits in Networks. Support Tools
- 9.5. Technical Audit of Security on the Internet and in Mobile Devices
 - 9.5.1. Internet Security Audit. Support Tools
 - 9.5.2. Mobile Devices Security Audit. Support Tools
 - 9.5.3. Annex 1. Structure of an Executive Report and Technical Report
 - 9.5.4. Annex 2. Tools Inventory
 - 9.5.5. Annex 3. Methods
- 9.6. Information Security Management System
 - 9.6.1. Security of IS: Properties and Influential Factors
 - 9.6.2. Business Risks and Risk Management: Implementing Controls
 - 9.6.3. Information Security Management System (ISMS): Concept and Critical Success Factors
 - 9.6.4. ISMS-PDCA Model
 - 9.6.5. ISMS ISO-IEC 27001: Organizational Context
 - 9.6.6. Annex 4. Context of the Organization
 - 9.6.7. Annex 5. Leadership.
 - 9.6.8. Annex 6. Education
 - 9.6.9. Annex 7. Support
 - 9.6.10. Annex 8. Operation
 - 9.6.11. Annex 9. Performance Evaluation
 - 9.6.12. Annex 10. Improvement
 - 9.6.13. Annex to ISO 27001/ISO-IEC 27002: Objectives and Controls
 - 9.6.14. ISMS Audit
- 9.7. Carrying Out the Audit
 - 9.7.1. Procedures
 - 9.7.2. Techniques
- 9.8. Traceability
 - 9.8.1. Methods
 - 9.8.2. Analysis
- 9.9. Copyright
 - 9.9.1. Techniques
 - 9.9.2. Results
- 9.10. Reports and Presenting Proof
 - 9.10.1. Types of Reports
 - 9.10.2. Analysis of Data
 - 9.10.3. Presenting Proof

Module 10. Online Application Security

- 10.1. Vulnerabilities and Security Issues in Online Applications
 - 10.1.1. Introduction to Online Application Security
 - 10.1.2. Security Vulnerabilities in the Design of Web Applications
 - 10.1.3. Security Vulnerabilities in the Implementation of Web Applications
 - 10.1.4. Security Vulnerabilities in the Deployment of Web Applications
 - 10.1.5. Official Lists of Security Vulnerabilities
- 10.2. Policies and Standards for Online Application Security
 - 10.2.1. Pillars for the Security of Online Applications
 - 10.2.2. Security Policy
 - 10.2.3. Information Security Management System
 - 10.2.4. Secure Software Development Life Cycle
 - 10.2.5. Standards for Application Security
- 10.3. Security in the Design of Web Applications
 - 10.3.1. Introduction to Web Application Security
 - 10.3.2. Security in the Design of Web Applications
- 10.4. Testing the Online Safety and Security of Web Applications
 - 10.4.1. Web Application Security Testing and Analysis
 - 10.4.2. Web Application Deployment and Production Security
- 10.5. Web Services Security
 - 10.5.1. Introduction to Web Services Security
 - 10.5.2. Web Services Security Functions and Technologies
- 10.6. Testing the Online Safety and Security of Web Services
 - 10.6.1. Evaluation of Web Services Security
 - 10.6.2. Online Protection. *Firewalls* and *XML Gateways*
- 10.7. Ethical *Hacking, Malware and Forensics*
 - 10.7.1. Ethical Hacking
 - 10.7.2. *Malware* Analysis
 - 10.7.3. Forensic Analysis



- 10.8. Incident Resolution on Web Services
 - 10.8.1. Monitoring
 - 10.8.2. Performance Measurement Tools
 - 10.8.3. Containment Measures
 - 10.8.4. Root Cause Analysis
 - 10.8.5. Proactive Problem Management
- 10.9. Best Practices to ensure Application Security
 - 10.9.1. Handbook of Best Practices in the Development of Online Applications
 - 10.9.2. Handbook of Good Practices in the Implementation of Online Applications
- 10.10. Common Errors that Undermine Application Security
 - 10.10.1. Common Errors in Development
 - 10.10.2. Common Errors in Hosting
 - 10.10.3. Common Production Errors

“*After completing this program you will master all aspects of planning, managing and directing a software project*”

06

Clinical Internship

Culminating the theoretical period with the possibility of putting into practice all that has been learned is TECH's way of showing its commitment to the professional growth of the graduate. For this reason, you will have 120 hours of stay in a leading software development company, where you will be able to actively participate in the projects being developed, acquiring a critical, strategic, but above all, real vision of the reality of the computer industry.



“

You will learn in detail the latest advances in web server programming and computing through logical and physical data modeling”

TECH is aware of the importance of having a practical period that allows graduates to develop their knowledge and perfect their professional skills. Therefore, it offers you the possibility of being part of a prestigious company in the field of computer engineering with international recognition and years of experience with the work of computer systems. This experience is developed over 120 hours, distributed in 3 weeks: from Monday to Friday with 8 consecutive hours with a specialist in the area.

In this training proposal, completely practical in nature, the activities are aimed at developing and perfecting the skills necessary for the provision of design and structure of Software, and are oriented towards specific training for the exercise of the activity high professional performance.

It is, therefore, a unique opportunity to catch up with the latest developments in the profession working in a leading company, with the most modern digital tools and the latest computer technology. In addition, the stay will allow the professional to complete a minimum number of practical activities, a valuable aspect in any recruitment process.

The 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 fellow trainees who facilitate teamwork and multidisciplinary integration as transversal competencies for the practice of La Software Engineering and Computer Systems (learning to be and learning to relate).

The practical procedures described below will be the basis of the practice of the training, and their performance will be subject to the availability of patients, their diagnostic and therapeutic indication, for each of them, during the Company.

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



A demonstrable experience to include in your resume that will open the doors of many companies in the IT sector"

Module	Practical Activity
Domain of the methodologies for the development and quality in Software Engineering and project management	Develop object-oriented or model-driven software as required for the job
	Follow agile methodologies such as SCRUM to support software development.
	Measure and test the software created based on quality processes.
	Plan the software project, from its idea and development phases to the tasks to be performed to be carried out
	Manage the development of the project schedule, time and planning.
	Establish an appropriate communication plan among all members of the development team. of the development team
Software development platforms and computing methods in the web client	Develop applications and graphical interface in Java, .NET, Android Studio or the appropriate program for the type of work to be done.
	Working in cloud application development environments such as Google Cloud Platform such as Google Cloud Platform
	Proficiency in HTML, CSS or Javascript design work
	Enhance web usability by following specific conceptual design guidelines and principles.
	Designing to overcome barriers and create more accessible sites
Management security management in the online application and in the software	Follow a comprehensive security process in the creation of software.
	Implement the principles of secure application coding in daily work.
	Automating security tasks through SOAR architecture
	Manage access control in the software project, as well as programs, processes and information security policies.
	Build a security architecture, involving processes for prevention, detection, reaction and recovery from potential threats.

Module	Practical Activity
Effective management administration of Web Servers	Handle the HTTP protocol in the implementation and administration of a web server.
	<i>Internet Information Services (IIS) Managing the Systemic Coaching</i>
	Manage Apache as the main tool in the administration of a web server.
Security in Software Engineering	Implement rigorous internal controls, with an appropriate organizational chart and classification, to ensure the security of information systems.
	Determine Technical Audit of Security on the Systemic Coaching Social Networks and in Mobile Devices
	Perform a security audit following the procedures and techniques advisable in software development environments.
	Detect possible vulnerabilities or security problems in the design of web applications. web application design
	Implement Testing Online Safety and Security of Web Applications
Develop its own manual of good practices to be followed to ensure the security of the applications of the applications	



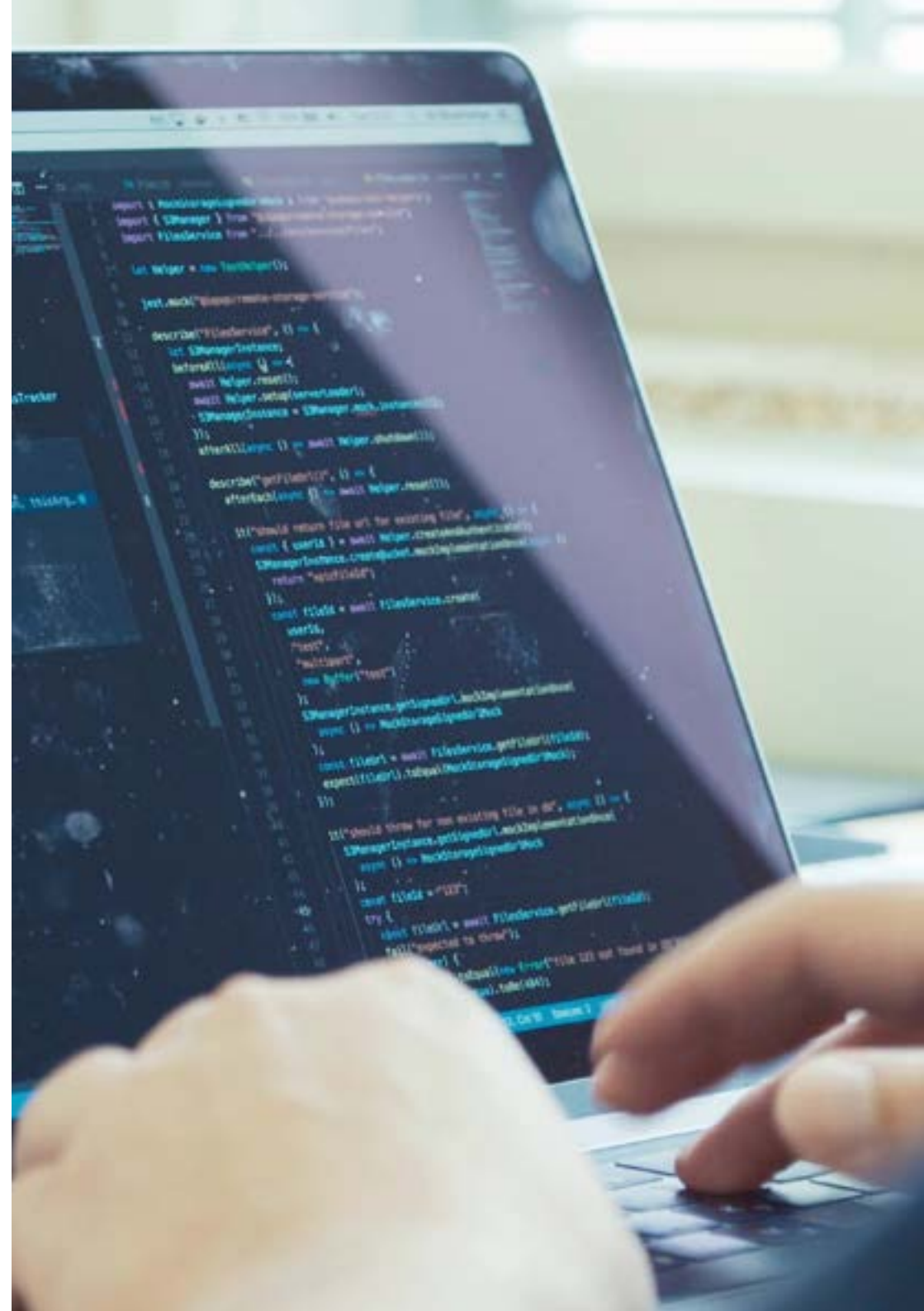
You will be able to work personally with the most advanced and sophisticated computer tools"

Civil Liability Insurance

This institution's main concern is to guarantee the safety of the trainees 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 entity commits to purchasing a civil liability insurance policy to cover any eventuality that may arise during the course of the internship at the 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 program agreement shall be 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?

Each year TECH selects a number of companies to offer as internships to complement their theoretical degrees. However, not any entity exceeds the criteria of exigency and quality imposed by this center, since the objective of this period is that the graduate obtains a beneficial and enriching experience for his or her future work. In addition, the trajectory and prestige that define them will also be a definitive characteristic in the final decision.

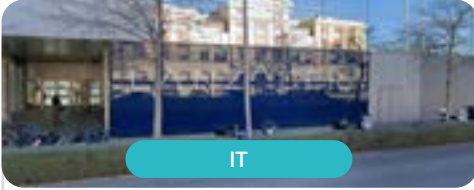




You will be able to work side by side with professionals with years of experience in software development, as well as learn from their successes and mistakes to establish your own strategy"



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



IT

GOOSE & HOPPER

Country	City
Spain	Valencia

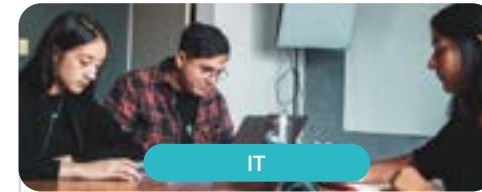
Address: La Marina de Valencia, Muelle de la Aduana S/N Edificio Lanzadera 46024

Advertising, design, technology and creative agency and creativity

Related internship programs:

- Digital Communication and Reputation Management
- Organic 3D Modeling





IT

Goose & Hopper

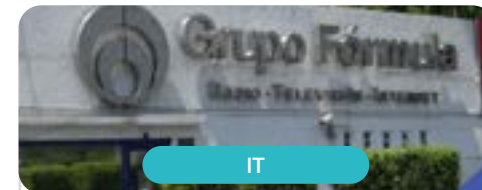
Country	City
Mexico	Michoacán de Ocampo

Address: Avenida Solidaridad Col. Nueva Chapultepec Morelia, Michoacan

Advertising, design, technology and creative agency and creativity

Related internship programs:

- Digital Communication and Reputation Management
- Organic 3D Modeling



IT

Grupo Fórmula

Country	City
Mexico	Mexico City

Address: Cda. San Isidro 44, Reforma Soc, Miguel Hidalgo, 11650 Ciudad de México, CDMX

Leading company in multimedia communication and content generation

Related internship programs:

- Graphic Design
- People Management

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.

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



“

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

This **Internship Program in Software Engineering and Computer Systems** contains the most complete and up-to-date program on the professional and academic field.

After the student has passed the assessments, they will receive their corresponding Hybrid Professional Master's Degree certificate issued by TECH Technological University via tracked delivery*.

In addition to the Certificate, students will be able to obtain an academic transcript, as well as a certificate outlining the contents of the program. In order to do so, students should contact their academic advisor, who will provide them with all the necessary information.

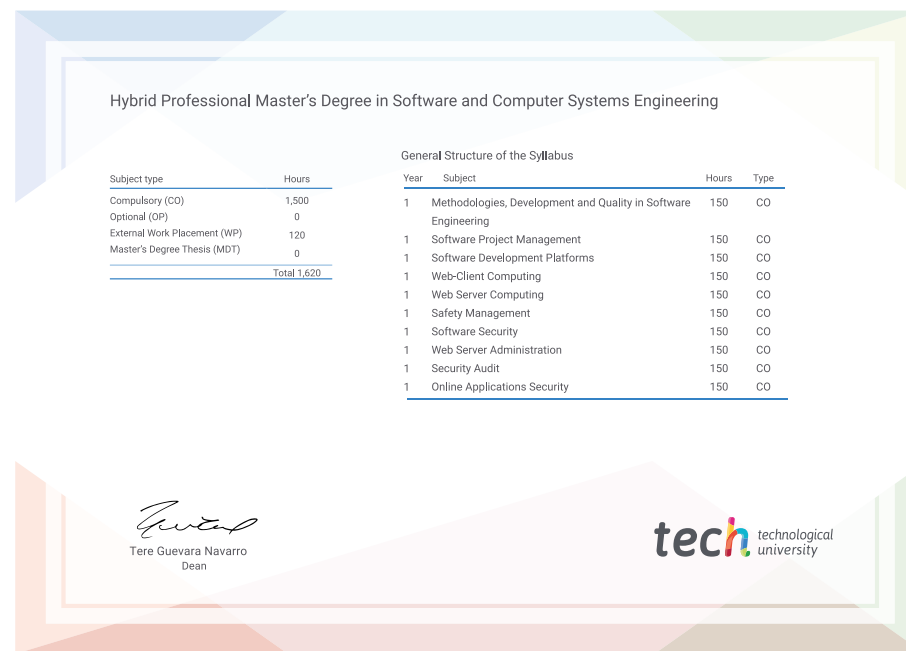
Title: **Professional Master's Degree in Software and Computer Systems Engineering**

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

Duration: **12 months.**

Certificate: **TECH Technological University**

Teaching Hours: **1,620 h.**



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



Hybrid Professional Master's Degree Software and Computer Systems Engineering

Modality: Hybrid (Online + Internship)

Duration: 12 months.

Certificate: TECH Technological University

Teaching Hours: 1,620 hours.

Hybrid Professional Master's Degree Software and Computer Systems Engineering