Professional Master's Degree

Creation of Interfaces and Network Applications

tech technological university

tech technological university

Professional Master's Degree
Creation of Interfaces and Network Applications

- Modality: **online**
- Duration: **12 months**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Website: www.techtitute.com/in/information-technology/professional-master-degree/master-creation-interfaces-network-applications

Index

01 Introduction p. 4

02 Objectives p. 8

03 Skills p. 14

04 Structure and Content p. 18

05 Methodology p. 32

06 Certificate p. 40

01

Introduction

Intervening competitively in the field of computing requires solid knowledge that includes the most relevant novelties and updates in software engineering, mathematical bases, statistics and other areas. In this program we have built an intensive path of specialization through which the professionals will be able to prepare scientifically and technologically in computer engineering, with quality, certainty and an ultimate goal of excellence.

Our innovative telepractice concept will give you the opportunity to learn through an immersive experience, which will provide you with a faster integration and a much more realistic view of the contents: "Learning from an expert"

This program will develop the necessary concepts to work in the creation of Interfaces, with the certainty that the mastery of all the areas of knowledge transversal to this subject, provide to the specialists. Throughout the syllabus, innovative educational approaches will be proposed to delve into the architecture of a distributed application. The particular aspects of the client-server architecture will be presented, and the fundamentals and essential developments of the most used programming languages will be discussed, differentiating between languages among many other fundamental issues for the professional.

This essential knowledge also becomes the first step to access the development capacity of this type of technology.

Throughout this program, a real working scenario is offered in order to be able to assess the convenience of its application in this project, evaluating its real indications, way of development and possible result expectations.

Through experience, it is possible to learn how to develop the necessary knowledge to advance in this area of work. This knowledge, which necessarily requires experience, is achieved in this specialization, reconciling distance learning and practical teaching, offering a unique option to give your resume the boost you are looking for.

This **Professional Master's Degree in Creation of Interfaces and Network Applications** contains the most complete and up-to-date program on the market. Outstanding features:

- The latest technology in online teaching software
- A highly visual teaching system, supported by graphic and schematic contents that are easy to assimilate and understand
- Practical cases presented by practising experts
- State-of-the-art interactive video systems
- Teaching supported by telepractice
- Continuous updating and recycling systems
- Autonomous learning: full compatibility with other occupations
- Practical exercises for self-assessment and learning verification
- Support groups and educational synergies: questions to the expert, debate and knowledge forums
- Communication with the teacher and individual reflection work
- Content that is available from any fixed or portable device with an Internet connection
- Supplementary documentation databases are permanently available, even after the program

Become one of the most demanded professionals of the moment: study computer engineering through the most complete and up-to-date Professional Master's Degree in the Creation of Interfaces and Network Applications"

With a methodological design based on proven teaching techniques, this innovative Professional Master's Degree in Creation of Interfaces and Network Applications will take you through different teaching approaches to allow you to learn in a dynamic and effective way"

Our teaching staff is made up of professionals from different fields related to this specialty. In this way TECH ensures that it delivers the educational up-to-date objectives that it aims for. A multidisciplinary team of professionals prepared and experienced in different areas, will cover the theoretical knowledge in an efficient way, but above all, will bring practical knowledge from their own experience to the program: one of the factors that makes this program unique.

This mastery of the subject matter is complemented by the effectiveness of the methodological design. Developed by a multidisciplinary team of *e-learning* experts, it integrates the latest advances in educational technology. As such, students will be able to study with a range of convenient and versatile multimedia tools that will give them the operability they need in their education.

The design of this program is based on Problem-Based Learning: an approach that conceives learning as a highly practical process. To achieve this remotely, telepractice will be used: with the help of an innovative system of interactive videos, and learning from an expert you will be able to acquire the knowledge as if you were facing the case you are learning at that moment. A concept that will make it possible to integrate and fix learning in a more realistic and permanent way.

A Professional Master's Degree that will enable you to work in all areas of the Creation of Interfaces and Network Applications with the solvency of a high-level professional.

With the experience of working professionals who will provide you with a real, immediate and concrete knowledge of this field of work.

02

Objectives

The objective of TECH is to provide the professional with updated information on the creation of network interfaces and applications. This is so that the computer engineer acquires new skills for the creation of more complete computer programs. A goal that, in just a few months, can be achieved through a resume focused on the current needs of the field.

Expand your knowledge in computer science and software engineering and get ready to compete among the best in the industry"

General Objectives

- Scientific and technological skills for computer engineering
- Obtain comprehensive knowledge in the field of computer science
- Obtain comprehensive knowledge in the field of computer structure
- Acquire the necessary knowledge in software engineering
- Review the mathematical, statistical and physical bases essential for this subject

A path to achieve education and professional growth that will propel you towards a greater level of competitiveness in the employment market"

Specific Objectives

Module 1. Human-Computer Interaction

- Acquire solid knowledge related to human-computer interaction and the creation of usable interfaces
- Understand the importance of application usability and why it is important to take it into account when designing our software
- Understand the different types of human diversity, the limitations they imply and how to adapt interfaces according to the specific needs of each of them
- Learn the process of interface design, from requirements analysis to evaluation, going through the different intermediate stages necessary to carry out an adequate interface
- Know the different accessibility guidelines, the standards that establish them and the tools that allow us to assess them
- Understand the different methods of interaction with the computer, by means of peripherals and devices

Module 2. Databases

- Learn the different applications and purposes of database systems, as well as their operation and architecture
- Understand the relational model, from its structure and operations to extended relational algebra
- Learn in depth what SQL databases are, how they work, the definition of data and the creation of queries from the most basic to the most advanced and complex
- Learn how to design databases using the entity-relationship model, how to create cdiagrams and the characteristics of the extended E-R model

- Delve into the design of relational databases, analyzing the different normal forms and decomposition algorithms
- Laying the groundwork for understanding the operation of NoSQL databases, as well as introducing the Mongo DB database

Module 3. Development of Web Applications

- Know the characteristics of the HTML markup language and its use in web creation together with CSS style sheets
- Learn how to use the browser-oriented programming language JavaScript, and some of its main features
- Understand the concepts of component-oriented programming and the component architecture
- Learn how to use the Bootstrap front-end framework for website design
- Understand the structure of the controller view model in the development of dynamic web sites
- Know the service-oriented architecture and the basics of the HTTP protocol

Module 4. Free Software and Open Knowledge

- Learn the concepts of Free Software and Open Knowledge, as well as the different types of associated licenses
- Know the main free tools available in different areas such as operating systems, business management, content management systems and multimedia content creation, among others
- Understand the importance and benefits of free software in the business world, both for its features and costs
- Delve into the knowledge of the GNU/Linux operating system, as well as the different existing distributions, and how you can make custom adaptations of them
- Learn about the operation and development of WordPress, given that this CMS accounts for more than 35% of the active websites in the world, and more than 60% in the particular case of CMSs
- Understand how the operating system for Android mobile devices works, as well as the basics for the development of mobile applications: both native development and with cross-platform frameworks

Module 5. Advanced Databases

- Introduce the different database systems currently available on the market
- Learn the use of XML and databases for the web
- Understand the operation of advanced databases such as parallel and distributed databases
- Understand the importance of indexing and association in database systems
- Understand how transactional processing and retrieval systems work
- Acquire knowledge related to non-relational databases and data mining

Module 6. Software Engineering

- Know the software engineering framework and the ISO/IEC 12207 standard
- Learn the characteristics of the unified software development process and planning in the context of agile software development
- Learn the different styles of distributed software design and service-oriented software architectures
- Learn the essential concepts in graphical user interface design
- Understand the basics of web application development
- Delve into software testing strategies and techniques, software quality factors and different metrics used

Module 7. Advanced Programming

- Delve into the knowledge of programming, especially as it relates to object-oriented programming, and the different types of relationships between object-oriented programming, and the different types of relationships between existing classes
- Know the different design patterns for object-oriented problems
- Learning about event-driven programming and user interface development with Qt
- Acquire the essential knowledge of concurrent programming, processes and threads
- Learn how to manage the use of threads and synchronization, as well as the resolution of common problems within Concurrent Programming
- Understand the importance of documentation and testing in software development

Module 8. Software Reuse

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

Module 9. Artificial Intelligence and Knowledge Engineering

- Lay the foundations of artificial intelligence and knowledge engineering, making a brief tour through the history of artificial intelligence up to the present day
- Understand the essential concepts of search in artificial intelligence, both informed and uninformed search
- Understand how artificial intelligence works in games
- Learn the fundamental concepts of neural networks and the use of genetic algorithms
- Acquire the appropriate mechanisms to represent knowledge, especially taking into account the semantic web
- Understand the functioning of expert systems and decision support systems

Module 10. Advanced Software Engineering

- Delve into the different agile methodologies used in software engineering
- Learn to develop using Scrum, extreme programming and reuse-based software development techniques
- Understand the different patterns of system architectures and software design, as well as the architecture of cloud applications
- Learn how to test software, with methodologies such as Test-Driven Development, Acceptance Test-Driven Development, Behavior-Driven Development, BDD and Cucumber
- Deepen the improvement of the software development process and software quality using ISO/IEC standards
- Introduce the DevOps concept and its main practices

03

Skills

This program in Creation of Interfaces and Network Applications has been created as a high-quality specialization tool for professionals. Its intensive syllabus will contribute significantly to the development and work of programmers and web portals, taking advantage of existing resources on the Internet and free software, in a practical and useful orientation.

With the skills that this Professional Master's Degree will allow you to acquire, you will be able to start developing applications and work as a software engineer, with the solvency of a complete and up-to-date program"

General Skill

- Acquire the necessary skills for the professional practice of computer engineering with the knowledge of all the necessary factors to carry it out with quality and solvency

An opportunity created for educators who are looking for an intensive and effective program to take a significant step forward in their career"

Specific Skills

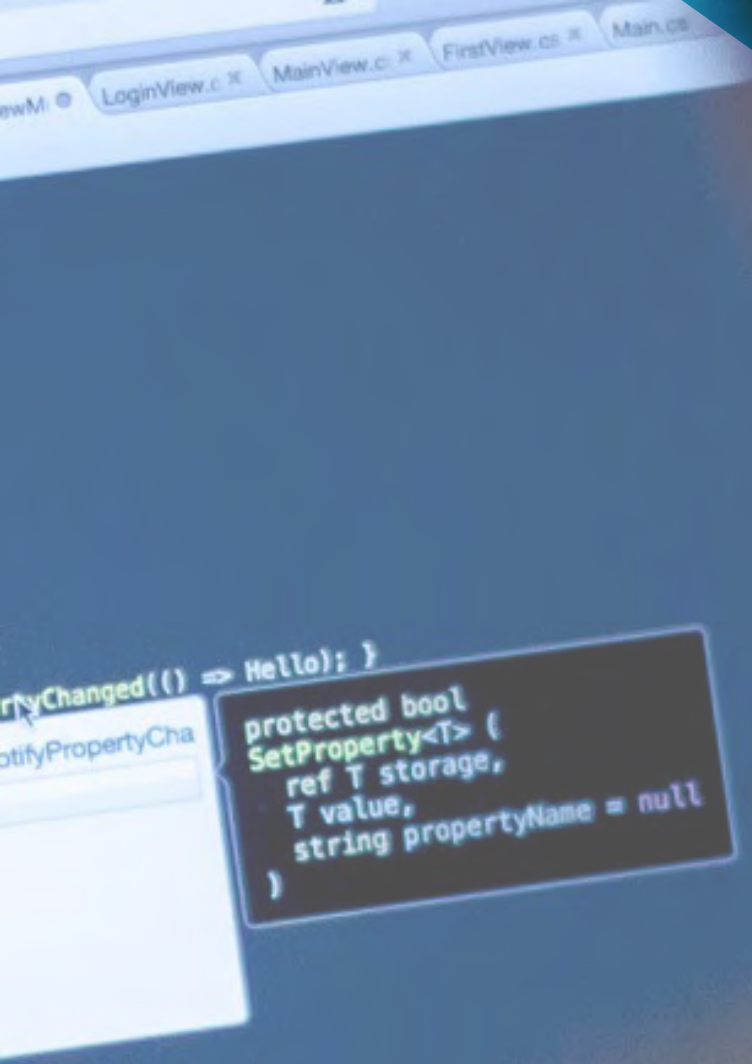
- Have an in-depth knowledge of all facets of human-computer interaction and how they involve IT developments
- Be proficient in the use of databases
- Develop different types of network applications
- Describe and take advantage of free software and open knowledge on the web
- Work as a software engineer
- Control the use of advanced databases
- Perform advanced programming
- Know how to reuse software
- Create interfaces and network applications
- Have mastery of the different systems of work in advanced software engineering

04

Structure and Content

The contents of this Professional Master's Degree have been developed by the different experts of this program, with a clear purpose: to ensure that our students acquire each and every one of the necessary skills to become true experts in this field.

A complete and well-structured program will take you to the highest standards of quality and success.



A complete teaching program, structured in complete and updated educational units based on the latest advances in the sector, oriented to a learning process compatible with your personal and professional life"

Module 1. Human-Computer Interaction

1.1. Introduction to Human-Computer Interaction
- 1.1.1. What is Human-Computer Interaction
- 1.1.2. Relationship of Human-Computer Interaction with Other Disciplines
- 1.1.3. The User Interface
- 1.1.4. Usability and Accessibility
- 1.1.5. User Experience and User-Centered Design

1.2. The Computer and Interaction: User Interface and Interaction Paradigms
- 1.2.1. Interaction
- 1.2.2. Paradigms and Styles of Interaction
- 1.2.3. Evolution of User Interfaces
- 1.2.4. Classic User Interfaces: WIMP/GUI, Commands, Voice, Virtual Reality
- 1.2.5. Innovative User Interfaces: Mobile, Wearable, Collaborative, BCI

1.3. The Human Factor: Psychological and Cognitive Aspects
- 1.3.1. The Importance of the Human Factor in Interaction
- 1.3.2. Human Information Processing
- 1.3.3. The Input and Output of Information: Visual, Auditory, and Tactile
- 1.3.4. Perception and Attention
- 1.3.5. Knowledge and Mental Models: Representation, Organization, and Acquisition

1.4. The Human Factor: Sensory and Physical Limitations
- 1.4.1. Functional Diversity, Disability and Impairment
- 1.4.2. Visual Diversity
- 1.4.3. Hearing Diversity
- 1.4.4. Cognitive Diversity
- 1.4.5. Motor Diversity
- 1.4.6. The Case of Digital Immigrants

1.5. The Design Process (I): Requirements Analysis for User Interface Design
- 1.5.1. User-Centered Design
- 1.5.2. What is Requirements Analysis?
- 1.5.3. Information Gathering
- 1.5.4. Analysis and Interpretation of the Information
- 1.5.5. Usability and Accessibility Analysis

1.6. The Design Process (II): Prototyping and Task Analysis
- 1.6.1. Conceptual Design
- 1.6.2. Prototyping
- 1.6.3. Hierarchical Task Analysis

1.7. The Design Process (III): Evaluation
- 1.7.1. Evaluation in the Design Process: Objectives and Methods
- 1.7.2. Evaluation Methods Without Users
- 1.7.3. Evaluation Methods with Users
- 1.7.4. Evaluation Standards and Norms

1.8. Accessibility: Definition and Guidelines
- 1.8.1. Accessibility and Universal Design
- 1.8.2. The WAI Initiative and the WCAG Guidelines
- 1.8.3. WCAG 2.0. and 2.1. Guidelines

1.9. Accessibility: Evaluation and Functional Diversity
- 1.9.1. Web Accessibility Evaluation Tools
- 1.9.2. Accessibility and Functional Diversity

1.10. The Computer and Interaction: Peripherals and Devices
- 1.10.1. Traditional Devices and Peripherals
- 1.10.2. Alternative Devices and Peripherals
- 1.10.3. Cell Phones and Tablets
- 1.10.4. Functional Diversity, Interaction and Peripherals

Module 2. Databases

2.1. Applications and Purposes of Database Systems
- 2.1.1. Applications of the Different Database Systems
- 2.1.2. Purpose of the Different Database Systems
- 2.1.3. View of the Data

2.2. Database and Architecture
- 2.2.1. Relational Database
- 2.2.2. Database Design
- 2.2.3. Object-Based and Semi-Structured Databases
- 2.2.4. Data Storage and Queries
- 2.2.5. Transaction Management
- 2.2.6. Data Mining and Analysis
- 2.2.7. Database Architecture

2.3. The Relational Model: Structure, Operations and Extended Relational Algebra
- 2.3.1. The Structure of Relational Databases
- 2.3.2. Fundamental Operations in the Relational Algebra
- 2.3.3. Other Relational Algebra Operations
- 2.3.4. Extended Relational Algebra Operations
- 2.3.5. Null Values
- 2.3.6. Database Modification

2.4. SQL (I)
- 2.4.1. What is SQL?
- 2.4.2. The Definition of Data
- 2.4.3. Basic Structure of SQL Queries
- 2.4.4. Operations on Sets
- 2.4.5. Aggregation Functions
- 2.4.6. Null Values

2.5. SQL (II)
- 2.5.1. Nested Subqueries
- 2.5.2. Complex Queries
- 2.5.3. Views
- 2.5.4. Cursors
- 2.5.5. Complex Queries
- 2.5.6. Triggers

2.6. Database Design and the E-R Model
- 2.6.1. Overview of the Design Process
- 2.6.2. The Entity-Relationship Model
- 2.6.3. Restrictions

2.7. Entity-Relationship Diagrams
- 2.7.1. Entity-Relationship Diagrams
- 2.7.2. Entity-Relationship Design Aspects
- 2.7.3. Weak Entity Sets

2.8. The Extended Entity-Relationship Model
- 2.8.1. Characteristics of the Extended E-R Model
- 2.8.2. Design of a Database
- 2.8.3. Reduction to Relational Schemas

2.9. Designing from Relational Databases
- 2.9.1. Characteristics of Good Relational Designs
- 2.9.2. Atomic Domains and the First Normal Form (1FN)
- 2.9.3. Decomposition by Functional Dependencies
- 2.9.4. Theory of Functional Dependencies
- 2.9.5. Decomposition Algorithms
- 2.9.6. Decomposition by Means of Multivalued Dependencies
- 2.9.7. More Normal Forms
- 2.9.8. Database Design Process

2.10. NoSQL Databases
- 2.10.1. What are NoSQL Databases?
- 2.10.2. Analysis of the Different NoSQL Options and their Characteristics.
- 2.10.3. Mongo DB

Module 3. Development of Web Applications

3.1. HTML5 Markup Languages
- 3.1.1. HTML Basics
- 3.1.2. New HTML 5 Elements
- 3.1.3. Forms: New Controls

3.2. Introduction to CSS Style Sheets
- 3.2.1. First Steps with CSS
- 3.2.2. Introduction to CSS3

3.3. Browser Scripting Language: JavaScript
- 3.3.1. JavaScript Basics
- 3.3.2. DOM
- 3.3.3. Events
- 3.3.4. JQuery
- 3.3.5. Ajax

3.4. Concept of Component-Oriented Programming
- 3.4.1. Context
- 3.4.2. Components and Interfaces
- 3.4.3. States of a Component

3.5. Component Architecture
- 3.5.1. Current Architectures
- 3.5.2. Component Integration and Deployment

3.6. Framework Front-End: Bootstrap
- 3.6.1. Grid Design
- 3.6.2. Forms
- 3.6.3. Components

3.7. Model View Controller
- 3.7.1. Web Development Methods
- 3.7.2. Design Pattern: MVC

3.8. Information Grid Technologies
- 3.8.1. Increased Computing Resources
- 3.8.2. Concept of Grid Technology

3.9. Service-Oriented Architecture
- 3.9.1. SOA and Web Services
- 3.9.2. Topology of a Web Service
- 3.9.3. Platforms for Web Services

3.10. HTTP Protocol
- 3.10.1. Messages
- 3.10.2. Persistent Sessions
- 3.10.3. Cryptographic System
- 3.10.4. HTTPS Protocol Operation

Module 4. Free Software and Open Knowledge

4.1. Introduction to Free Software
- 4.1.1. History of Free Software
- 4.1.2. "Freedom" in Software
- 4.1.3. Licenses for the Use of Software Tools
- 4.1.4. Intellectual Property of Software
- 4.1.5. What is the Motivation for Using Free Software?
- 4.1.6. Free Software Myths
- 4.1.7. Top500

4.2. Open Knowledge and CC Licenses
- 4.2.1. Basic Concepts
- 4.2.2. Creative Commons Licenses
- 4.2.3. Other Content Licenses
- 4.2.4. Wikipedia and Other Open Knowledge Projects

4.3. Main Free Software Tools
- 4.3.1. Operating Systems
- 4.3.2. Office Applications
- 4.3.3. Business Management Applications
- 4.3.4. Web Content Managers
- 4.3.5. Multimedia Content Creation Tools
- 4.3.6. Other Applications

4.4. The Company: Free Software and its Costs
- 4.4.1. Free Software: Yes or No?
- 4.4.2. Truths and Lies about Free Software
- 4.4.3. Business Software Based on Free Software
- 4.4.4. Software Costs
- 4.4.5. Free Software Models

4.5. The GNU/Linux Operating System
- 4.5.1. Architecture
- 4.5.2. Basic Directory Structure
- 4.5.3. File System Characteristics and Structure
- 4.5.4. Internal Representation of the Files

4.6. The Android Mobile Operating System
- 4.6.1. History
- 4.6.2. Architecture
- 4.6.3. Android Forks
- 4.6.4. Introduction to Android Development
- 4.6.5. Frameworks for Mobile Application Development

4.7. Website Creation with WordPress
- 4.7.1. WordPress Features and Structure
- 4.7.2. Creation of Sites on WordPress.com
- 4.7.3. Installation and Configuration of WordPress on your own Server
- 4.7.4. Installing Plugins and Extending WordPress
- 4.7.5. Creation of WordPress Plugins
- 4.7.6. WordPress Theme Creation

4.8. Free Software Trends
- 4.8.1. Cloud-Based Environments
- 4.8.2. Monitoring Tools
- 4.8.3. Operating Systems
- 4.8.4. Big Data and Open Data 2.0.
- 4.8.5. Quantum Computing

4.9. Version Control
- 4.9.1. Basic Concepts
- 4.9.2. Git
- 4.9.3. Cloud and Self-hosted Git Services
- 4.9.4. Other Version Control Systems

4.10. Custom GNU/Linux Distributions
- 4.10.1. Main Distributions
- 4.10.2. Distributions Derived from Debian
- 4.10.3. Deb Package Creation
- 4.10.4. Modification of the Distribution
- 4.10.5. ISO Image Generation

Module 5. Advanced Databases

5.1. Introduction to the Different Database Systems
- 5.1.1. Historical Recap
- 5.1.2. Hierarchical Databases
- 5.1.3. Network Databases
- 5.1.4. Relational Databases
- 5.1.5. Non-Relational Databases

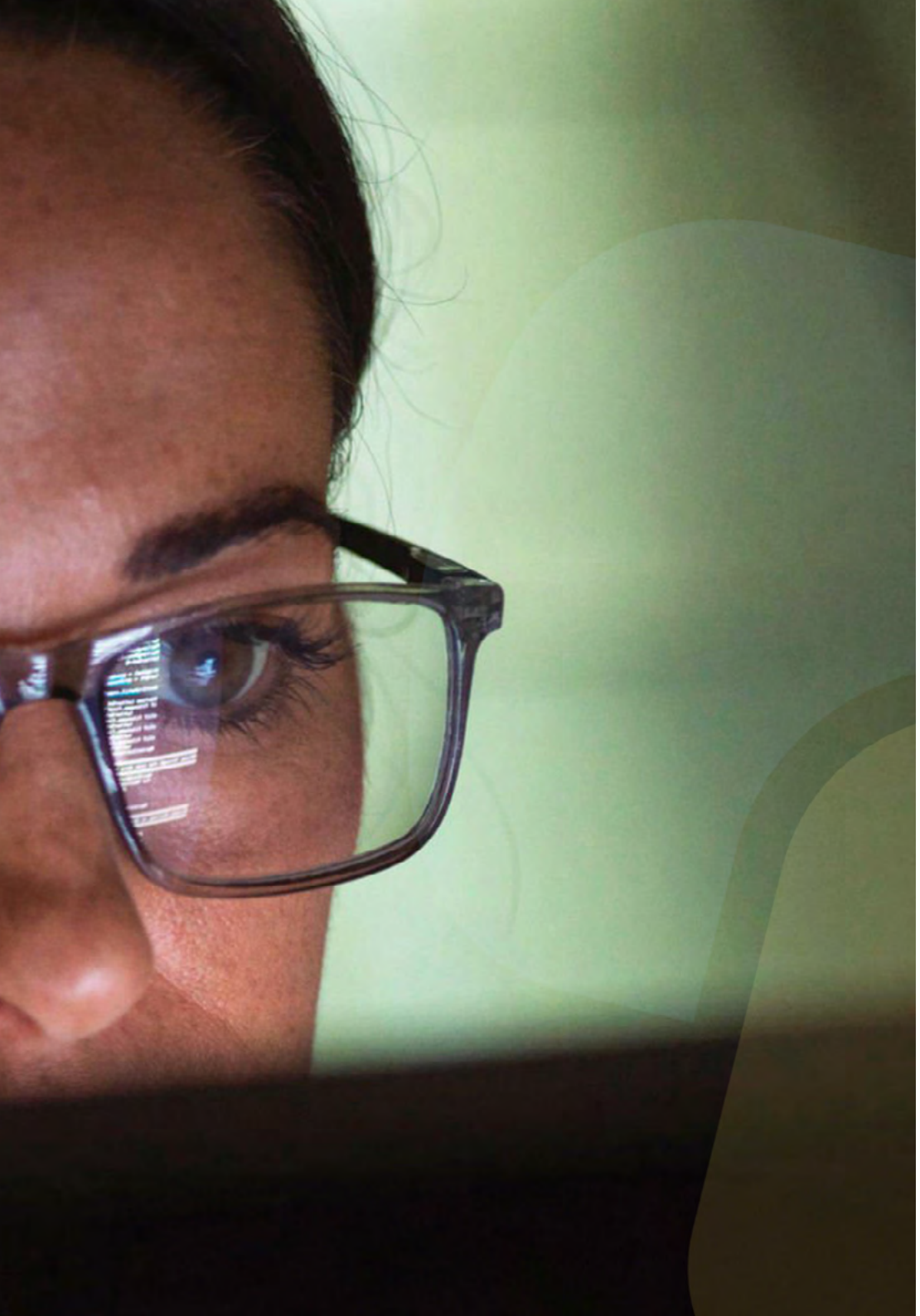
5.2. XML and Databases for the Web
- 5.2.1. Validation of XML Documents
- 5.2.2. XML Document Transformations
- 5.2.3. XML Data Storage
- 5.2.4. XML Relational Databases
- 5.2.5. SQL/XML
- 5.2.6. Native XML Databases

5.3. Parallel Databases
- 5.3.1. Parallel Systems
- 5.3.2. Parallel Database Architectures
- 5.3.3. Parallelism in Queries
- 5.3.4. Query Parallelism
- 5.3.5. Design of Parallel Systems
- 5.3.6. Parallel Processing in SQL

5.4. Distributed Databases
- 5.4.1. Distributed Systems
- 5.4.2. Distributed Storage
- 5.4.3. Availability
- 5.4.4. Distributed Query Processing
- 5.4.5. Distributed Database Providers

5.5. Indexing and Association
- 5.5.1. Ordered Indexes
- 5.5.2. Dense and Sparse Indexes
- 5.5.3. Multilevel Indices
- 5.5.4. Index Updating
- 5.5.5. Static Association
- 5.5.6. How to Use Indexes in Databases

5.6. Introduction to Transactional Processing
- 5.6.1. States of a Transaction
- 5.6.2. Implementation of atomicity and durability.
- 5.6.3. Sequentiality
- 5.6.4. Recoverability
- 5.6.5. Isolation Implementation



5.7. Recovery Systems
- 5.7.1. Failure Classification
- 5.7.2. Storage Structures
- 5.7.3. Recovery and Atomicity
- 5.7.4. Retrieval Based on Historical Record
- 5.7.5. Concurrent Transactions and Retrieval
- 5.7.6. High Availability in Databases

5.8. Execution and Processing of Queries
- 5.8.1. Cost of a Query
- 5.8.2. Selection Operation
- 5.8.3. Sorting
- 5.8.4. Introduction to Query Optimization
- 5.8.5. Performance Monitoring

5.9. Non-Relational Databases
- 5.9.1. Document-Oriented Databases
- 5.9.2. Graph-Oriented Databases
- 5.9.3. Key-Value Databases

5.10. Data Warehouse, OLAP and Data Mining
- 5.10.1. Components of Data Warehouses
- 5.10.2. Architecture of a Data Warehouse
- 5.10.3. OLAP
- 5.10.4. Data Mining Functionality
- 5.10.5. Other Types of Mining

Module 6. Software Engineering

6.1. Software Engineering Framework
- 6.1.1. Software Features
- 6.1.2. The Main Processes in Software Engineering
- 6.1.3. Software Development Process Models
- 6.1.4. Standard Reference Framework for the Software Development Process: The ISO/IEC 12207 Standard

6.2. Unified Software Development Process
- 6.2.1. The Unified Process
- 6.2.2. Dimensions of the Unified Process
- 6.2.3. Case Studies Driven Development Process
- 6.2.4. Fundamental Workflows of Unified Processes

6.3. Planning in the Context of Agile Software Development
- 6.3.1. Characteristics of Agile Software Development
- 6.3.2. Different Planning Time Horizons in Agile Development
- 6.3.3. Scrum Agile Development Framework and Planning Time Horizons
- 6.3.4. User Stories as a Planning and Estimating Unit
- 6.3.5. Common Techniques for Deriving an Estimate
- 6.3.6. Scales for Interpreting Estimates
- 6.3.7. Planning Poker
- 6.3.8. Common Scheduling Types: Delivery Scheduling and Iteration Scheduling

6.4. Distributed Software Design Styles and Service-Oriented Software Architectures
- 6.4.1. Communication Models in Distributed Software Systems
- 6.4.2. Middleware
- 6.4.3. Architecture Patterns for Distributed Systems
- 6.4.4. General Software Service Design Process
- 6.4.5. Design Aspects of Software Services
- 6.4.6. Composition of Services
- 6.4.7. Web Services Architecture
- 6.4.8. Infrastructure and SOA Components

6.5. Introduction to Model Driven Software Development
- 6.5.1. The Model Concept
- 6.5.2. Model-Driven Software Development
- 6.5.3. MDA Model-Driven Development Framework
- 6.5.4. Elements of a Transformation Model

6.6. Graphical User Interface Design
- 6.6.1. Principles of User Interface Design
- 6.6.2. Architectural Design Patterns for Interactive Systems: Model View Controller (MVC)
- 6.6.3. UX User Experience
- 6.6.4. User-Centered Design
- 6.6.5. Graphical User Interface Analysis and Design Process
- 6.6.6. Usability of User Interfaces
- 6.6.7. Accessibility in User Interfaces

6.7. Web Application Design
- 6.7.1. Characteristics of Web Applications
- 6.7.2. Web Application User Interface
- 6.7.3. Navigation Design
- 6.7.4. Basic Interaction Protocol for Web Applications
- 6.7.5. Architecture Styles for Web Applications

6.8. Software Testing Strategies and Techniques and Software Quality Factors
- 6.8.1. Testing Strategies
- 6.8.2. Test Case Designs
- 6.8.3. Value for Money
- 6.8.4. Quality Models
- 6.8.5. ISO/IEC 25000 Family of Standards (SQuaRE)
- 6.8.6. Product Quality Model (ISO 2501n)
- 6.8.7. Data Quality Models (ISO 2501n)
- 6.8.8. Software Quality Management

6.9. Introduction to Software Engineering Metrics
- 6.9.1. Basic Concepts: Measurements, Metrics and Indicators
- 6.9.2. Types of Metrics in Software Engineering
- 6.9.3. The Measurement Process
- 6.9.4. ISO 25024. External and Quality Metrics in Use
- 6.9.5. Object-Oriented Metrics

6.10. Software Maintenance and Reengineering
- 6.10.1. Maintenance Process
- 6.10.2. Standard Maintenance Process Framework. ISO/EIEC 14764
- 6.10.3. Software Reengineering Process Model
- 6.10.4. Inverse Engineering

Module 7. Advanced Programming

7.1. Introduction to Object-Oriented Programming
- 7.1.1. Introduction to Object-Oriented Programming
- 7.1.2. Class Design
- 7.1.3. Introduction to UML for Problem Modeling

7.2. Relationships Between Classes
- 7.2.1. Abstraction and Inheritance
- 7.2.2. Advanced Inheritance Concepts
- 7.2.3. Polymorphism
- 7.2.4. Composition and Aggregation

7.3. Introduction to Design Patterns for Object-Oriented Problems
- 7.3.1. What are Design Patterns?
- 7.3.2. Factory Pattern
- 7.3.3. Singleton Pattern
- 7.3.4. Observer Pattern
- 7.3.5. Composite Pattern

7.4. Exceptions
- 7.4.1. What are Exceptions?
- 7.4.2. Exception Catching and Handling
- 7.4.3. Throwing Exceptions
- 7.4.4. Exception Creation

7.5. User Interfaces
- 7.5.1. Introduction to Qt
- 7.5.2. Positioning
- 7.5.3. What Are Events?
- 7.5.4. Events: Definition and Catching
- 7.5.5. User Interface Development

7.6. Introduction to Concurrent Programming
- 7.6.1. Introduction to Concurrent Programming
- 7.6.2. The Concept of Process and Thread
- 7.6.3. Interaction Between Processes or Threads
- 7.6.4. Threads in C++
- 7.6.5. Advantages and Disadvantages of Concurrent Programming

7.7. Thread Management and Synchronization
- 7.7.1. Life Cycle of a Thread
- 7.7.2. Thread Class
- 7.7.3. Thread Planning
- 7.7.4. Thread Groups
- 7.7.5. Daemon Threads
- 7.7.6. Synchronization
- 7.7.7. Locking Mechanisms
- 7.7.8. Communication Mechanisms
- 7.7.9. Monitors

7.8. Common Problems in Concurrent Programming
- 7.8.1. The Problem of Consuming Producers
- 7.8.2. The Problem of Readers and Writers
- 7.8.3. The Problem of the Philosophers' Dinner Party

7.9. Software Documentation and Testing
- 7.9.1. Why is it Important to Document Software?
- 7.9.2. Design Documentation
- 7.9.3. Documentation Tool Use

7.10. Software Testing
- 7.10.1. Introduction to Software Testing
- 7.10.2. Types of Tests
- 7.10.3. Unit Test
- 7.10.4. Integration Test
- 7.10.5. Validation Test
- 7.10.6. System Test

Module 8. Software Reuse

8.1. General Overview of the Software Reuse
- 8.1.1. What is Software Reuse?
- 8.1.2. Advantages and Disadvantages of Software Reuse
- 8.1.3. Main Techniques of Software Reuse

8.2. Introduction to Design Patterns
- 8.2.1. What is a Design Patterns?
- 8.2.2. Catalog of the Main Design Patterns
- 8.2.3. How to Use Patterns to Solve Design Problems
- 8.2.4. How to Select the Best Design Pattern

8.3. Creation Patterns
- 8.3.1. Creation Patterns
- 8.3.2. Abstract Factory Pattern
- 8.3.3. Example of Abstract Factory Pattern implementation
- 8.3.4. Builder Pattern
- 8.3.5. Builder Implementation Example
- 8.3.6. Abstract Factory Pattern vs. Builder

8.4. Creation Patterns (II)
- 8.4.1. Factory Method Pattern
- 8.4.2. Factory Method vs. Abstract Factory
- 8.4.3. Singleton Pattern

8.5. Structural Patterns
- 8.5.1. Structural Patterns
- 8.5.2. Adapter Pattern
- 8.5.3. Bridge Pattern

8.6. Structural Patterns (II)
- 8.6.1. Composite Pattern
- 8.6.2. Decorator Pattern

8.7. Structural Patterns (III)
- 8.7.1. Facade Pattern
- 8.7.2. Proxy Pattern

8.8. Behavioral Patterns
- 8.8.1. Concept of Behavioral Patterns
- 8.8.2. Behavior Pattern: Chain of Responsibility
- 8.8.3. Behavior Pattern Order

8.9. Behavioral Patterns (II)
- 8.9.1. Interpreter Pattern
- 8.9.2. Iterator Pattern
- 8.9.3. Observer Pattern
- 8.9.4. Strategy Pattern

8.10. Frameworks
- 8.10.1. Concept of Frameworks
- 8.10.2. Development using Frameworks
- 8.10.3. Model View Controller Pattern
- 8.10.4. Framework for Graphical User Interface Design
- 8.10.5. Frameworks for Web Application Development
- 8.10.6. Frameworks for Managing Object Persistence in Databases

Module 9. Artificial Intelligence and Knowledge Engineering

9.1. Introduction to Artificial Intelligence and Knowledge Engineering
- 9.1.1. Brief History of Artificial Intelligence
- 9.1.2. Artificial Intelligence Today
- 9.1.3. Knowledge Engineering

9.2. Searching
- 9.2.1. Common Search Concepts
- 9.2.2. Uninformed Search
- 9.2.3. Informed Search

9.3. Boolean Satisfiability, Constraint Satisfiability and Automatic Planning
- 9.3.1. Boolean Satisfiability
- 9.3.2. Constraint Satisfiability Problems
- 9.3.3. Automatic Planning and PDDL
- 9.3.4. Planning as a Heuristic Search
- 9.3.5. Planning with SAT

9.4. Artificial Intelligence in Games
- 9.4.1. Game Theory
- 9.4.2. Minimax and Alpha-Beta Pruning
- 9.4.3. Simulation: Monte Carlo

9.5. Supervised and Unsupervised Learning
- 9.5.1. Introduction to Machine Learning
- 9.5.2. Classification
- 9.5.3. Regression
- 9.5.4. Validation of Results
- 9.5.5. Clustering

9.6. Neural Networks
- 9.6.1. Biological Fundamentals
- 9.6.2. Computational Model
- 9.6.3. Supervised and Unsupervised Neural Networks
- 9.6.4. Simple Perceptron
- 9.6.5. Multilayer Perceptron

9.7. Genetic Algorithms
- 9.7.1. History
- 9.7.2. Biological Basis
- 9.7.3. Problem Coding
- 9.7.4. Generation of the Initial Population
- 9.7.5. Main Algorithm and Genetic Operators
- 9.7.6. Evaluation of Individuals: Fitness

9.8. Thesauri, Vocabularies, Taxonomies
- 9.8.1. Vocabulary
- 9.8.2. Taxonomy
- 9.8.3. Thesauri
- 9.8.4. Ontologies

9.9. Knowledge Representation: Semantic Web
- 9.9.1. Semantic Web
- 9.9.2. Specifications: RDF, RDFS and OWL
- 9.9.3. Inference/ Reasoning
- 9.9.4. Linked Data

9.10. Expert Systems and DSS
- 9.10.1. Expert Systems
- 9.10.2. Decision Support Systems

Module 10. Advanced Software Engineering

10.1. Introduction to Agile Methodologies
- 10.1.1. Process Models and Methodologies
- 10.1.2. Agility and Agile Processes
- 10.1.3. Agile Manifesto
- 10.1.4. Some Agile Methodologies
- 10.1.5. Agile vs. Traditional

10.2. Scrum
- 10.2.1. Origins and Philosophy of Scrum
- 10.2.2. Scrum Values
- 10.2.3. Scrum Process Flow
- 10.2.4. Scrum Roles
- 10.2.5. Scrum Artifacts
- 10.2.6. Scrum Events
- 10.2.7. User Stories
- 10.2.8. Scrum Extensions
- 10.2.9. Agile Estimates
- 10.2.10. Scrum Scaling

10.3. Extreme Programming
- 10.3.1. Justification and Overview of XP
- 10.3.2. The XP Life Cycle
- 10.3.3. The Five Core Values
- 10.3.4. The Twelve Basic Practices in XP
- 10.3.5. Roles of Participants
- 10.3.6. XP Industrial
- 10.3.7. Critical Assessment of XP

10.4. Software Development Based on Reusability
- 10.4.1. Software Reuse
- 10.4.2. Code Reuse Levels
- 10.4.3. Specific Reuse Techniques
- 10.4.4. Component-Based Development
- 10.4.5. Benefits and Problems of Reuse
- 10.4.6. Reuse Planning

10.5. System Architecture and Software Design Patterns
- 10.5.1. Architectural Design
- 10.5.2. General Architectural Patterns
- 10.5.3. Fault Tolerant Architectures
- 10.5.4. Distributed Systems Architectures
- 10.5.5. Design Patterns
- 10.5.6. Gamma Patterns
- 10.5.7. Interaction Design Patterns

10.6. Cloud Application Architecture
- 10.6.1. Cloud Computing Fundamentals
- 10.6.2. Cloud Application Quality
- 10.6.3. Architectural Styles
- 10.6.4. Design Patterns

10.7. Software Testing: TDD, ATDD and BDD
- 10.7.1. Software Verification and Validation
- 10.7.2. Software Testing
- 10.7.3. Test-Driven Development (TDD)
- 10.7.4. Acceptance Test-Driven Development (ATDD)
- 10.7.5. Test-Driven Development (BDD)
- 10.7.6. BDD and Cucumber

10.8. Software Process Improvement
- 10.8.1. Software Process Improvement
- 10.8.2. The Process Improvement Approach
- 10.8.3. Maturity Models
- 10.8.4. The CMMI Model
- 10.8.5. CMMI V2.0.
- 10.8.6. CMMI and Agile

10.9. The Quality of the Software Product: SQuaRE
- 10.9.1. Software Quality
- 10.9.2. Software Product Quality Models
- 10.9.3. ISO/IEC 25000 Family
- 10.9.4. ISO/IEC 25010: Quality Model and Quality Characteristics
- 10.9.5. ISO/IEC 25012: the Quality of the Data
- 10.9.6. ISO/IEC 25020 Software Quality Measurement.
- 10.9.7. ISO/IEC 25022, 25023 and 25024: Software and Data Quality Metrics
- 10.9.8. ISO/IEC 25040 Software Assessment
- 10.9.9. Accreditation Process

10.10. Introduction to DevOps
- 10.10.1. DevOps Concept
- 10.10.2. Core Practices

A unique, key, and decisive experience to boost your professional development"

05

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

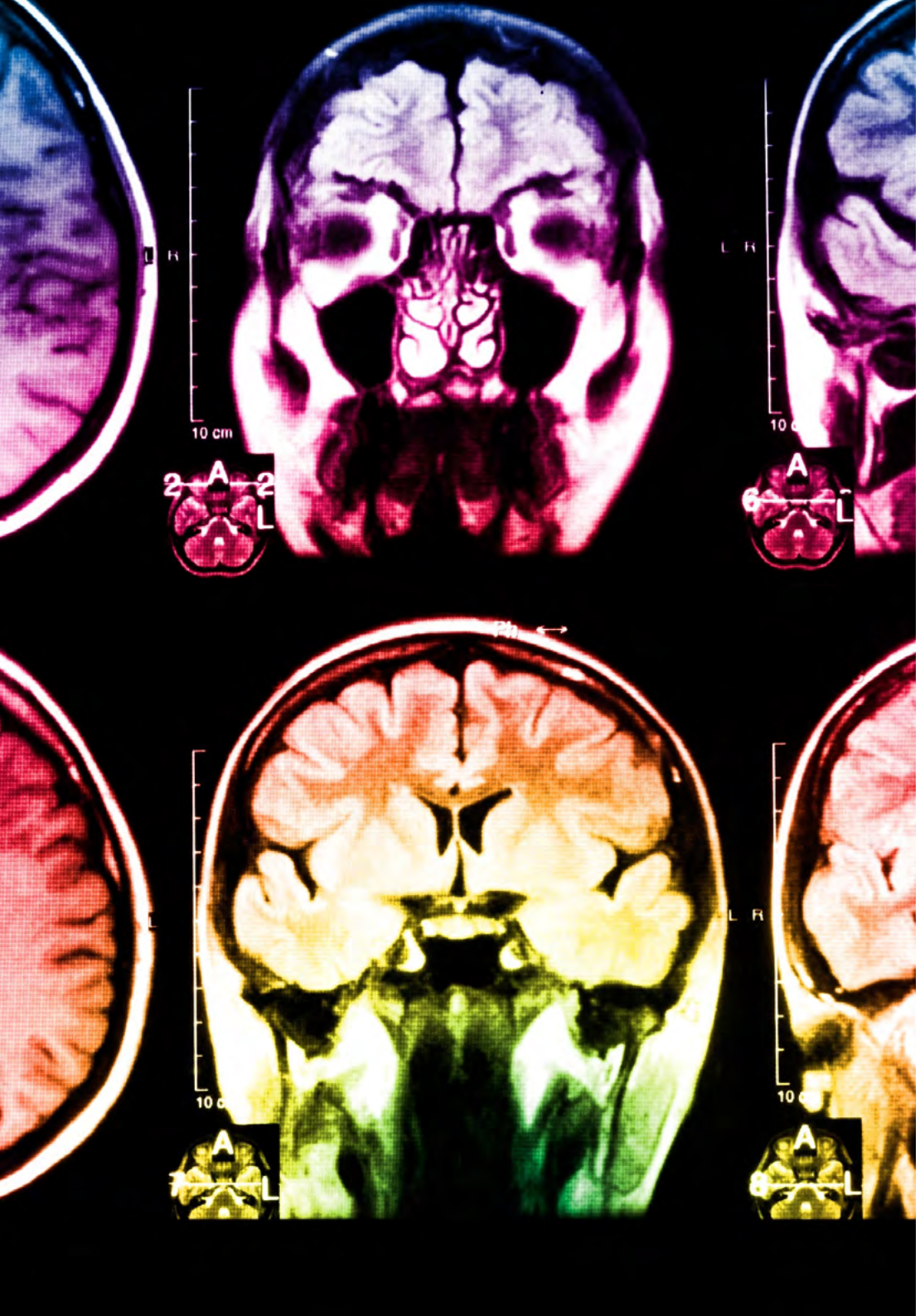
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

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.

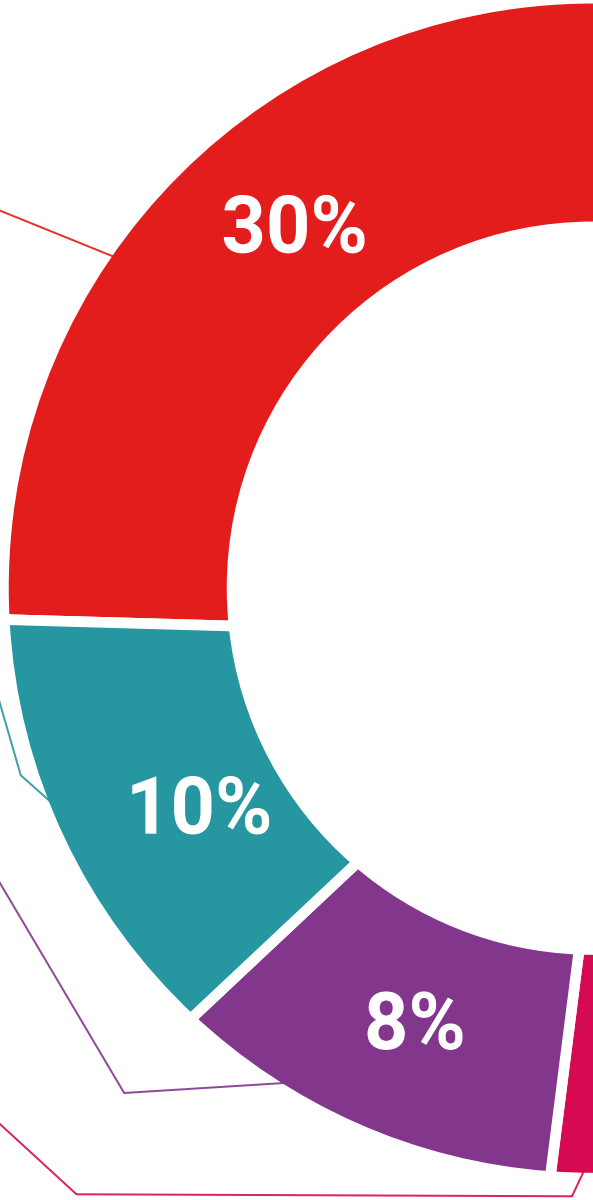
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

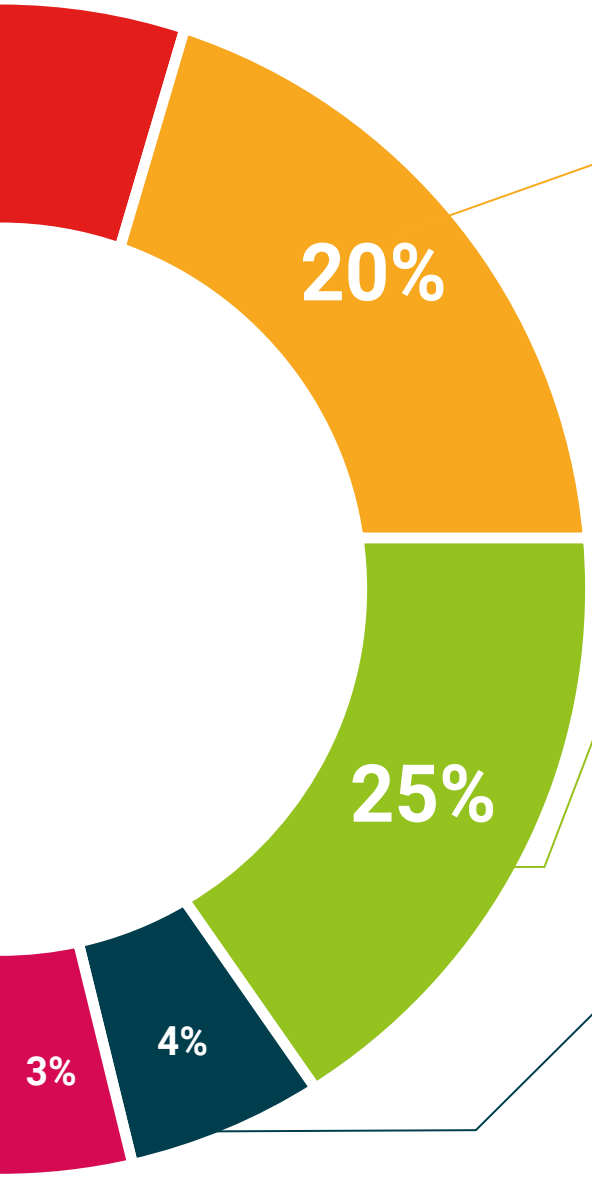
Practising Skills and Abilities

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.

06

Certificate

The Professional Master's Degree in Creation of Interfaces and Network Applications 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 degree without having to travel or fill out laborious paperwork"

This **Professional Master's Degree in Creation of Interfaces and Network Applications** contains the most complete and up-to-date program on the market.

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

The certificate issued by **TECH Technological University** will reflect the qualification obtained in the Professional Master's Degree, and meets the requirements commonly demanded by labor exchanges, competitive examinations, and professional career evaluation committees.

Degree: **Professional Master's Degree in Creation of Interfaces and Network Applications**

Official No. of Hours: **1,500 h.**

tech technological university

Awards the following

CERTIFICATE

to

Mr./Ms. _____, with identification number _____.

For having successfully passed and accredited the following program

PROFESSIONAL MASTER'S DEGREE

in

Creation of Interfaces and Network Applications

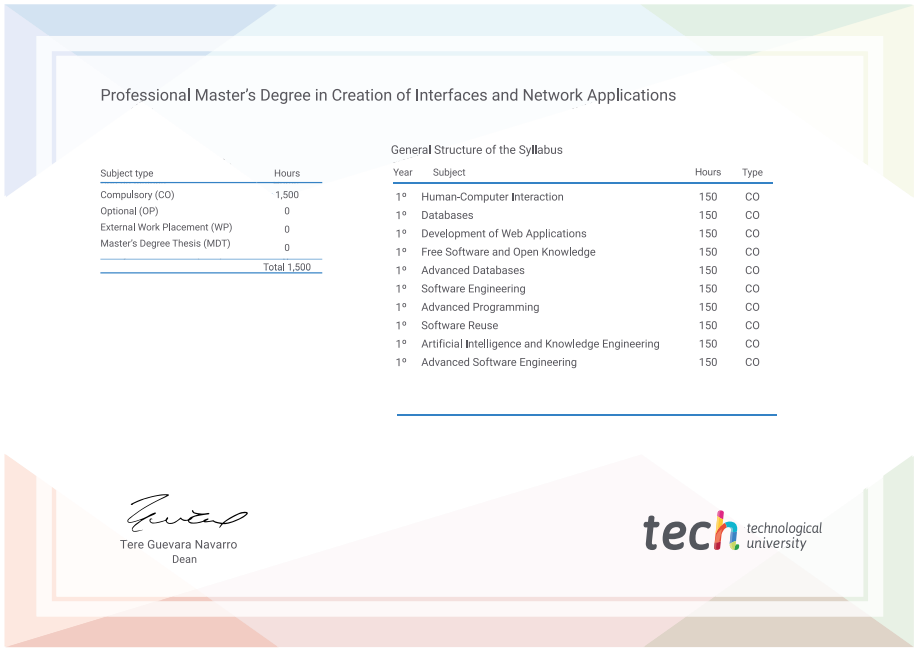
This is a qualification awarded by this University, equivalent to 1,500 hours, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH is a Private Institution of Higher Education recognized by the Ministry of Public Education as of June 28, 2018.

June 17, 2020

Tere Guevara Navarro
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates


Professional Master's Degree in Creation of Interfaces and Network Applications

Subject type	Hours
Compulsory (CO)	1,500
Optional (OP)	0
External Work Placement (WP)	0
Master's Degree Thesis (MDT)	0
	Total 1,500

General Structure of the Syllabus

Year	Subject	Hours	Type
1º	Human-Computer Interaction	150	CO
1º	Databases	150	CO
1º	Development of Web Applications	150	CO
1º	Free Software and Open Knowledge	150	CO
1º	Advanced Databases	150	CO
1º	Software Engineering	150	CO
1º	Advanced Programming	150	CO
1º	Software Reuse	150	CO
1º	Artificial Intelligence and Knowledge Engineering	150	CO
1º	Advanced Software Engineering	150	CO

Tere Guevara Navarro
Dean

tech technological university

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

tech technological university

Professional Master's Degree

Creation of Interfaces and Network Applications

- Modality: **online**
- Duration: **12 months**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Professional Master's Degree

Creation of Interfaces and Network Applications

tech technological university