Master's Degree
Software Development

False
= False
= True
at the end - add back the
_ob.select= 1
ifier_ob.select=1
context.scene.objects.active = modif
("Selected" + str(modifier_ob))
mirror_ob.select = 0
mirror modifier object
_ob
modifier ob is the active ob

tech global university

tech global university

Master's Degree
Software Development

- Modality: **online**
- Duration: **12 months.**
- Certificate: **TECH Global University**
- Accreditation: **60 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

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

Index

01

Introduction to the Program

Software Engineering is experiencing unprecedented growth, with a constant demand for experts capable of creating efficient, scalable, and secure solutions. According to the United Nations, more than 60% of companies face difficulties in filling vacancies in technological fields, particularly in Software Development. In this context, it is essential to have up-to-date knowledge aligned with the real needs of the computing environment. In this regard, TECH presents an innovative, 100% online qualification in the field of Software Development, designed for professionals who wish to advance toward a highly competitive technical profile adapted to the current challenges of the sector.

> *Access a university program of high academic value where you will learn to implement new trends in software development using AI technologies"*

Software Development is a strategic pillar for digital transformation across all productive sectors, from Healthcare and Education to Commerce. In this regard, the creation of efficient, scalable, and secure digital solutions makes a decisive difference in the competitiveness of companies and institutions. In this context, having specialists capable of designing, building, and maintaining complex technological systems has become a global priority, driving demand for highly qualified profiles in this field.

For this reason, TECH launches an innovative Master's Degree in Software Development. This program enables the acquisition of key competencies in programming, systems architecture, agile methodologies, and emerging technologies, always with a practical and up-to-date approach. Through a rigorous curriculum focused on solving real-world problems, the program fosters a comprehensive vision of Software Development. This preparation not only enhances career opportunities but also facilitates access to roles in technical leadership, consulting, and the design of innovative solutions with impact across various sectors.

It should be noted that the online format of the program allows participants to progress without geographical barriers or time constraints. Likewise, the virtual platform provides continuous access to high-quality content, interactive resources, and applied exercises, promoting flexible, autonomous learning adapted to each individual pace. This modality facilitates the balance with other professional and personal activities, while promoting an educational experience aligned with the current dynamics of the technological and professional environment. In addition, the program features the participation of a renowned International Guest Director who will deliver 10 rigorous Masterclasses.

This **Master's Degree in Software Development** offers the most complete and up-to-date university program on the market. Its most notable features are:

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

A prestigious International Guest Director will deliver 10 exclusive Masterclasses on the latest trends in Software Development"

You will develop distributed and networked applications that address the connectivity, security, and performance challenges required by digital transformation"

You will integrate relational databases into complex architectures, optimizing the management of large volumes of information.

Through the Relearning system used by TECH, you will reduce long hours of study and memorization.

The teaching staff includes professionals from the field of Software Development, who contribute their practical experience to the program, as well as renowned specialists from leading professional associations and prestigious universities.

Its multimedia content, developed using the latest educational technology, enables professionals to engage in situated and contextualized learning—that is, a simulated environment that provides an immersive learning experience designed to prepare them for real-world situations.

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

02

Why Study at TECH?

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

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

The world's best online university, according to FORBES

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

Forbes
The best online university in the world

The most complete syllabus

TOP international faculty

The most effective methodology

World's No.1
The World's largest online university

The most complete syllabuses on the university scene

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

The best top international faculty

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

A unique learning method

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

The world's largest online university

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

The official online university of the NBA

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

Leaders in employability

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

Google Partner

PREMIER 2025

The Official Online University of the NBA

4.9/5 ★★★★★ ★ global score

99% maximun employability guaranteed

Google Premier Partner

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

The top-rated university by its students

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

03
Syllabus

This syllabus integrates content aligned with the current demands of the technology sector, combining advanced theory with practical application. In addition, it incorporates approaches to cybersecurity, multiplatform development, and service-oriented architecture, providing a comprehensive and modern view of the computing environment. In this way, it offers a rigorous and up-to-date academic experience, designed to address the challenges of digital transformation with technical rigor and an innovative perspective.

```
sci/index.php?sid=' + Math.r
eckLangItem("contact-text-1")?></div>
}?>
<?if($send==2){?>
<td align="right" colspan="2">
<input type="button" value="<?=checkLangItem("contact-atpakal")?>" onclick="window.location='kontakti.ph
</td>
</tr>
<tr>
<td align="right" colspan="2">
<div class="text-2">
  <?=checkLangItem("contact-paldies")?>
</div>
</td>
```

You will master the essential foundations of Software Development to build solid, efficient technological solutions adapted to the needs of the professional environment"

Module 1. Programming Fundamentals

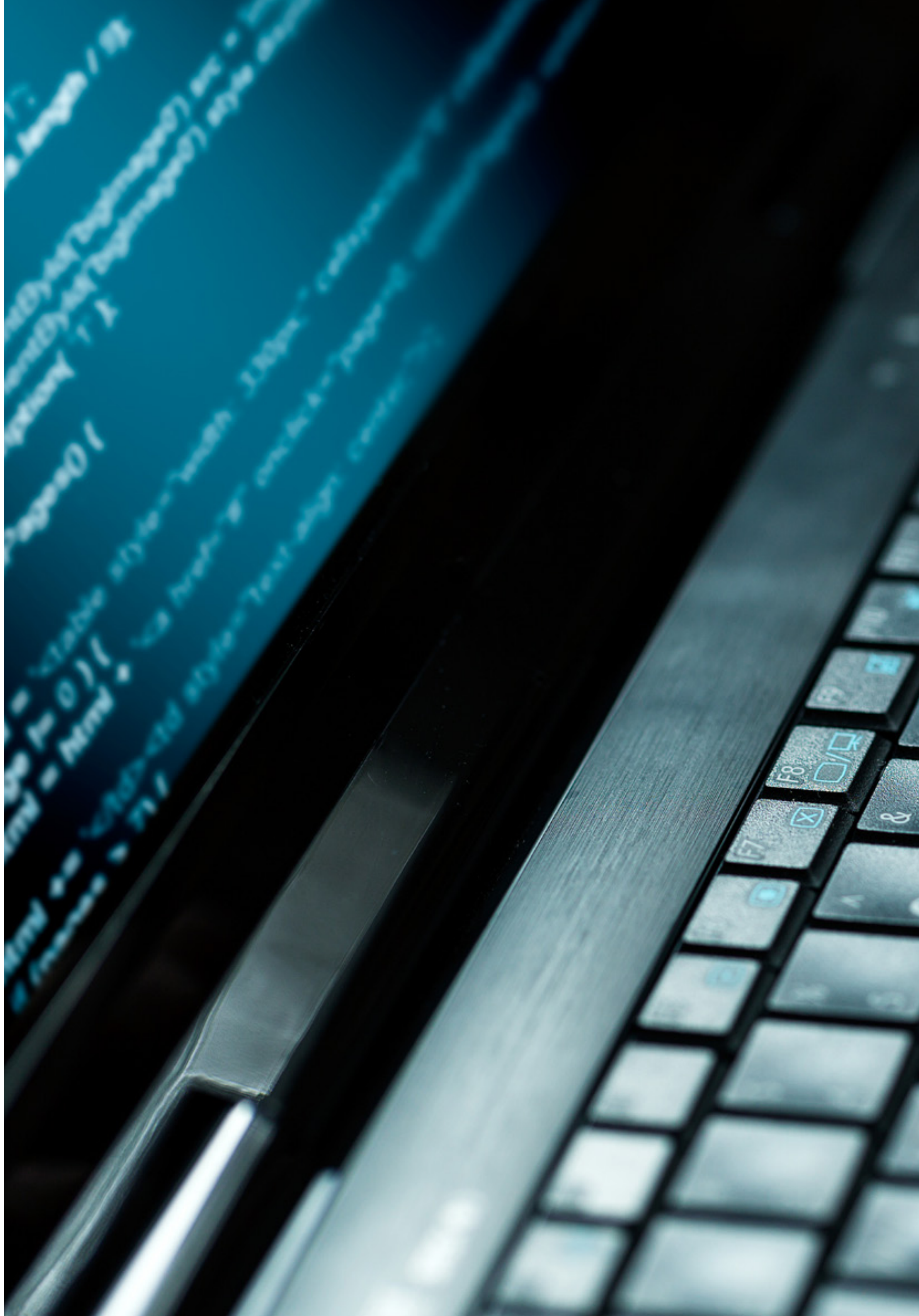
1.1. Introduction to Programming
- 1.1.1. Basic Structure of a Computer
- 1.1.2. Software
- 1.1.3. Programming Languages
- 1.1.4. Life Cycle of a Software Application

1.2. Algorithm Design
- 1.2.1. Problem Solving
- 1.2.2. Descriptive Techniques
- 1.2.3. Algorithm Elements and Structure

1.3. Elements of a Program
- 1.3.1. C++ Origin and Features
- 1.3.2. Development Environment
- 1.3.3. Concept of Program
- 1.3.4. Types of Fundamental Data
- 1.3.5. Operators
- 1.3.6. Expressions
- 1.3.7. Statements
- 1.3.8. Data Input and Output

1.4. Control Sentences
- 1.4.1. Statements
- 1.4.2. Branches
- 1.4.3. Loops

1.5. Abstraction and Modularity: Functions
- 1.5.1. Modular Design
- 1.5.2. Concept of Function and Utility
- 1.5.3. Definition of a Function
- 1.5.4. Execution Flow in a Function Call
- 1.5.5. Function Prototypes
- 1.5.6. Results Return
- 1.5.7. Calling a Function: Parameters
- 1.5.8. Passing Parameters by Reference and by Value
- 1.5.9. Scope Identifier



1.6. Static Data Structures
- 1.6.1. Arrays
- 1.6.2. Matrices. Polyhedra
- 1.6.3. Searching and Sorting
- 1.6.4. Chaining: I/O Functions for Chains
- 1.6.5. Structures. Unions
- 1.6.6. New Types of Data

1.7. Dynamic Data Structures: Pointers
- 1.7.1. Concept. Definition of Pointer
- 1.7.2. Pointer Operators and Operations
- 1.7.3. Pointer Arrays
- 1.7.4. Pointers and Arrays
- 1.7.5. Chain Pointers
- 1.7.6. Structure Pointers
- 1.7.7. Multiple Indirection
- 1.7.8. Function Pointers
- 1.7.9. Passing of Functions, Structures, and Arrays as Function Parameters

1.8. Files
- 1.8.1. Basic Concepts
- 1.8.2. File Operations
- 1.8.3. Types of Files
- 1.8.4. File Organization
- 1.8.5. Introduction to C++ Files
- 1.8.6. Managing Files

1.9. Recursion
- 1.9.1. Definition of Recursion
- 1.9.2. Types of Recursion
- 1.9.3. Advantages and Disadvantages
- 1.9.4. Considerations
- 1.9.5. Iterative Recursive Conversion
- 1.9.6. Recursion Stack

1.10. Testing and Documentation

1.10.1. Program Testing

1.10.2. White Box Testing

1.10.3. Black Box Testing

1.10.4. Testing Tools

1.10.5. Program Documentation

Module 2. Data Structures

2.1. Introduction to C++ Programming

2.1.1. Classes, Constructors, Methods and Attributes

2.1.2. Variables

2.1.3. Conditional Expressions and Loops

2.1.4. Objects

2.2. Abstract Data Types (ADT)

2.2.1. Types of Data

2.2.2. Basic Structures and TADs

2.2.3. Vectors and Arrays

2.3. Linear data Structures

2.3.1. TAD List Definition

2.3.2. Linked and Doubly Linked Lists

2.3.3. Sorted Lists

2.3.4. Lists in C++

2.3.5. TAD Stack

2.3.6. TAD Queue

2.3.7. Stack and Queue in C++

2.4. Hierarchical Data Structures

2.4.1. TAD Tree

2.4.2. Paths

2.4.3. N-Ary Trees

2.4.4. Binary Trees

2.4.5. Binary Search Trees

2.5. Hierarchical Data Structures: Complex Trees

2.5.1. Perfectly Balanced or Minimum Height Trees

2.5.2. Multipath Trees

2.5.3. Bibliographic References

2.6. Mounds and Priority Queue

2.6.1. TAD Mounds

2.6.2. TAD Priority Queue

2.7. Hash Tables

2.7.1. TAD Hash Table

2.7.2. Hash Functions

2.7.3. Hash Function in Hash Tables

2.7.4. Rehashing

2.7.5. Open Hash Tables

2.8. Graphs

2.8.1. TAD Graph

2.8.2. Graph Types

2.8.3. Graphical Representation and Basic Operations

2.8.4. Graph Design

2.9. Advanced Graph Algorithms and Concepts

2.9.1. Graph Problems

2.9.2. Path Algorithms

2.9.3. Search or Path Algorithms

2.9.4. Other Algorithms

2.10. Other Data Structures

2.10.1. Sets

2.10.2. Parallel Arrays

2.10.3. Symbol Tables

2.10.4. Tries

Module 3. Algorithmics and Complexity

3.1. Introduction to Algorithm Design Strategies

3.1.1. Recursion

3.1.2. Divide and Conquer

3.1.3. Other Strategies

3.2. Efficiency and Analysis of Algorithms

3.2.1. Efficiency Measures

3.2.2. Measuring the Size of the Input

3.2.3. Measuring Execution Time

- 3.2.4. Worst, Best and Average Case
- 3.2.5. Asymptotic Notation
- 3.2.6. Criteria for Mathematical Analysis of Non-Recursive Algorithms
- 3.2.7. Mathematical Analysis of Recursive Algorithms
- 3.2.8. Empirical Analysis of Algorithms

3.3. Sorting Algorithms

- 3.3.1. Concept of Sorting
- 3.3.2. Bubble Sorting
- 3.3.3. Sorting by Selection
- 3.3.4. Sorting by Insertion
- 3.3.5. Merge Sort
- 3.3.6. Quick Sort

3.4. Algorithms with Trees

- 3.4.1. Tree Concept
- 3.4.2. Binary Trees
- 3.4.3. Tree Paths
- 3.4.4. Representing Expressions
- 3.4.5. Ordered Binary Trees
- 3.4.6. Balanced Binary Trees

3.5. Algorithms Using Heaps

- 3.5.1. Heaps
- 3.5.2. The Heapsort Algorithm
- 3.5.3. Priority Queues

3.6. Graph Algorithms

- 3.6.1. Representation
- 3.6.2. Traversal in Width
- 3.6.3. Depth Travel
- 3.6.4. Topological Sorting

3.7. Greedy Algorithms

- 3.7.1. Greedy Strategy
- 3.7.2. Elements of the Greedy Strategy
- 3.7.3. Currency Exchange
- 3.7.4. Traveler's Problem
- 3.7.5. Backpack Problem

3.8. Minimal Path Finding

- 3.8.1. The Minimum Path Problem
- 3.8.2. Negative Arcs and Cycles
- 3.8.3. Dijkstra's Algorithm

3.9. Greedy Algorithms on Graphs

- 3.9.1. Minimum Spanning Tree
- 3.9.2. Prim's Algorithm
- 3.9.3. Kruskal's Algorithm
- 3.9.4. Complexity Analysis

3.10. Backtracking

- 3.10.1. Backtracking
- 3.10.2. Alternative Techniques

Module 4. Databases

4.1. Applications and Purposes of Database Systems

- 4.1.1. Applications of the Different Database Systems
- 4.1.2. Purpose of the Different Database Systems
- 4.1.3. View of the Data

4.2. Database and Architecture

- 4.2.1. Relational Database
- 4.2.2. Database Design
- 4.2.3. Object-Based and Semi-Structured Databases
- 4.2.4. Data Storage and Queries
- 4.2.5. Transaction Management
- 4.2.6. Data Mining and Analysis
- 4.2.7. Database Architecture

4.3. The Relational Model: Structure, Operations and Extended Relational Algebra

- 4.3.1. The Structure of Relational Databases
- 4.3.2. Fundamental Operations in the Relational Algebra
- 4.3.3. Other Relational Algebra Operations
- 4.3.4. Extended Relational Algebra Operations
- 4.3.5. Null Values
- 4.3.6. Database Modification

4.4. SQL (I)
- 4.4.1. What Is SQL?
- 4.4.2. The Definition of Data
- 4.4.3. Basic Structure of SQL Queries
- 4.4.4. Operations on Sets
- 4.4.5. Aggregation Functions
- 4.4.6. Null Values

4.5. SQL (II)
- 4.5.1. Nested Subqueries
- 4.5.2. Complex Queries
- 4.5.3. Views
- 4.5.4. Cursors
- 4.5.5. Complex Queries
- 4.5.6. Triggers

4.6. Database Design and the E-R Model
- 4.6.1. Overview of the Design Process
- 4.6.2. The Entity-Relationship Model
- 4.6.3. Constraints

4.7. Entity-Relationship Diagrams
- 4.7.1. Entity-Relationship Diagrams
- 4.7.2. Aspects of Entity-Relationship Design
- 4.7.3. Weak Entity Sets

4.8. The Extended Entity-Relationship Model
- 4.8.1. Characteristics of the Extended E-R Model
- 4.8.2. Design of a Database
- 4.8.3. Reduction to Relational Schemas

4.9. Designing from Relational Databases
- 4.9.1. Characteristics of Good Relational Designs
- 4.9.2. Atomic Domains and the First Normal Form (1FN)
- 4.9.3. Decomposition by Functional Dependencies
- 4.9.4. Theory of Functional Dependencies
- 4.9.5. Decomposition Algorithms
- 4.9.6. Decomposition by Means of Multivalued Dependencies
- 4.9.7. More Normal Forms
- 4.9.8. Database Design Process

4.10. NoSQL Databases
- 4.10.1. What are NoSQL Databases?
- 4.10.2. Analysis of the Different NoSQL Options and their Characteristics.
- 4.10.3. MongoDB

Module 5. Advanced Databases

5.1. Introduction to the Different Database Systems
- 5.1.1. Historical Overview
- 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. Advanced Algorithms Design

6.1. Analysis of Recursive and Divide and Conquer Algorithms
- 6.1.1. Posing and Solving Homogeneous and Non-Homogeneous Recurrence Equations
- 6.1.2. General Description of the Divide and Conquer Strategy

6.2. Amortized Analysis
- 6.2.1. Aggregate Analysis
- 6.2.2. The Accounting Method
- 6.2.3. The Potential Method

6.3. Dynamic Programming and Algorithms for NP Problems
- 6.3.1. Characteristics of Dynamic Programming
- 6.3.2. Backtracking Backtracking
- 6.3.3. Branch and Bound

6.4. Combinatorial Optimization
- 6.4.1. Representation
- 6.4.2. 1D Optimization

6.5. Randomization Algorithms
- 6.5.1. Examples of Randomization Algorithms
- 6.5.2. The Buffon Theorem
- 6.5.3. Monte Carlo Algorithm
- 6.5.4. Las Vegas Algorithm

6.6. Local Search and Candidate Search
- 6.6.1. Gradient Ascent
- 6.6.2. Hill Climbing
- 6.6.3. Simulated Annealing
- 6.6.4. Tabu Search
- 6.6.5. Candidate Search

6.7. Formal Verification of Programs
- 6.7.1. Specification of Functional Abstractions
- 6.7.2. The Language of First-Order Logic
- 6.7.3. Hoare's Formal System

6.8. Verification of Iterative Programs
- 6.8.1. Rules of Hoare's Formal System
- 6.8.2. Concept of Invariant Iterations

6.9. Numeric Methods
6.9.1. The Bisection Method
6.9.2. Newton Raphson's Method
6.9.3. The Secant Method

6.10. Parallel Algorithms
6.10.1. Parallel Binary Operations
6.10.2. Parallel Operations with Networks
6.10.3. Parallelism in Divide and Conquer
6.10.4. Parallelism in Dynamic Programming

Module 7. Human-Computer Interaction

7.1. Introduction to Human-Computer Interaction
7.1.1. What is Human-Computer Interaction
7.1.2. Relationship of Human-Computer Interaction with Other Disciplines
7.1.3. The User Interface
7.1.4. Usability and Accessibility
7.1.5. User Experience and User-Centered Design

7.2. The Computer and Interaction: User Interface and Interaction Paradigms
7.2.1. Interaction
7.2.2. Paradigms and Styles of Interaction
7.2.3. Evolution of User Interfaces
7.2.4. Classic User Interfaces: WIMP/GUI, Commands, Voice, Virtual Reality.
7.2.5. Innovative User Interfaces: Mobile, Wearable, Collaborative, BCI

7.3. The Human Factor: Psychological and Cognitive Aspects
7.3.1. The Importance of the Human Factor in Interaction
7.3.2. Human Information Processing
7.3.3. The Input and Output of Information: Visual, Auditory, and Tactile
7.3.4. Perception and Attention
7.3.5. Knowledge and Mental Models: Representation, Organization, and Acquisition

7.4. The Human Factor: Sensory and Physical Limitations
7.4.1. Functional Diversity, Disability and Impairment
7.4.2. Visual Diversity
7.4.3. Hearing Diversity
7.4.4. Cognitive Diversity
7.4.5. Motor Diversity
7.4.6. The Case of Digital Immigrants

7.5. The Design Process (I): Requirements Analysis for User Interface Design
7.5.1. User-Centered Design
7.5.2. What is Requirements Analysis?
7.5.3. Information Gathering
7.5.4. Analysis and Interpretation of the Information
7.5.5. Usability and Accessibility Analysis

7.6. The Design Process (II): Prototyping and Task Analysis
7.6.1. Conceptual Design
7.6.2. Prototyping
7.6.3. Hierarchical Task Analysis

7.7. The Design Process (III): Evaluation
7.7.1. Evaluation in the Design Process: Objectives and Methods
7.7.2. Evaluation Methods Without Users
7.7.3. Evaluation Methods with Users
7.7.4. Evaluation Standards and Norms

7.8. Accessibility: Definition and Guidelines
7.8.1. Accessibility and Universal Design
7.8.2. The WAI Initiative and the WCAG Guidelines
7.8.3. WCAG 2.0 and 2.1 Guidelines

7.9. Accessibility: Evaluation and Functional Diversity
7.9.1. Web Accessibility Evaluation Tools
7.9.2. Accessibility and Functional Diversity

7.10. The Computer and Interaction: Peripherals and Devices
7.10.1. Traditional Devices and Peripherals
7.10.2. Alternative Devices and Peripherals
7.10.3. Cell Phones and Tablets
7.10.4. Functional Diversity, Interaction and Peripherals

Module 8. Advanced Programming

8.1. Introduction to Object-Oriented Programming
8.1.1. Introduction to Object-Oriented Programming
8.1.2. Class Design
8.1.3. Introduction to UML for Problem Modeling

8.2. Relationships Between Classes
- 8.2.1. Abstraction and Inheritance
- 8.2.2. Advanced Inheritance Concepts
- 8.2.3. Polymorphism
- 8.2.4. Composition and Aggregation

8.3. Introduction to Design Patterns for Object-Oriented Problems
- 8.3.1. What Are Design Patterns?
- 8.3.2. Factory Pattern
- 8.3.3. Singleton Pattern
- 8.3.4. Observer Pattern
- 8.3.5. Composite Pattern

8.4. Exceptions
- 8.4.1. What Are Exceptions?
- 8.4.2. Exception Catching and Handling
- 8.4.3. Throwing Exceptions
- 8.4.4. Exception Creation

8.5. User Interfaces
- 8.5.1. Introduction to Qt
- 8.5.2. Positioning
- 8.5.3. What Are Events?
- 8.5.4. Events: Definition and Catching
- 8.5.5. User Interface Development

8.6. Introduction to Concurrent Programming
- 8.6.1. Introduction to Concurrent Programming
- 8.6.2. The Concept of Process and Thread
- 8.6.3. Interaction Between Processes or Threads
- 8.6.4. Threads in C++
- 8.6.5. Advantages and Disadvantages of Concurrent Programming

8.7. Thread Management and Synchronization
- 8.7.1. Life Cycle of a Thread
- 8.7.2. Thread Class
- 8.7.3. Thread Planning
- 8.7.4. Thread Groups
- 8.7.5. Daemon Threads
- 8.7.6. Synchronization
- 8.7.7. Locking Mechanisms
- 8.7.8. Communication Mechanisms
- 8.7.9. Monitors

8.8. Common Problems in Concurrent Programming
- 8.8.1. The Problem of Consuming Producers
- 8.8.2. The Problem of Readers and Writers
- 8.8.3. The Problem of the Philosophers' Dinner Party

8.9. Software Documentation and Testing
- 8.9.1. Why is it Important to Document Software?
- 8.9.2. Design Documentation
- 8.9.3. Documentation Tool Use

8.10. Software Testing
- 8.10.1. Introduction to Software Testing
- 8.10.2. Types of Tests
- 8.10.3. Unit Test
- 8.10.4. Integration Test
- 8.10.5. Validation Test
- 8.10.6. System Test

Module 9. Network Application Development

9.1. HTML5 Markup Languages
- 9.1.1. HTML Basics
- 9.1.2. New HTML 5 Elements
- 9.1.3. Forms: New Controls

9.2. Introduction to CSS Style Sheets
- 9.2.1. First Steps with CSS
- 9.2.2. Introduction to CSS3

9.3. Browser Scripting Language: JavaScript
- 9.3.1. JavaScript Basics
- 9.3.2. DOM
- 9.3.3. Events
- 9.3.4. JQuery
- 9.3.5. Ajax

- 9.4. Concept of Component-Oriented Programming
    - 9.4.1. Context
    - 9.4.2. Components and Interfaces
    - 9.4.3. States of a Component
- 9.5. Component Architecture
    - 9.5.1. Current Architectures
    - 9.5.2. Component Integration and Deployment
- 9.6. Frontend Framework: Bootstrap
    - 9.6.1. Grid Design
    - 9.6.2. Forms
    - 9.6.3. Components
- 9.7. Model View Controller
    - 9.7.1. Web Development Methods
    - 9.7.2. Design Pattern: MVC
- 9.8. Information Grid Technologies
    - 9.8.1. Increased Computing Resources
    - 9.8.2. Concept of Grid Technology
- 9.9. Service-Oriented Architecture
    - 9.9.1. SOA and Web Services
    - 9.9.2. Topology of a Web Service
    - 9.9.3. Platforms for Web Services
- 9.10. HTTP Protocol
    - 9.10.1. Messages
    - 9.10.2. Persistent Sessions
    - 9.10.3. Cryptographic System
    - 9.10.4. HTTPS Protocol Operation

Module 10. Software Engineering

- 10.1. Introduction to Software Engineering and Modeling
    - 10.1.1. The Nature of Software
    - 10.1.2. The Unique Nature of WebApps
    - 10.1.3. Software Engineering
    - 10.1.4. The Software Process
    - 10.1.5. Software Engineering Practice
    - 10.1.6. Software Myths
    - 10.1.7. How It All Begins
    - 10.1.8. Object-Oriented Concepts
    - 10.1.9. Introduction to UML
- 10.2. The Software Process
    - 10.2.1. A General Process Model
    - 10.2.2. Prescriptive Process Models
    - 10.2.3. Specialized Process Models
    - 10.2.4. The Unified Process
    - 10.2.5. Personal and Team Process Models
    - 10.2.6. What Is Agility?
    - 10.2.7. What Is an Agile Process?
    - 10.2.8. Scrum
    - 10.2.9. Agile Process Toolkit
- 10.3. Principles Guiding Software Engineering Practice
    - 10.3.1. Principles Guiding the Process
    - 10.3.2. Principles Guiding the Practice
    - 10.3.3. Principles of Communication
    - 10.3.4. Planning Principles
    - 10.3.5. Modeling Principles
    - 10.3.6. Construction Principles
    - 10.3.7. Deployment Principles
- 10.4. Understanding the Requirements
    - 10.4.1. Requirements Engineering
    - 10.4.2. Establish the Basis
    - 10.4.3. Inquiry of Requirements
    - 10.4.4. Development of Cases Studies
    - 10.4.5. Elaboration of the Requirements Model
    - 10.4.6. Negotiation of Requirements
    - 10.4.7. Validation of Requirements
- 10.5. Requirements Modeling: Scenarios, Information and Analysis Classes
    - 10.5.1. Analysis of Requirements
    - 10.5.2. Scenario-Based Modeling

10.5.3. UML Models that Provide the Case Study

10.5.4. Data Modeling Concepts

10.5.5. Class-Based Modeling

10.5.6. Class Diagrams

10.6. Requirements Modeling: Flow, Behavior and Patterns

10.6.1. Requirements that Shape Strategies

10.6.2. Flow-Oriented Modeling

10.6.3. Status Diagrams

10.6.4. Creation of a Behavioral Model

10.6.5. Sequence Diagrams

10.6.6. Communication Diagrams

10.6.7. Patterns for Requirements Modeling

10.7. Design Concepts

10.7.1. Design in the Software Engineering Context

10.7.2. The Design Process

10.7.3. Design Concepts

10.7.4. Object-Oriented Design Concepts

10.7.5. Model of the Design

10.8. Designing the Architecture

10.8.1. Software Architecture

10.8.2. Architectural Genres

10.8.3. Architectural Styles

10.8.4. Architectural Design

10.8.5. Evolution of Alternative Designs for Architecture

10.8.6. Mapping the Architecture Using the Data Flow

10.9. Component-Level and Pattern-Based Design

10.9.1. What Is a Component?

10.9.2. Class-Based Component Design

10.9.3. Realization of the Design at the Component Level

10.9.4. Design of Traditional Components

10.9.5. Component-Based Development

10.9.6. Design Patterns

10.9.7. Pattern-Based Software Design

10.9.8. Architectural Patterns

10.9.9. Design Patterns at the Component Level

10.9.10. User Interface Design Patterns

10.10. Software Quality and Project Management

10.10.1. Software Quality

10.10.2. The Software Quality Dilemma

10.10.3. Achieving Software Quality

10.10.4. Software Quality Assurance

10.10.5. The Administrative Spectrum

10.10.6. The Personnel

10.10.7. Product Strategy

10.10.8. The Process

10.10.9. The Project

10.10.10. Principles and Practices

> *You will apply quality, performance, and sustainability criteria throughout the entire software life cycle”*

04

Teaching Objectives

The teaching objectives of this program respond to the demands of a constantly evolving digital ecosystem. The aim is not only to develop advanced technical competencies, but also to foster critical thinking, agile problem-solving, and the capacity for innovation. Through a multidisciplinary and up-to-date approach, the program seeks to train professionals capable of leading technological projects, adapting to new tools, and efficiently applying emerging methodologies. In addition, it promotes a deep understanding of the complete development life cycle, integrating key aspects such as security, scalability, and software sustainability in highly competitive and dynamic environments.

```
a";
($_GET[type]==1
"foto-galerija.php?t
<div id="left_sidebar">
    <div id="left_ico">
    <p <?if($_COOKIE['la

<?
if($_COOKIE['lang'] =='eng'){
    echo "Wood-frame houses";
```

> *Specialize in the design of user-centered digital experiences, applying human–computer interaction and accessibility techniques”*

General Objectives

- Develop advanced technical competencies in programming, design, software architecture, and related methodologies
- Apply agile methodologies and collaborative development approaches in technological projects of varying complexity
- Integrate automation tools, version control, and testing in modern professional development environments
- Analyze and design scalable, secure, efficient, and sustainable solutions across diverse digital contexts and sectors
- Master multiplatform development environments and emerging technologies applied to the design of digital solutions
- Understand the complete software life cycle and its impact on current business processes
- Evaluate the technical and economic feasibility of technology development projects in different professional environments
- Enhance the capacity for innovation applied to the design of complex digital products, services, and systems

Specific Objectives

Module 1. Programming Fundamentals

- Understand the basic principles of structured and object-oriented programming
- Understand the basic principles of structured and object-oriented programming
- Develop simple algorithms using good coding practices
- Interpret common errors and debug programs efficiently

Module 2. Data Structures

- Analyze the usefulness of data structures in solving computational problems
- Implement lists, stacks, queues, trees, and graphs in specific contexts
- Evaluate the efficiency of different structures according to the required use
- Apply appropriate data structures to algorithm design

Module 3. Algorithmics and Complexity

- Design efficient algorithms considering their computational cost
- Compare algorithm performance using Big O notation
- Apply analysis techniques to identify bottlenecks
- Optimize algorithms to improve their behavior with large volumes of data

Module 4. Databases

- Design relational models applying normalization principles
- Implement databases using conventional database management systems
- Develop SQL queries to retrieve, insert, and modify information
- Understand the fundamentals of integrity, transactions, and concurrency

Module 5. Advanced Databases

- Apply query optimization techniques in complex environments
- Manage distributed and non-relational databases
- Use stored procedures, triggers, and views in real-world scenarios
- Analyze database solutions oriented toward Big Data

Module 6. Advanced Algorithms Design

- Develop techniques such as divide and conquer, dynamic programming, and backtracking
- Solve complex problems using classical and heuristic algorithms
- Identify NP-complete problems and evaluate possible solutions
- Implement efficient algorithms adapted to specific needs

Module 7. Human-Computer Interaction

- Design interfaces centered on user experience
- Apply usability and accessibility principles in interactive systems
- Evaluate prototypes through user testing and interaction metrics
- Integrate interface design patterns into real applications

Module 8. Advanced Programming

- Implement complex applications using modern programming techniques
- Apply concepts such as threads, dynamic memory, and controlled exceptions
- Develop modular, reusable, and scalable software
- Integrate external libraries and tools into development projects

Module 9. Network Application Development

- Design and implement client–server applications using standard protocols
- Manage connections, concurrency, and security in distributed environments

- Apply serialization and remote process communication techniques
- Use tools for the development and testing of network services

Module 10. Software Engineering

- Apply development methodologies such as Scrum, XP, and DevOps
- Manage the software life cycle from planning to maintenance
- Develop clear and well-structured technical documentation for software projects
- Coordinate teams and tools for the collaborative development of complex systems

> *You will stand out for your commitment to quality, code security, and the ethical development of software"*

05
Career Opportunities

Software Development has become one of the pillars of the digital economy, driving demand for specialized profiles across multiple sectors. Accordingly, this program opens the door to professional opportunities in areas such as application engineering, systems architecture, cybersecurity, and data analysis. In addition, due to constant technological evolution and global digital transformation, the competencies acquired enable access to strategic roles in technology companies, startups, and international institutions. It is therefore a field with high employability, international projection, and a real capacity to impact business innovation.

> *Advance your professional profile toward strategic roles such as DevOps Engineer, Software Architect, or Systems Analyst”*

Graduate Profile

The graduate profile of this university program at TECH is characterized by a strong combination of technical skills, analytical thinking, and the ability to adapt to changing technological environments. With a comprehensive vision of Software Development, graduates are prepared to design, implement, and optimize efficient, scalable, and secure digital solutions. In addition, they master current tools and languages, work with agile methodologies, and understand the principles of modern software engineering. This versatility enables them to successfully integrate into multidisciplinary teams, lead innovative projects, and respond effectively to the challenges of a highly competitive global market.

Would you like to work as a Software Architect? This curriculum will provide you with the keys to achieve it in just a few months

- **Logical and Structured Thinking:** Key for the design and optimization of complex algorithms
- **Capacity for Abstraction:** Ability to model technological solutions based on real-world problems
- **Applied Innovation:** Competency focused on the creation of digital products with differentiated value
- **Digital Responsibility and Professional Ethics:** Skill in the use and development of technologies with social impact

After completing the university program, you will be able to apply your knowledge and skills in the following positions:

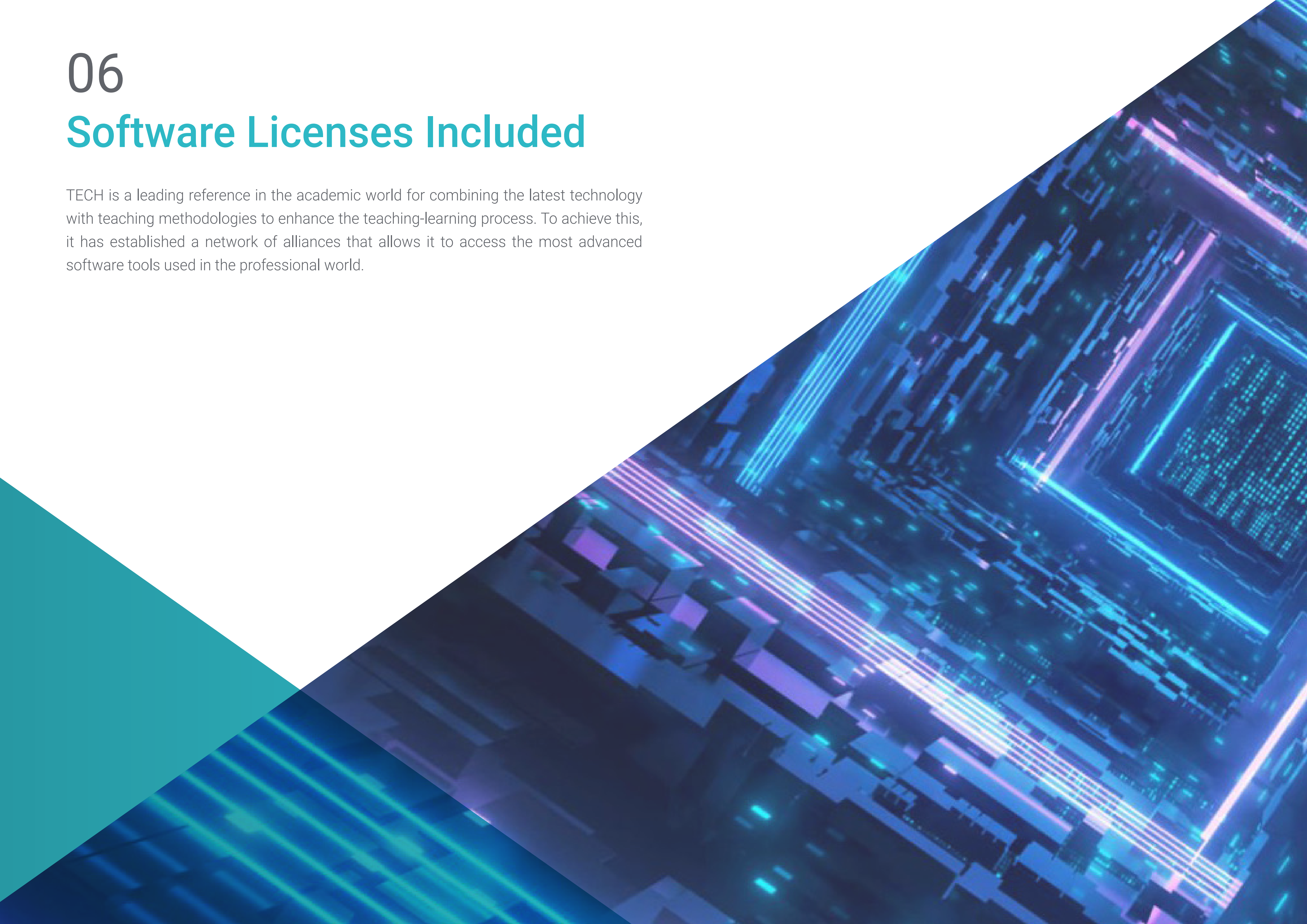
1. ***Software* Developer:** Designs, codes, and tests applications tailored to the needs of companies and users across different technological environments
2. **Software Engineer:** Plans, builds, and maintains complex systems, applying principles of quality, scalability, and security
3. **Software Architect:** Defines the technical structure of applications and coordinates key decisions to ensure efficiency and sustainability
4. **Systems Analyst:** Evaluates requirements, designs digital solutions, and acts as a liaison between technical teams and business objectives
5. **DevOps Engineer:** Automates development and continuous integration processes, optimizing the software life cycle
6. Back-End Developer: Specializes in the internal logic of systems, managing databases, servers, and services
7. Full-Stack Developer: Masters both front-end and back-end development, participating in all phases of a digital project
8. **QA and Testing Specialist:**
   Ensures software quality through functional, automated, and performance testing

> *You will create automated tests for software validation in integration environments, ensuring optimal delivery"*

06

Software Licenses Included

TECH is a leading reference in the academic world for combining the latest technology with teaching methodologies to enhance the teaching-learning process. To achieve this, it has established a network of alliances that allows it to access the most advanced software tools used in the professional world.



Upon enrolling, you will receive, completely free of charge, academic credentials for the following professional software applications"

TECH has established a network of professional alliances with the leading providers of software applied to various professional fields. These alliances allow TECH to access hundreds of software applications and licenses, making them available to its students.

The software licenses for academic use will allow students to utilize the most advanced applications in their professional field, enabling them to become familiar with and master these tools without incurring any costs. TECH will handle the licensing process so that students may use these resources without limitation throughout the duration of the Master's Degree in Software Development, completely free of charge.

TECH will provide free access to the following software applications:

Google Career Launchpad

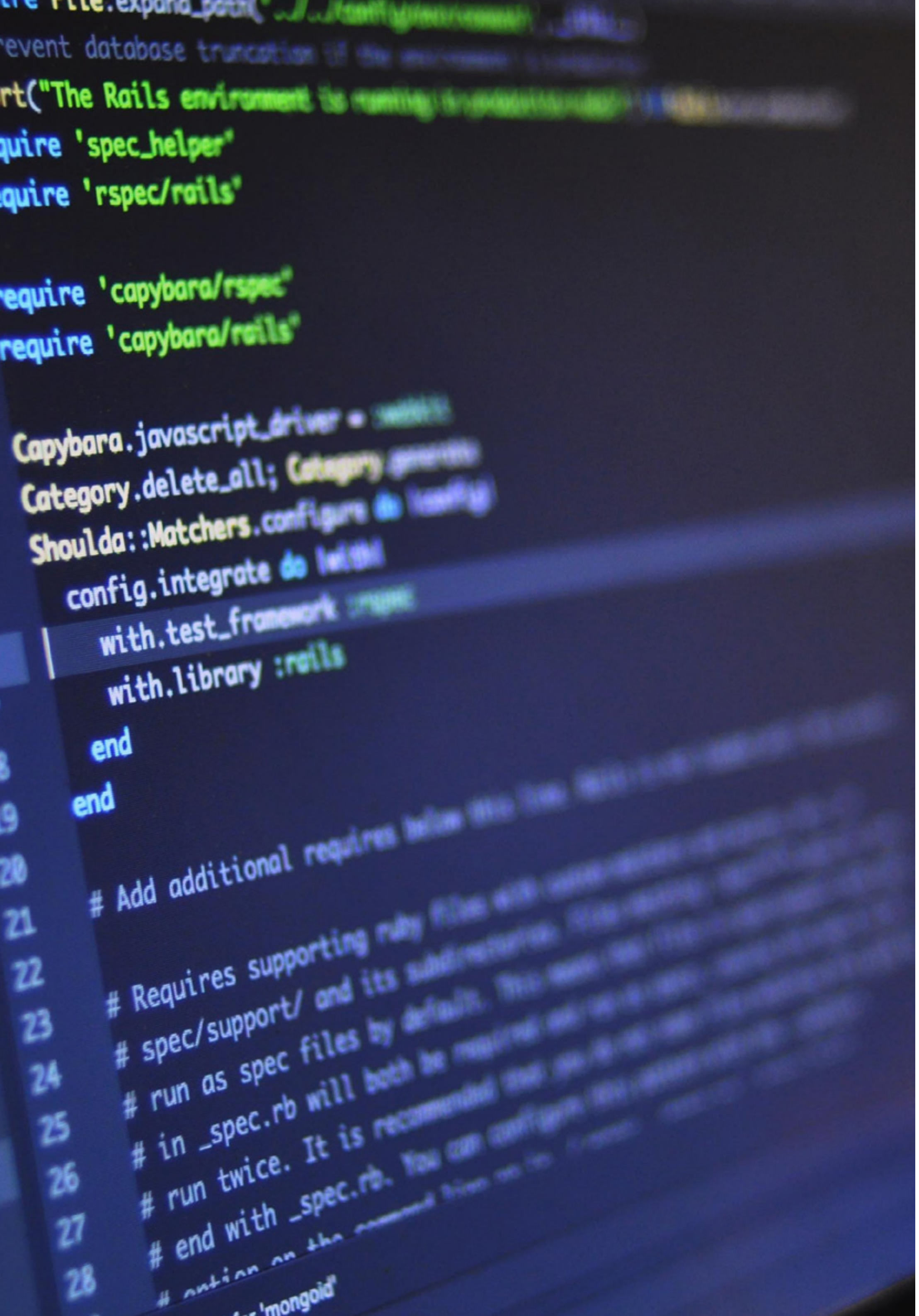
Google Career Launchpad is a solution for developing digital skills in technology and data analysis. With an estimated value of **5,000 dollars**, it is included **for free** in TECH's university program, providing access to interactive labs and certifications recognized in the industry.

This platform combines technical training with practical cases, using technologies such as BigQuery and Google AI. It offers simulated environments to work with real data, along with a network of experts for personalized guidance.

Key Features:

- **Specialized Courses:** Updated content in cloud computing, machine learning, and data analysis
- **Live Labs:** Hands-on practice with real Google Cloud tools, no additional configuration required
- **Integrated Certifications:** Preparation for official exams with international validity
- **Professional Mentoring:** Sessions with Google experts and technology partners
- **Collaborative Projects:** Challenges based on real-world problems from leading companies

In conclusion, **Google Career Launchpad** connects users with the latest market technologies, facilitating their entry into fields such as artificial intelligence and data science with industry-backed credentials.



```
event database truncation
rt("The Rails environment is
quire 'spec_helper'
quire 'rspec/rails'

equire 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver =
Category.delete_all;
Shoulda::Matchers.configure do
  config.integrate do
    with.test_framework
    with.library :rails
  end
end

# Add additional requires
# Requires supporting
# spec/support/ and its
# run as spec files by
# in _spec.rb will
# run twice. It is
# end with _spec.rb.
```

DBeaver Enterprise Edition

DBeaver Enterprise Edition is the professional version of the renowned database management tool DBeaver, with a commercial price of approximately **250 euros** annually. It is offered for free during the university program at TECH, enabling graduates to professionally and securely manage, develop, and analyze data in complex environments.

This platform equips TECH graduates to optimize the management of relational and non-relational databases, generate smart SQL queries, design advanced schemas, and visualize information with interactive charts. Additionally, it integrates business analysis functions by connecting with *Business Intelligence* tools, transforming data into strategic knowledge for decision-making.

Key Features:

- **Wide Compatibility:** Supports Oracle, SQL Server, PostgreSQL, MongoDB, Cassandra, and more
- **Advanced SQL Editor:** Autocompletion, debugging, and smart assistants
- **Data Visualization:** Interactive dashboards and integrated charts
- **Integration with Tableau:** Direct connection to Business Intelligence tools
- **Schema Design:** ERD editing and reverse engineering
- **Full Administration:** Backup, restoration, comparison, and user management

In conclusion, **DBeaver Enterprise Edition** empowers TECH graduates to master data management with precision, efficiency, and innovation.

07

Study Methodology

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

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

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

The student: the priority of all TECH programs

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

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

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

The most comprehensive academic programs worldwide

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

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

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

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

Case Studies or Case Method

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

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

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

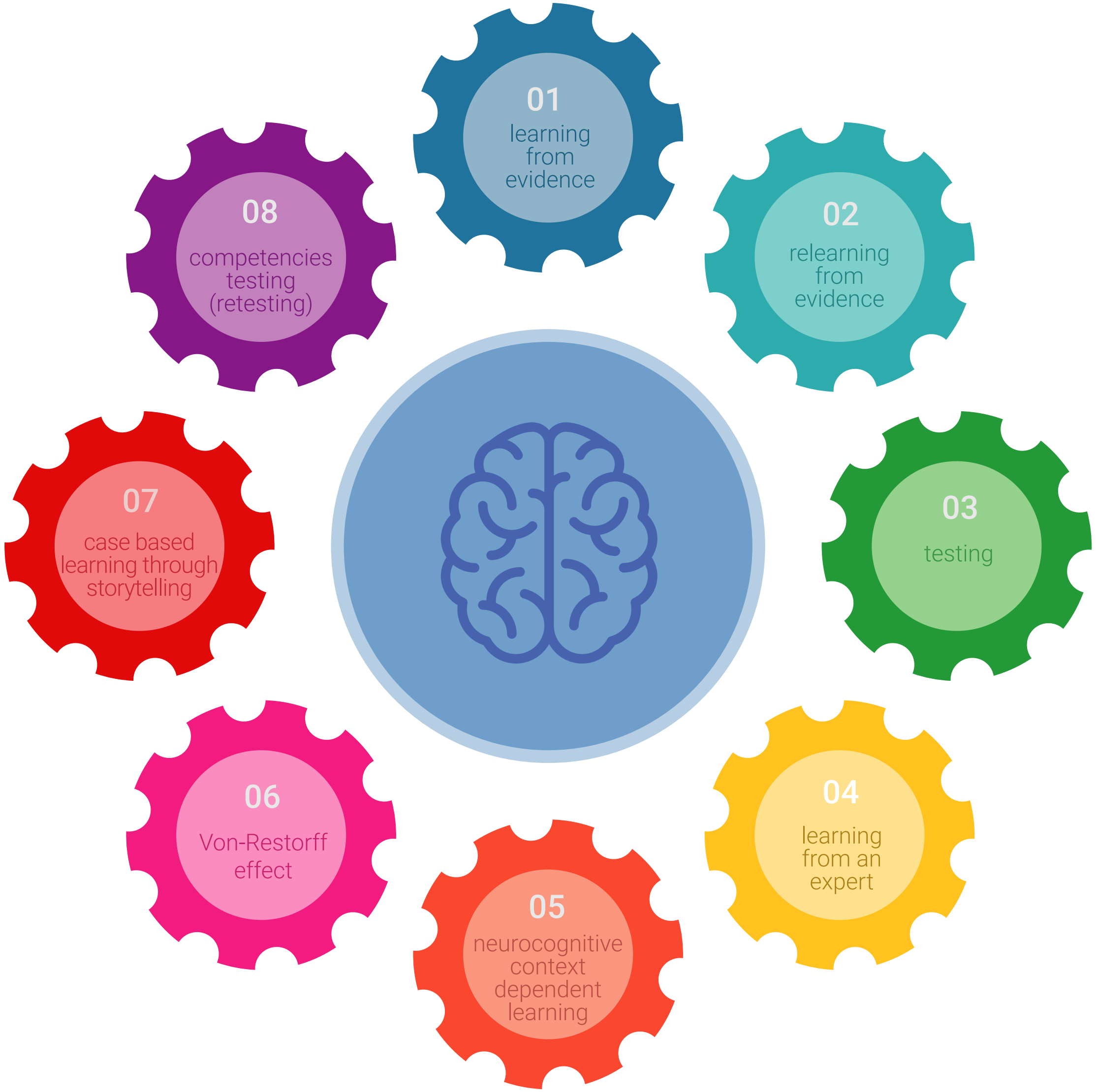
Relearning Method

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

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

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

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


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

A 100% online Virtual Campus with the best teaching resources

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

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

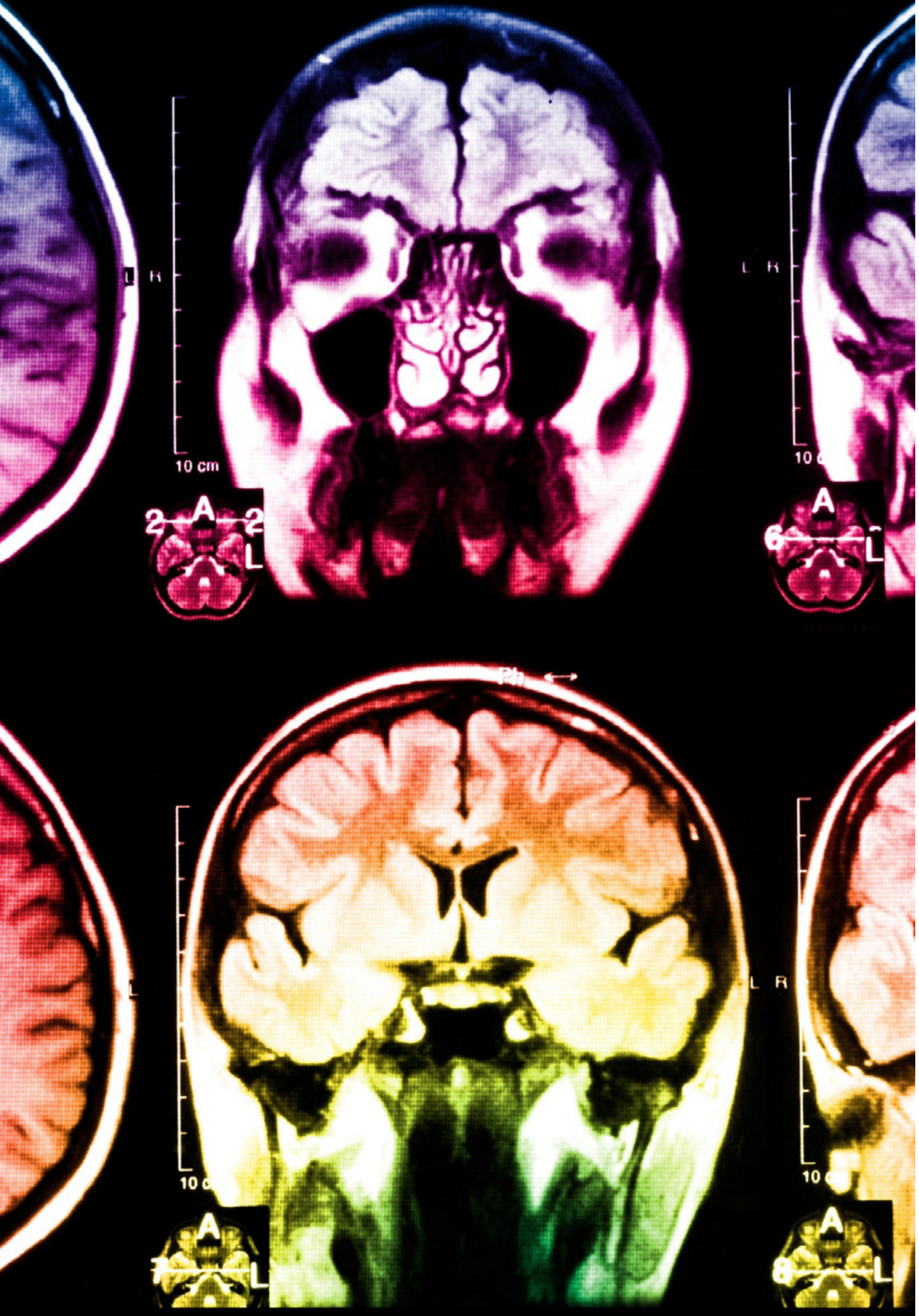
In order to facilitate tutor-student contact as much as possible, you will have a wide range of communication possibilities, both in real time and delayed (internal messaging, telephone answering service, email contact with the technical secretary, chat and videoconferences).

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

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

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

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



The university methodology best rated by its students

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

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

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

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

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

Study Material

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

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

Practicing Skills and Abilities

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

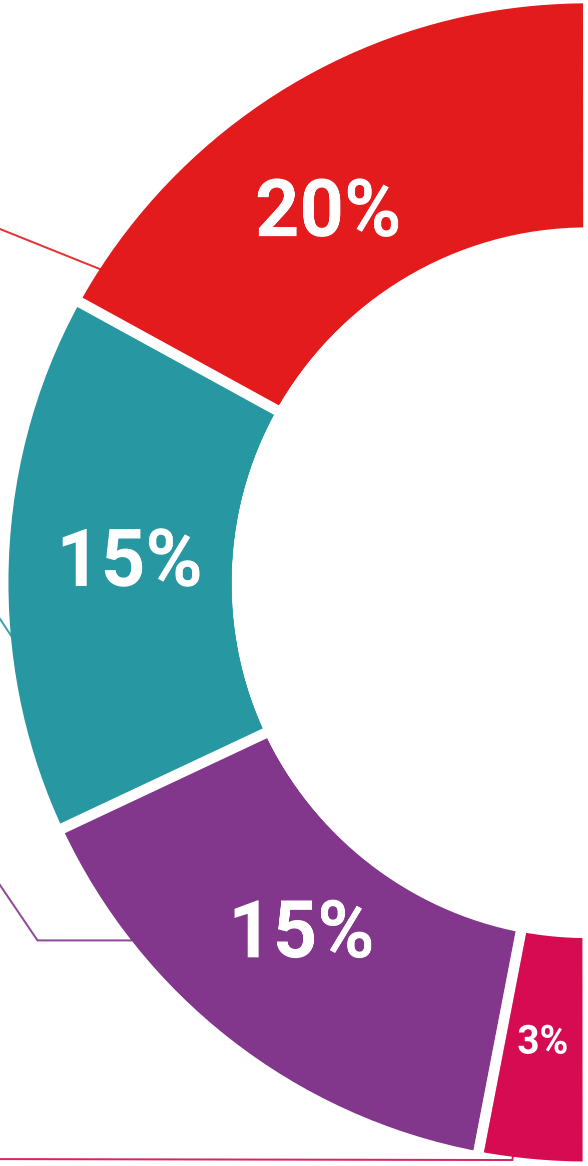
Interactive Summaries

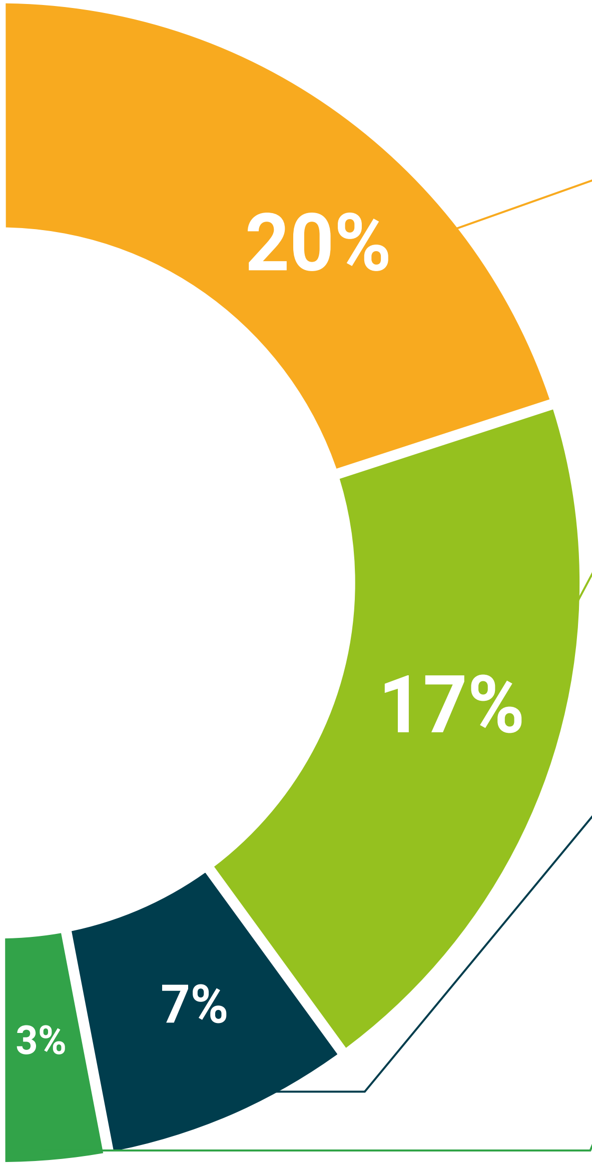
We present the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

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

Additional Reading

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





Case Studies

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

Testing & Retesting

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

Classes

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

Learning from an expert strengthens knowledge and memory, and generates confidence for future difficult decisions.

Quick Action Guides

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

08

Teaching Staff

The faculty of this program is composed of active professionals with direct experience in leading companies within the technology sector. This combination of academic knowledge and real-world practice enables the delivery of up-to-date content aligned with current market demands. In addition, the diversity of specialties within the teaching team ensures a multidisciplinary approach, encompassing areas ranging from software engineering to innovation in distributed systems. This fosters a comprehensive understanding of the digital environment and facilitates the acquisition of competencies that respond to current challenges and emerging needs within the global industry.

> *You will have access to a curriculum designed by a renowned and highly qualified faculty specializing in Software Development, ensuring an efficient and high-quality learning experience"*

International Guest Director

Considered as an international reference in the application of digital strategies, Amey Patil is a prestigious **Software Engineer** specialized in the use of state-of-the-art technological tools.

In this sense, he has worked in recognized institutions such as **Google** in California. In this way, he has been in charge of managing multiple digital infrastructures among which **Ads Creatlve Studio, Display & Video 360** or **Google Analytics** stand out. Under his leadership, these products have generated significant revenues that have driven the **strategic growth** of the company, while improving the **user experience**.

In this way, his work philosophy is based on designing and implementing **customized online solutions** for institutions in order to optimize both their online presence and their advertising performance. Thanks to this, he has managed numerous **marketing campaigns** that have maximized the **return on investment**; at the same time they have also strengthened the reputation of organizations in society.

On the other hand, he has been a driving force in various initiatives that promote **innovation** in the technological field. An example of this is the analysis of how **Artificial Intelligence** enables companies to automate complex tasks such as the processing of large volumes of information. With this, it has helped a wide range of companies to improve their operational processes and make informed decisions. It has also enabled businesses to identify **trends** early on to create unique assets that have positioned themselves in the market.

In his commitment to excellence, Amey Patil participates as a speaker at international **scientific conferences**. Thereby, he shares his comprehensive knowledge on subjects such as the adoption of **Agile Methodologies**, systems to ensure **Quality Assurance in Computer Applications** and the latest advances in the management of **Databases** in corporate environments.

Mr. Patil, Amey

- Director of Software Engineering at Google in California, United States
- Director of Engineering at Dell Technologies, California
- Software Developer at Mealpit, California
- Quality Assurance Engineer at Amdocs, Cyprus
- Google Ads and Google Analytics Specialist
- Master of Science in Engineering Science from the University of Maryland
- Bachelor of Science in Telecommunications from University of Mumbai

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

09

Certificate

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

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

This private qualification will allow you to obtain a diploma for the **Master's Degree in Software Development** endorsed by **TECH Global University**, the world's largest online university.

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

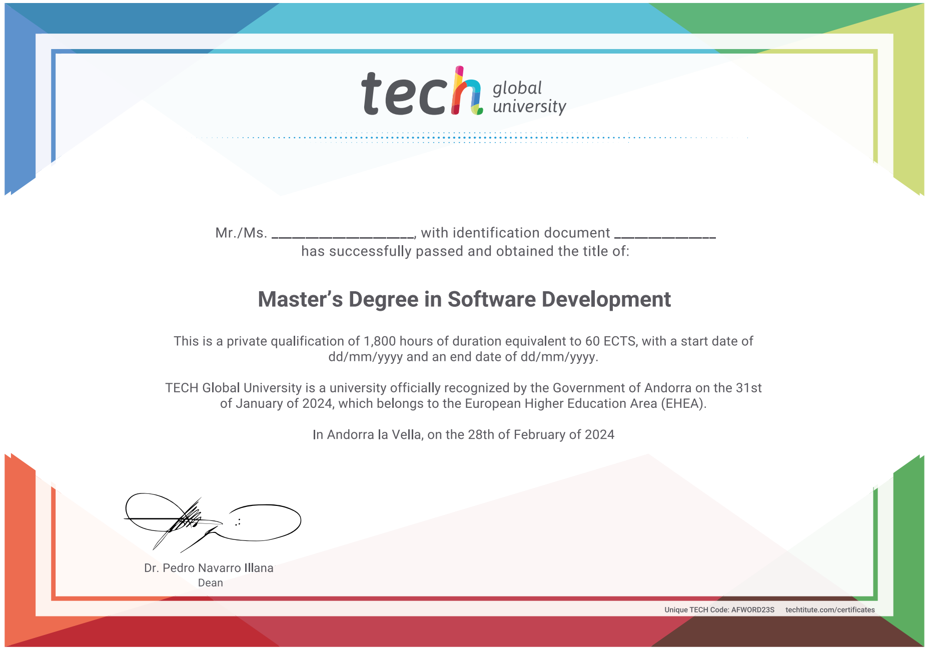
This private qualification from **TECH Global University** is a European continuing education and professional development program that guarantees the acquisition of competencies in its area of expertise, providing significant curricular value to the student who successfully completes the program.

Title: **Master's Degree in Software Development**

Modality: **online**

Duration: **12 months.**

Accreditation: **60 ECTS**



tech global university

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

Master's Degree in Software Development

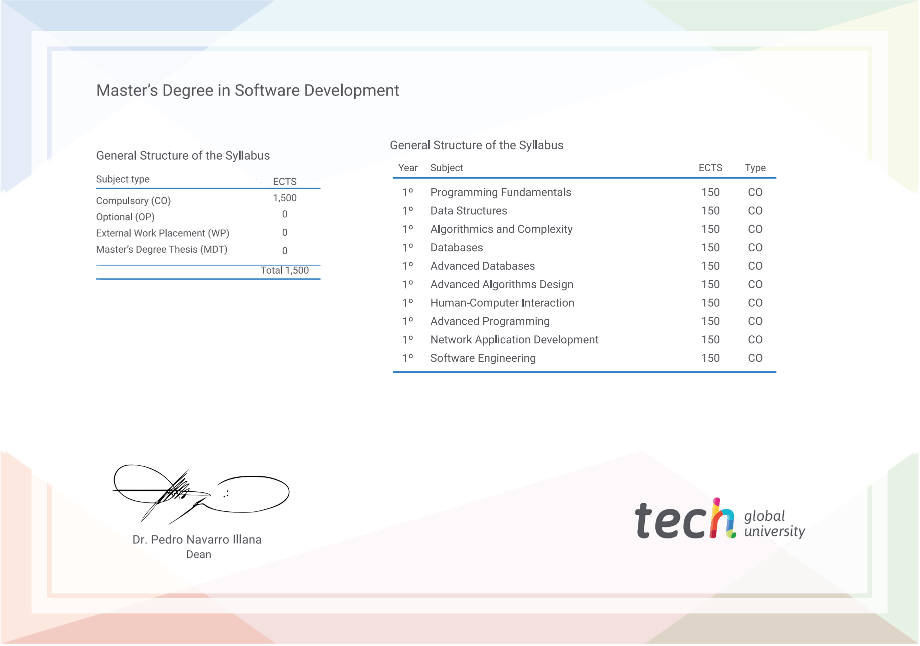
This is a private qualification of 1,800 hours of duration equivalent to 60 ECTS, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

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

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

Dr. Pedro Navarro Illana
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates



Master's Degree in Software Development

General Structure of the Syllabus

Subject type	ECTS
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	ECTS	Type
1º	Programming Fundamentals	150	CO
1º	Data Structures	150	CO
1º	Algorithmics and Complexity	150	CO
1º	Databases	150	CO
1º	Advanced Databases	150	CO
1º	Advanced Algorithms Design	150	CO
1º	Human-Computer Interaction	150	CO
1º	Advanced Programming	150	CO
1º	Network Application Development	150	CO
1º	Software Engineering	150	CO

Dr. Pedro Navarro Illana
Dean

tech global university

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

tech global university

Master's Degree
Software Development

- Modality: **online**
- Duration: **12 months.**
- Certificate: **TECH Global University**
- Accreditation: **60 ECTS**
- Schedule: **at your own pace**
- Exams: **online**

Master's Degree
Software Development

tech global university