Master's Degree
Programming from Scratch

tech global university

tech global university

Master's Degree
Programming from Scratch

- 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-programming-scratch

Index

01

Introduction to the Program

Programming has become an essential skill in the 21st century, playing a key role in the digital transformation of societies and economies. According to an OECD report, more than 60% of current jobs require advanced digital skills, with programming being one of the most in-demand. Taking into account that it is an essential field in the digital age, TECH has designed this Master's Degree that will address the fundamentals of this discipline and its application in languages for global use. Through a 100% online methodology, specialists will discover how to structure algorithms, manage data and develop functional projects, while building a solid foundation for further advancement in the technological world.

Knowing how to program has never been so accessible. With TECH's Master's Degree you will master languages such as Python and JavaScript from anywhere, thanks to the 100% online methodology. Enroll now!"

Programming is at the heart of the digital transformation that modern societies are experiencing. From the development of mobile applications to the automation of business processes, knowing how to program is a fundamental skill for making one's way in a constantly evolving work environment. In this sense, professionals who master this field will not only find better opportunities for growth, but will also enhance their ability to adapt and prosper in a world where technology is constantly evolving.

Aware of this need, TECH has designed the Master's Degree in Programming from Scratch, a comprehensive program that will provide the most up-to-date knowledge related to this field. The syllabus, designed by experts from the industry, will cover everything from the basics of algorithms and data structures to the practical use of languages such as Python, JavaScript and HTML. It will also include specialized modules in web design, application development and database management. In this way, specialists will acquire technical knowledge and develop analytical and creative skills to solve problems through programming. All this, with a practical approach that will guarantee the implementation of the concepts learned.

By specializing in this field, graduates will not only find new opportunities in the technology sector, but will also excel in industries such as finance, healthcare and logistics, where programming plays a key role. In addition, they will be qualified to lead development projects, automate processes and optimize resources, increasing their job profile and facilitating their access to highly competitive opportunities.

Finally, the 100% online format of this program will offer the necessary flexibility to combine learning with other personal or professional responsibilities. TECH will provide students with an innovative platform accessible 24 hours a day. In addition, its Relearning methodology will optimize learning by reiterating key concepts in different contexts, facilitating a progressive and efficient assimilation of the content.

This **Master's Degree in Programming from Scratch** contains the most complete and up-to-date educational program on the market. Its most notable features are:

- Practical cases presented by experts in programming
- The graphic, schematic and eminently practical contents with which it is conceived gather scientific and practical information on those disciplines that are indispensable for professional practice
- Practical exercises where the process of 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

"From basic algorithms to application development, this program will take you from zero to expert in no time. You will become the professional that the technology industry needs!"

Enter the world of programming with an unbeatable syllabus. You will master Python, JavaScript and web development from scratch with the most complete Master's Degree on the market"

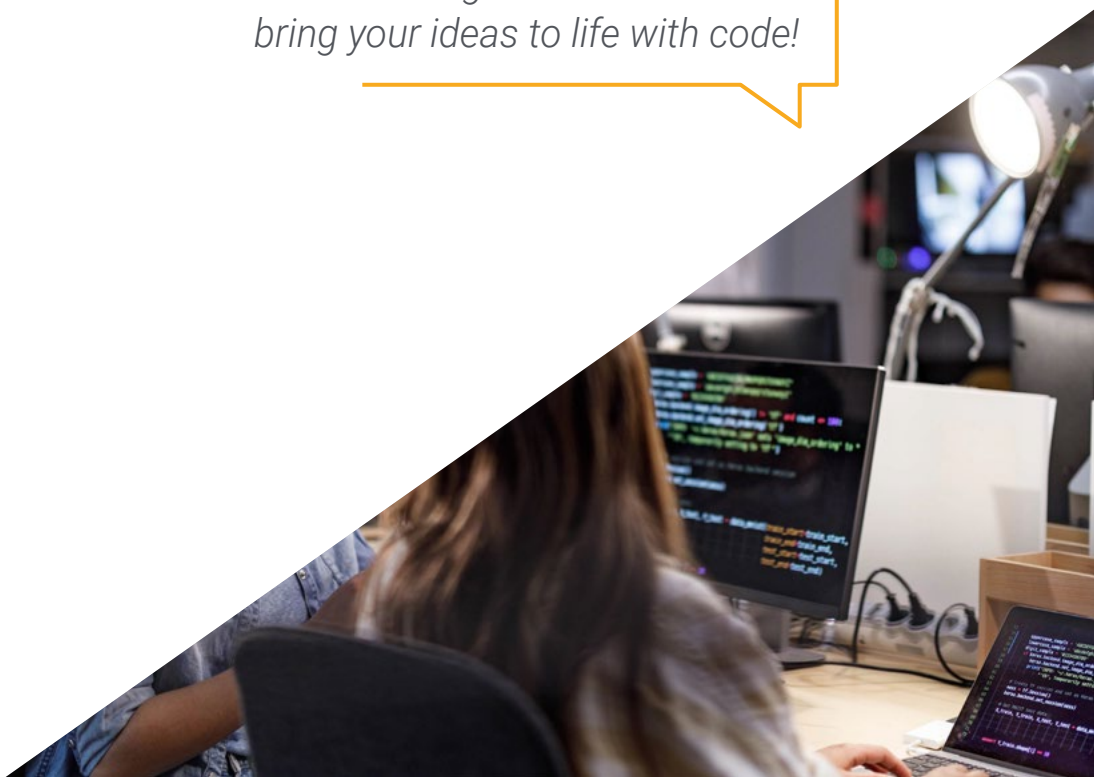
With the support of expert technology teachers, this qualification will guide you step by step to master the most in-demand languages on the market. Start today and build a better future!

You will discover an innovative approach that combines theory and real projects in this Master's Degree. Enroll now and bring your ideas to life with code!

Its teaching staff includes professionals from the field of programming, who bring their work experience to this program, as well as renowned specialists from leading companies and prestigious universities.

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

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



02

Why Study at TECH?

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

Study at the world's largest online university and guarantee your professional success. The future starts at TECH"

The world's best online university, according to FORBES

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

The best top international faculty

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

The world's largest online university

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

The most complete syllabuses on the university scene

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

A unique learning method

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

The official online university of the NBA

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

Leaders in employability

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

Google Premier Partner

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

The top-rated university by its students

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

03

Syllabus

Throughout the program, professionals will address the fundamentals of algorithms, data structures and programming logic, and then delve into languages such as Python, JavaScript and HTML. In addition, they will explore essential areas such as web design, application development and database management, offering a comprehensive view of the world of programming. Thanks to a progressive and structured approach, this syllabus will not only guarantee the acquisition of technical knowledge, but also the development of analytical and creative skills, which are essential in a competitive work environment.

This postgraduate degree is the ideal option to start your career in the technology sector and acquire the necessary tools to excel in any professional field related to Programming"

Module 1. Programming and Software Development from Scratch

1.1. Software and Hardware. Relationship and Differences
- 1.1.1. Software
- 1.1.2. Differences between Software and Hardware
- 1.1.3. Relationship between Software, Hardware and Programming

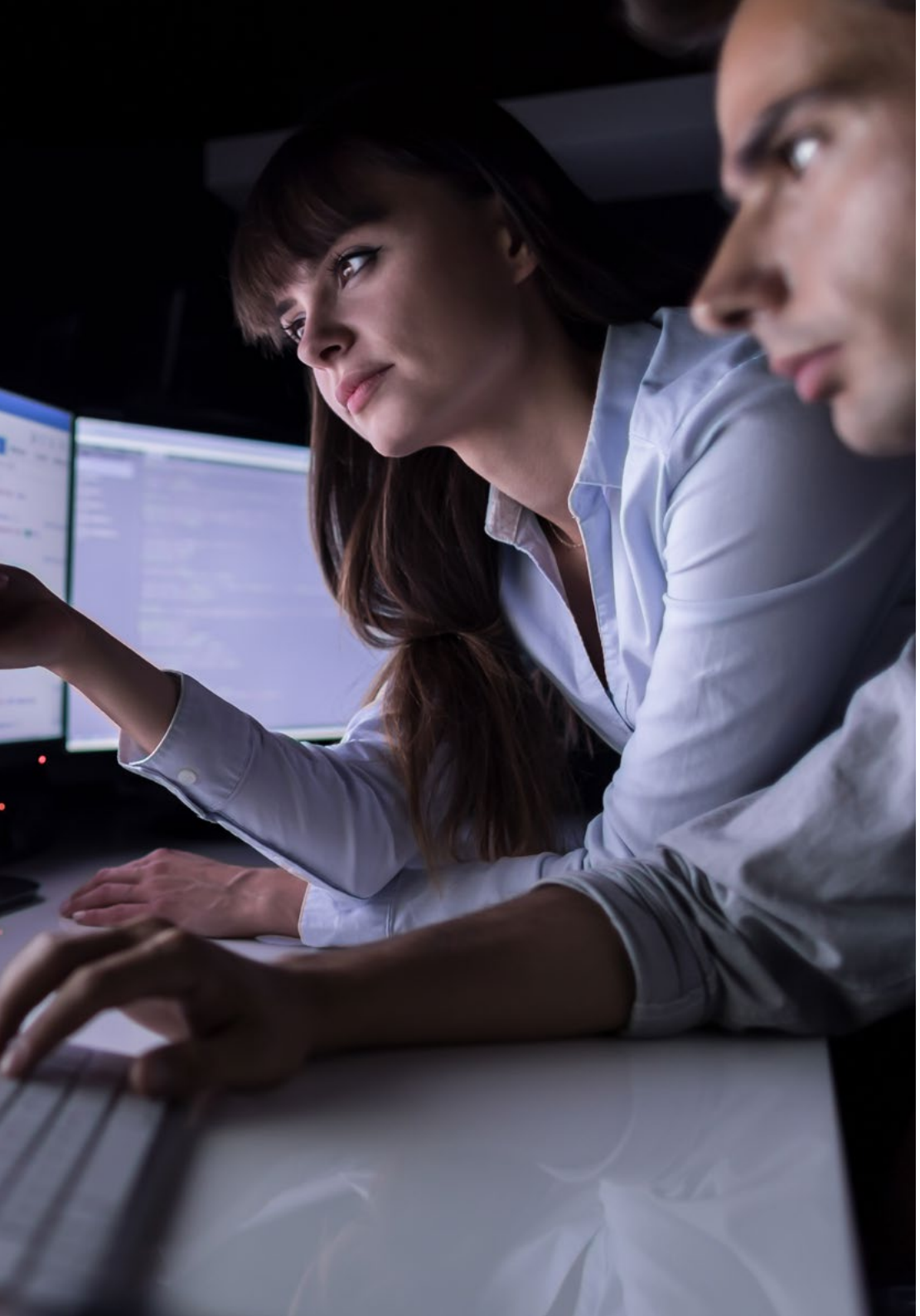
1.2. Programming. Key Aspects
- 1.2.1. Programming
- 1.2.2. Objectives and Applications
- 1.2.3. Programs, Source Code, Compilation and Execution
- 1.2.4. Errors: Syntax, Execution and Logical Errors

1.3. Programming from Scratch
- 1.3.1. Program Structure
- 1.3.2. Algorithms: Structure and Examples
- 1.3.3. Relationship between Algorithms and Programs
- 1.3.4. Problem Solving Using Algorithms

1.4. Programming Language Paradigms and Types
- 1.4.1. Programming Paradigms
  - 1.4.1.1. Imperative Paradigms
  - 1.4.1.2. Object-Oriented Paradigm
  - 1.4.1.3. Functional Paradigm
  - 1.4.1.4. Declarative Paradigm
- 1.4.2. Low-Level vs. High-Level Languages
- 1.4.3. Compiled vs. Interpreted Languages

1.5. Translation of Programming Languages
- 1.5.1. The Compiler. Compilation Process
- 1.5.2. The Interpreter: Interpretation Process
- 1.5.3. Differences between Compilation and Interpretation

1.6. Bits, Binary Operations and Logic Gates
- 1.6.1. Bit. Binary Representation
- 1.6.2. Basic Operations with Bits: AND, OR, XOR, NOT
- 1.6.3. Conversion between Binary and Decimal
- 1.6.4. Logic Gates: OR, AND, XOR, NOT, NOR and NAND



1.7. Designing Algorithms from Scratch
  1.7.1. Designing an Algorithm from Scratch
  1.7.2. Sequential, Conditional and Repetitive Algorithms
  1.7.3. Recursive Algorithms and their Comparison with Iterative Algorithms
1.8. Elements of the Program: Components and Structure
  1.8.1. Data Input and Output
  1.8.2. Variables and Constants: Use
  1.8.3. Data Processing and Manipulation
  1.8.4. Basic Functions and Procedures with Flowcharts
1.9. Control Structures with Flowcharts
  1.9.1. Control Structures. Functions in Programming
  1.9.2. Conditional Structures: Examples with Flowcharts
  1.9.3. Repetitive Structures: Examples with Flowcharts
1.10. Software Development Life Cycle and Models
  1.10.1. Software Life Cycle. Phases
  1.10.2. Development Models: Waterfall, Iterative and Agile
  1.10.3. Testing and Maintenance in Software Development

Module 2. Back-End Development I: Python from Scratch

2.1. Python from Scratch. Installation
  2.1.1. Python Language. Features
  2.1.2. Installing Python on Windows, macOS and Linux
  2.1.3. Setting up the Development Environment: IDEs and Code Editors
  2.1.4. First Program in Python: "Hello World"
2.2. Syntax and Variables in Python
  2.2.1. Structure of Python Code: Indentation
  2.2.2. Comments in Python
  2.2.3. Variables and Data Types in Python
  2.2.4. Arithmetic and Logical Operations in Python
2.3. Flow Control: Conditionals
  2.3.1. Control Structures
  2.3.2. Conditional Statements: if, elif, else
  2.3.3. Ternary Conditional Operator

2.4. Loops in Python
- 2.4.1. Use of Loops in Programming
- 2.4.2. “for” and "while" Loops
- 2.4.3. Flow Control in Loops: Break and Continue
- 2.4.4. Nested Loops

2.5. Functions in Python
- 2.5.1. Functions in Python. Uses
- 2.5.2. Parameters and Arguments of Functions
- 2.5.3. Return Values
- 2.5.4. Predefined Functions vs. User-Created Functions

2.6. Lists and Tuples in Python
- 2.6.1. Creation and Use of List in Phyton
- 2.6.2. Common Operations with Lists: Add, Remove, Modify
- 2.6.3. Tuples: Differences with Lists
- 2.6.4. Creating and Using Lists in Python

2.7. Dictionaries and Sets in Python
- 2.7.1. Dictionaries: Key-Value
- 2.7.2. Methods for Manipulating Dictionaries
- 2.7.3. Sets: Use
- 2.7.4. Comparison of Dictionaries and Sets.

2.8. File Handling in Python from Scratch
- 2.8.1. Opening and Closing Files
- 2.8.2. Opening Modes: Reading, Writing and Appending
- 2.8.3. Reading and Writing Text Files

2.9. Handling Errors and Exceptions
- 2.9.1. Types of Exceptions
- 2.9.2. Using Try, Except to Handle Errors
- 2.9.3. Creating Custom Exceptions

2.10. Best Practices and Debugging in Python
- 2.10.1. Debugging: Purpose
- 2.10.2. Debugging Techniques: Use of Print and Breakpoints
- 2.10.3. Best Practices in Code Writing

Module 3. Back-End Development II - Algorithms and Data Structures with Python from Scratch

3.1. Search Algorithms in Data Structures
- 3.1.1. Purpose of Search Algorithms in Data Structures
- 3.1.2. Linear Search: Implementation and Use Cases
- 3.1.3. Binary Search: Examples
- 3.1.4. Efficiency Comparison: Linear vs Binary Search

3.2. Sorting Algorithms in Data Structures (I). Basic Sorting Techniques Bubble Sort and Insertion Sort
- 3.2.1. Bubble Sort: Implementation and Analysis
- 3.2.2. Insertion Sort: Implementation and Use Cases
- 3.2.3. Comparison between Bubble Sort and Insertion Sort

3.3. Sorting Algorithms in Data Structures (II). Advanced Sorting Techniques: Selection Sort, Merge Sort and Quick Sort
- 3.3.1. Selection Sort. Implementation and Analysis
- 3.3.2. Merge Sort: Implementation
- 3.3.3. Quick Sort: Implementation
- 3.3.4. Comparison of Efficiency between Sorting Algorithms

3.4. Recursive Algorithms for Searching in Data Structures
- 3.4.1. Recursion. Use
- 3.4.2. Direct and Indirect Recursion
- 3.4.3. Recursive Algorithms: Factorial and Fibonacci. Examples

3.5. Complexity of Search Algorithms in Data Structures
- 3.5.1. Algorithmic Complexity. Efficiency Measurement
- 3.5.2. Big-O Notation
- 3.5.3. Complexity Analysis in Search and Sorting Algorithms

3.6. Advanced Data Structures
- 3.6.1. Trees: Terminology
- 3.6.2. Binary Trees: Operations
- 3.6.3. Binary Search Trees (BST): Search, Insertion and Deletion

3.7. Graph Algorithms
- 3.7.1. Graphs. Representation
- 3.7.2. Graph Algorithms: DFS and BFS
- 3.7.3. Comparison between DFS and BFS

3.8. Dynamic Programming
- 3.8.1. Dynamic Programming. Application
- 3.8.2. Differences between Dynamic Programming and Recursion
- 3.8.3. Optimization through Dynamic Programming

3.9. Data Search Algorithm Optimization Techniques
- 3.9.1. Data Search Algorithm Optimization Importance
- 3.9.2. Optimization Techniques: Memoization
- 3.9.3. Divide and Conquer Divide y vencerás

3.10. Other Algorithms in Python
- 3.10.1. Permutation and Combination Algorithms
- 3.10.2. Basic Hashing Algorithms
- 3.10.3. Counting and Subsets Generation Algorithms

Module 4. Back-End Development III - Flask, API Creation and Basic Architecture from Scratch

4.1. Flask as Back-End Framework
- 4.1.1. Back-End Framework. Purpose
- 4.1.2. Flask. Features
- 4.1.3. Preparing the Development Environment and Installing Flask
- 4.1.4. First Project with Flask: "Hello World"

4.2. Routes and HTTP Requests in Flask
- 4.2.1. Routes: How They Work in a Web Application
- 4.2.2. HTTP Methods in Flask: GET, POST, PUT and DELETE
- 4.2.3. Routes in Flask with Parameters and Data
- 4.2.4. Organizing Routes in a Project

4.3. Controllers and Handling Responses in Flask
- 4.3.1. Controller: Function and Responsibilities
- 4.3.2. Types of Responses in Flask: Text, JSON and HTML
- 4.3.3. Creating Controllers for APIs in Flask
- 4.3.4. CRUD Operations in Controllers

4.4. RESTful APIs
- 4.4.1. RESTful APIs. Principles
- 4.4.2. HTTP Methods and Conventions in RESTful APIs
- 4.4.3. Creating a RESTful API in Flask
- 4.4.4. Designing a RESTful API with CRUD Operations

4.5. Databases and Flask with SQLite
- 4.5.1. Databases in Web Applications
- 4.5.2. Connecting to SQLite in Flask Projects
- 4.5.3. Creating Tables and Models using SQLAlchemy
- 4.5.4. CRUD Queries in SQLite for Data Management

4.6. Authentication and Basic Security in APIs
- 4.6.1. Authentication and Authorization in APIs
- 4.6.2. Creating a User Authentication System
- 4.6.3. Using Tokens for Authentication in Flask
- 4.6.4. Protecting User Paths and Data in APIs

4.7. Data Validation and Error Handling
- 4.7.1. Error and Exception Handling in Flask
- 4.7.2. Data Validation in API Requests
- 4.7.3. Creating Custom Error Messages
- 4.7.4. Validation Strategies and Error Handling in CRUD

4.8. Structuring Scalable APIs
- 4.8.1. Organization and Structure of a Scalable Flask Project
- 4.8.2. Modularization and Separation of Responsibilities in APIs
- 4.8.3. Basic Optimization of APIs for Performance and Scalability
- 4.8.4. Organizational Strategy for Large Projects

4.9. Real-Time Communication with WebSockets
- 4.9.1. Websockets. Applications
- 4.9.2. Implementation of WebSockets in Flask with Flask-SocketIO
- 4.9.3. Real-Time Communication in Flask Applications

4.10. Application Deployment and Maintenance
- 4.10.1. Preparing Flask Applications for Production
- 4.10.2. Deployment on Popular Platforms such as Heroku and Render
- 4.10.3. Using Docker for Containerized Deployment
- 4.10.4. Monitoring and Maintaining Back-end Applications

Module 5. Object-Oriented Programming and Design Patterns from Scratch

5.1. Object-Oriented Programming (OOP) from Scratch
- 5.1.1. Object Oriented Programming
- 5.1.2. Differences between OOP and Structured Programming
- 5.1.3. OOP Elements: Classes, Objects, Methods and Attributes

5.2. Classes and Objects in Python
- 5.2.1. Creation of Classes and Objects in Python
- 5.2.2. Instance and Class Attributes
- 5.2.3. Special Methods (init, str, repr, etc.)
- 5.2.4. Static and Class Methods: Uses

5.3. Encapsulation and Abstraction in Classes
- 5.3.1. Encapsulation: Uses
- 5.3.2. Access Modifiers in Python
  - 5.3.2.1. Public, Protected and Private
- 5.3.3. Abstraction: Hiding Details and Improving Simplicity
- 5.3.4. Use of Properties (@property) for Access Control

5.4. Inheritance in Python. Usefulness in OOP
- 5.4.1. Inheritance: Usefulness in OOP
- 5.4.2. Creating Derived Classes and Multiple Inheritance in Python
- 5.4.3. Inherited Methods and Attributes and Overloading in Inheritance
- 5.4.4. Class Hierarchies and Base Class Management

5.5. Polymorphism and Overloading in Python
- 5.5.1. Polymorphism: Duck Typing
- 5.5.2. Polymorphism with Classes and Methods in Python
- 5.5.3. Overloading and Overwriting Methods in Python
- 5.5.4. Polymorphism in Software Design. Applications and Advantages

5.6. Class Relations and Complex Structure Design
- 5.6.1. Types of Relations: Association, Aggregation and Composition
- 5.6.2. Differences between Aggregation and Composition: Examples
- 5.6.3. Design of Complex Structures Using Class Relations

5.7. Design Patterns and SOLID Principles
- 5.7.1. Relevance of Design Patterns
- 5.7.2. Application of Design Patterns in OOP Projects. Advantages
- 5.7.3. Classification of Design Patterns.
- 5.7.4. SOLID Principles and their Importance in Object-Oriented Design

5.8. Creative Design Patterns
- 5.8.1. Purpose of the Creational Design Patterns
- 5.8.2. Singleton Pattern
- 5.8.3. Factory and Factory Method Pattern
- 5.8.4. Builder Pattern

5.9. Structural Design Patterns
- 5.9.1. Purpose of the Structural Design Patterns
- 5.9.2. Adapter Pattern
- 5.9.3. Decorator Pattern
- 5.9.4. Facade Pattern

5.10. Behavioral Design Patterns
- 5.10.1. Behavioral Patterns. Applications
- 5.10.2. Observer Pattern
- 5.10.3. Strategy Pattern

Module 6. Front End I - HTML and CSS from Scratch

6.1. HTML from Scratch
- 6.1.1. HTML. Purpose in Web Development
- 6.1.2. Structure of an HTML Document: DOCTYPE, <html>, <head>, <body>
- 6.1.3. Semantic and Content Tags: <header>, <nav>, <section>, <footer>
- 6.1.4. Essential Elements: Paragraphs (<p>), Lists (, <ol>), Links (<a>), Images (<img>)
- 6.1.5. Best Practices in HTML

6.2. Text and Multimedia Elements in HTML
- 6.2.1. Essential Text Tags: <p>, Headings, Lists, Bold and Italics
- 6.2.2. Multimedia Embedding: Attributes of <img>, <audio>, <video>
- 6.2.3. Essential Attributes for Accessibility (alt, aria-label)

6.3. HTML Forms
- 6.3.1. Form Structure and Components: <form>, <input>, <label>, <button>
- 6.3.2. Types of Input: Text, Email, Password, Submit Buttons
- 6.3.3. HTML5 Validation: Client-Side Field Validation
- 6.3.4. Forms with Basic Validation. Examples

6.4. CSS from Scratch
- 6.4.1. CSS Language from Scratch: Use and Relationship with HTML
- 6.4.2. CSS Syntax: Selectors, Properties and Values
- 6.4.3. Application of Styles in Line, Internal and External
- 6.4.4. Advanced Selectors: Of Type, Class, ID, Pseudoclasss (:hover, :focus)

6.5. Box Model in CSS
- 6.5.1. Box Model: Importance in CSS
- 6.5.2. Key Properties: Margin, Padding, Border, Width, Height
- 6.5.3. Using Box-Sizing for Precise Control of the Box Model
- 6.5.4. Design Applied to the Box Model. Examples

6.6. Typography and Text Styles in CSS
- 6.6.1. Color and Font Properties: Color, Font-Family, Font-Size
- 6.6.2. Advanced Text Styles: Bold, Italic, Alignment (text-align)
- 6.6.3. Text Spacing and Separation: Line-Height, Letter-Spacing
- 6.6.4. CSS Units of Measurement (px, em, rem) and Their Use in Typography

6.7. Layout Design with CSS - Flexbox
- 6.7.1. Flexbox: Purpose
- 6.7.2. Flexbox Properties: justify-content, align-items, flex-direction
- 6.7.3. Element Distribution and Alignment in Flexbox
- 6.7.4. Examples of Layouts with Flexbox

6.8. CSS Grid and Responsive Design with CSS
- 6.8.1. CSS Grid: Rows, Columns and Areas
- 6.8.2. Media Queries: Structure and Application on Different Devices
- 6.8.3. Responsive Design for Mobile, Tablet and Desktop
- 6.8.4. Adjusting Typography and Fluid Units in Responsive Design

6.9. Animations and Transitions in CSS
- 6.9.1. Transitions: Transition Property, Effects on :hover
- 6.9.2. Animations with CSS: Using @keyframes, Basic Animations
- 6.9.3. Techniques for Smoothing Transitions and Animations on the Web

6.10. Web Accessibility in Design
- 6.10.1. Web Accessibility: Importance
- 6.10.2. Accessible Site Design. Best Practices
- 6.10.3. ARIA Tags and Accessibility Validation Tools

Module 7. Front End II - JavaScript from Scratch

7.1. JavaScript from Scratch
- 7.1.1. JavaScript Language
- 7.1.2. Integration of JavaScript in HTML
- 7.1.3. First Program in JavaScript: “Hello World”

7.2. Variables and Data Types in JavaScript
- 7.2.1. Declaring Variables with var, let and const
- 7.2.2. Data Types: Numbers, Strings, Booleans
- 7.2.3. Converting Between Data Types

7.3. Control Structures in JavaScript
- 7.3.1. Conditional Statements: if, else if, else
- 7.3.2. Loops: for, while, do...while
- 7.3.3. Switch-case: Alternative to Multiple Conditional Statements
- 7.3.4. Break and Continue in Loops

7.4. JavaScript Functions
- 7.4.1. Function Declaration
- 7.4.2. Parameters, Return Values and Scope
- 7.4.3. Arrow Functions (=>) and Anonymous Functions
- 7.4.4. Callbacks and Recursion in Functions

7.5. DOM (Document Object Model) Manipulation with JavaScript
- 7.5.1. DOM: Structure of an HTML Document
- 7.5.2. Selecting DOM Elements (getElementById, querySelector)
- 7.5.3. Manipulating Elements: Changing Text, Styles, and Attributes
- 7.5.4. Events: Click, Input, Submit, and More

7.6. Arrays and Objects in JavaScript
- 7.6.1. Declaring and Using Arrays
- 7.6.2. Common Array Methods: push, pop, map, filter
- 7.6.3. Creating and Using Objects
- 7.6.4. Iterating over Arrays and Objects

7.7. Promises and Async in JavaScript
- 7.7.1. Async and Using Callbacks in JavaScript
- 7.7.2. Promises in JavaScript: Creation and Handling
- 7.7.3. Using Async and Await in JavaScript

7.8. APIs and Fetch in JavaScript
- 7.8.1. API in JavaScript: Purpose
- 7.8.2. Consuming REST APIs with Fetch
- 7.8.3. Handling Errors and Request Status

7.9. Local Storage in Web Browsers
- 7.9.1. Local Storage and Session Storage in Web Services and Applications
- 7.9.2. Storage and Retrieval of Data in Local Storage
- 7.9.2. IndexedDB as a Browser Database
- 7.9.3. Handling Cookies in JavaScript
- 7.9.4. Browser Storage: Examples

7.10. Best Practices in JavaScript and Tools for Developers
- 7.10.1. Code in JavaScript: Best Practices
- 7.10.2. Use of Browser Development Tools in JavaScript
- 7.10.3. Debugging and Error Handling in JavaScript

Module 8. Front End III - React.js from Scratch

8.1. React.js from Scratch
- 8.1.1. React JS as a Library for Developing Web Applications
- 8.1.2. Components and Virtual DOM in React JS: Architecture and Operation
- 8.1.3. Installation and Configuration with the NextJS Frameworks
- 8.1.4. First Component in React: "Hello World"

8.2. JavaScript XML or JSX, and Components in React
- 8.2.1. JSX: Syntax and Features
- 8.2.2. Creating Functional Components in React.js
- 8.2.3. Using Props to Pass Data Between Components
- 8.2.4. Functional Components vs. Class Components for Development in React.js

8.3. State and Events in React.js
- 8.3.1. Component State in React
- 8.3.2. Use of useState for State Management
- 8.3.3. Event Handling in React.js: onClick, onChange, Among Others
- 8.3.4. Examples of State and Event Management in React.js

8.4. Component Lifecycle and Effects in React
- 8.4.1. Component Lifecycle in React
- 8.4.2. Using useEffect to Manage Effects in React
- 8.4.3. Components with Mounting, Updating and Unmounting in React

8.5. Routing with React Router
- 8.5.1. SPA (Single Page Applications) and Routing in Web Applications
- 8.5.2. React Router Installation and Configuration
- 8.5.3. Creating Routes and Navigating Between Pages with React Router

8.6. Forms and Validation in React
- 8.6.1. Creating Interactive Forms in React
- 8.6.2. Handling User Input and Sending Data in React
- 8.6.3. Real-Time Form Validation in React

8.7. Consuming APIs in React
- 8.7.1. Consuming APIs with Fetch and Axios in React
- 8.7.2. Handling Loading, Success and Error States in React
- 8.7.3. Updating Components According to API Data in React

8.8. Reusable Components and External Libraries in React
- 8.8.1. Reusable Components in React
- 8.8.2. Creating Reusable Components in React
- 8.8.3. Using External Libraries like Material UI and Bootstrap in React

8.9. Global State Management in React
- 8.9.1. Global State Management with Native Options: Context API and Custom Hooks
- 8.9.2. External Libraries for Data Management
- 8.9.3. Comparing Approaches to Global State Management. Examples

8.10. React Application Deployment and Optimization
- 8.10.1. Preparing a React Application for Production
- 8.10.2. Deploying on Platforms such as Netlify and Vercel
- 8.10.3. Performance Optimization: Lazy Loading, Memoization, Server Components and Code Splitting
- 8.10.4. Monitoring and Maintaining React Applications in Production. Tools and Performance Analysis

Module 9. Database Management and Optimization from Scratch

9.1. Database from Scratch
- 9.1.1. Databases. Types
- 9.1.2. Relational vs. Non-relational Databases
- 9.1.3. SQL and NoSQL Programming Languages

9.2. Relational Data Modeling
- 9.2.1. Relational Database Model
- 9.2.2. Tables, Rows, and Columns in a Relational Database
- 9.2.3. Primary and Foreign Keys: Relationships Between Tables
- 9.2.4. Normalization: 1NF, 2NF, 3NF

9.3. SQL Language: DML and DDL
- 9.3.1. SQL: Structured Query Language
- 9.3.2. Create and Delete Queries: CREATE, DROP
- 9.3.3. SELECT, INSERT, UPDATE and DELETE Queries
- 9.3.4. Filtering and Sorting Data with SQL

9.4. Advanced SQL Queries
- 9.4.1. Joins: INNER JOIN and OUTER JOIN
- 9.4.2. Subqueries and Nested Queries
- 9.4.3. Aggregate Functions in SQL: SUM, AVG, COUNT

9.5. NoSQL Databases and MongoDB
- 9.5.1. NoSQL Database
- 9.5.2. Comparison between SQL and NoSQL
- 9.5.3. MongoDB: Document Database
- 9.5.4. Flexible Schemas in NoSQL

9.6. Database Optimization
- 9.6.1. Importance of Query Optimization
- 9.6.2. Using Indexes in Relational Databases
- 9.6.3. NoSQL Database Optimization

9.7. Database Security
- 9.7.1. Database Security
- 9.7.2. Encryption of Sensitive Data
- 9.7.3. User and Permission Management in Databases
- 9.7.4. Database Protection Strategies against Attacks

9.8. Database Scalability
- 9.8.1. Database Scalability
- 9.8.2. Horizontal and Vertical Partitioning
- 9.8.3. Database Replication and Clustering

9.9. Data Backup and Recovery
- 9.9.1. Importance of Database Backup
- 9.9.2. Automatic and Manual Backup Techniques
- 9.9.3. Data Recovery in Relational and NoSQL Databases

9.10. Database Implementation in Projects
- 9.10.1. Database Design for a Real Project
- 9.10.2. Database Integration with Back-End Applications

Module 10. Development Tools from Scratch: Linux, Version Control, CI/CD, Docker and Agile Methodologies

10.1. Linux from Scratch
- 10.1.1. Linux
- 10.1.2. Differences between Linux and Other Operating Systems
- 10.1.3. Popular Linux Distributions for Developers
- 10.1.4. Configuration and Customization of the Development Environment
- 10.1.5. Text Editors in Linux

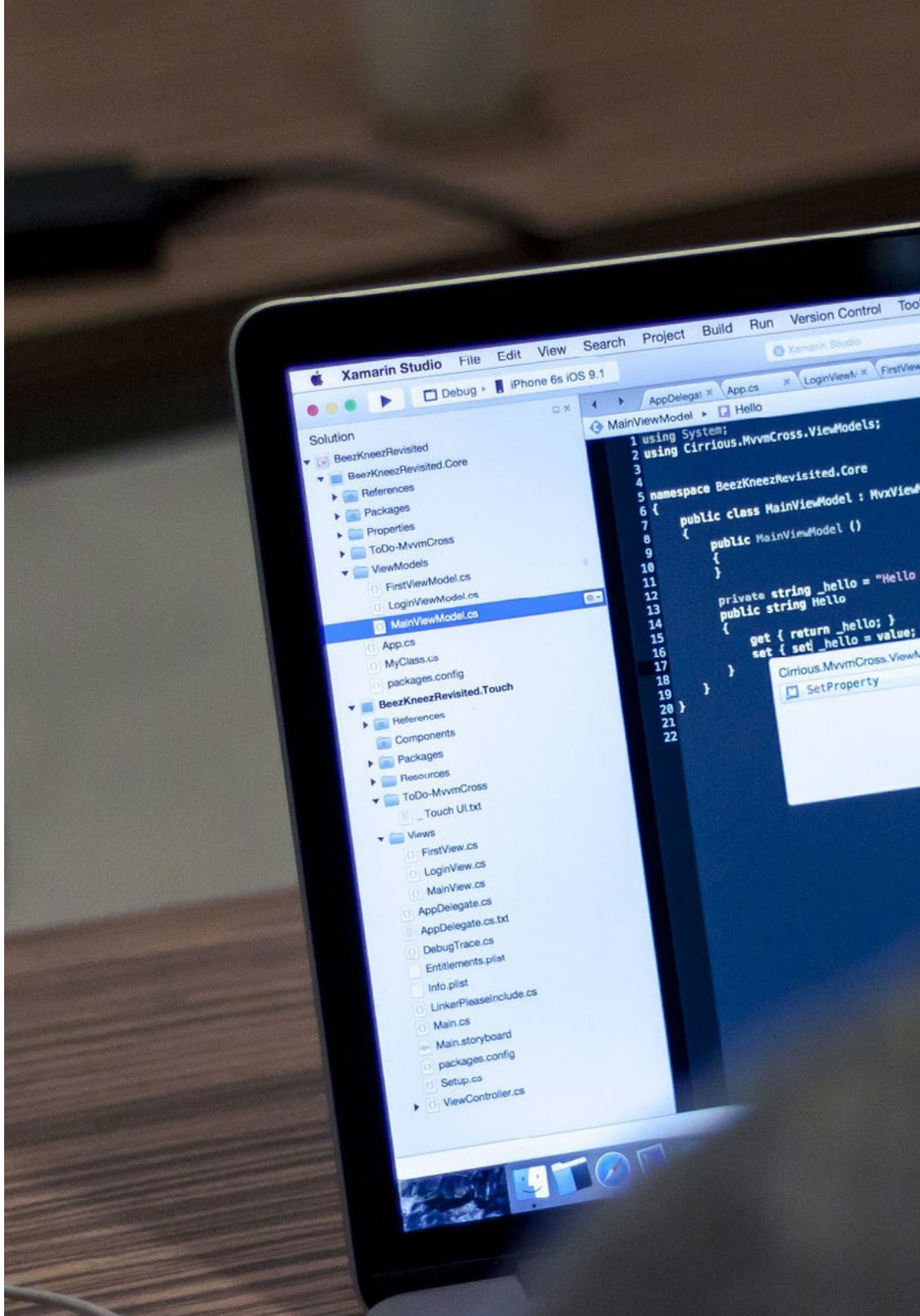
10.2. Using the Linux Terminal from Scratch
- 10.2.1. The Terminal. Uses and Functions
- 10.2.2. Navigation Commands and File Management in the Terminal
- 10.2.3. File and Directory Permissions in Linux
- 10.2.4. Command Redirection and Use of Pipes to Optimize Tasks

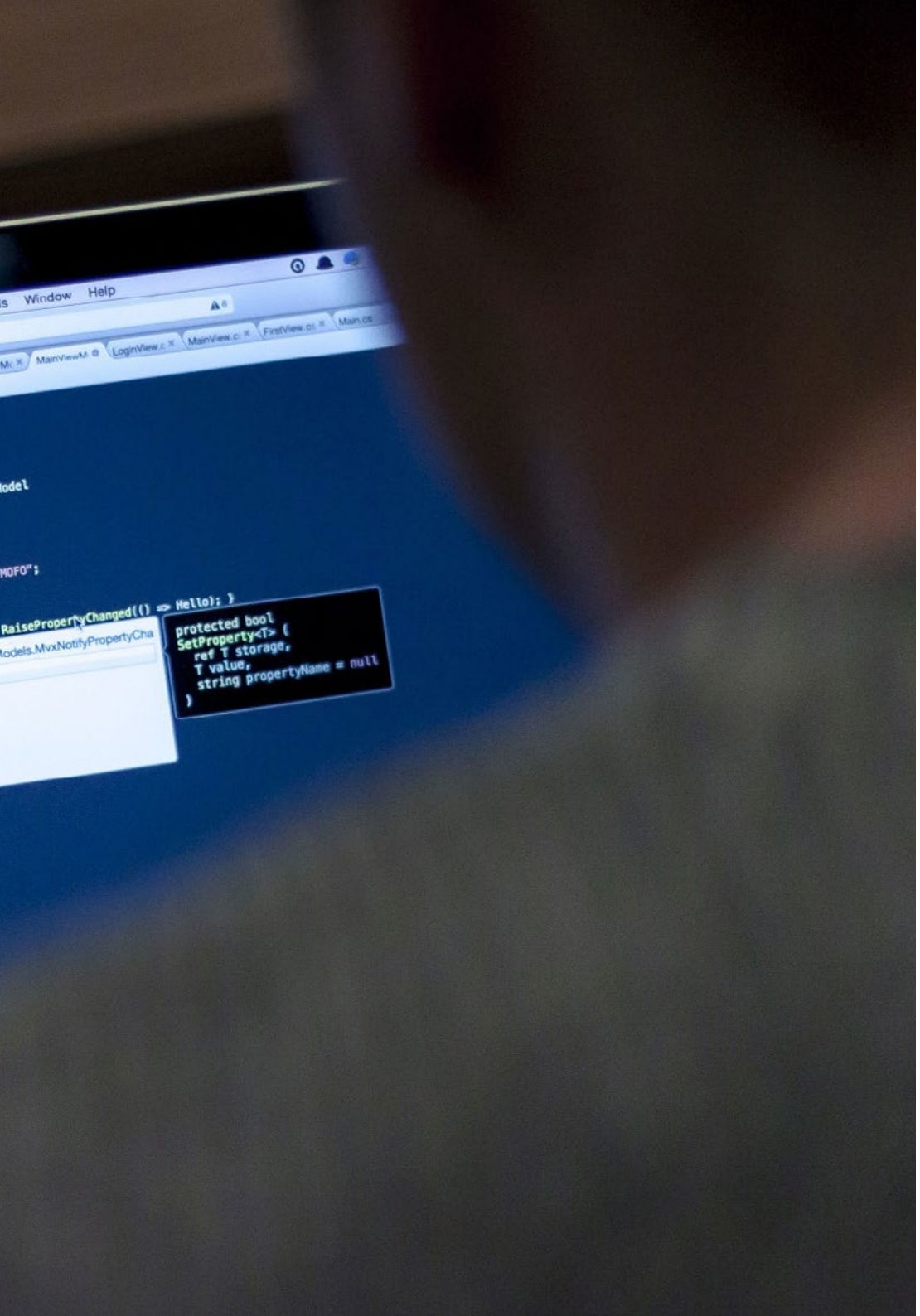
10.3. Version Control with Git from Scratch
- 10.3.1. Git: Cloud Providers
- 10.3.2. Creation and Management of Repositories
- 10.3.3. Workflow: git init, git add, git commit and git status
- 10.3.4. Working with Branches: Creation, Merging and Conflict Resolution

10.4. Collaborating in Teams with GitHub from Scratch
- 10.4.1. GitHub: Remote Repositories
- 10.4.2. Connecting a Local Repository to GitHub: git remote. Initial Configuration
- 10.4.3. Synchronization with Remote Repositories
- 10.4.4. Pull Requests and Collaborative Code Review

10.5. CI/CD (I) - Continuous Integration (CI) with GitHub Actions from Scratch
- 10.5.1. Continuous Integration (CI)
- 10.5.2. Workflow Configuration in GitHub Actions
- 10.5.3. Automation of Tests and Deployments

10.6. Docker from Scratch
- 10.6.1. Docker and Containers
- 10.6.2. Docker Installation and Configuration
- 10.6.3. Docker Container Creation and Management
- 10.6.4. Dockerfiles: Custom Image Creation





10.7. CI/CD (II) - Continuous Delivery (CD) with Docker and GitHub Actions from Scratch
- 10.7.1. Continuous Delivery (CD)
- 10.7.2. CD Pipeline Configuration with Docker and GitHub Actions
- 10.7.3. Automated Deployment with Docker Compose

10.8. Agile Methodologies from Scratch (I). Principles and Values
- 10.8.1. Agile Methodologies: Principles
- 10.8.2. The Agile Manifesto: Fundamental Values and Principles
- 10.8.3. Comparison with Traditional Methodologies: Waterfall vs. Agile

10.9. Agile Methodologies (II): Scrum from Scratch
- 10.9.1. Scrum and its Applicability
- 10.9.2. Key Roles in Scrum: Product Owner, Scrum Master and Development Team
- 10.9.3. Scrum Artifacts: Product Backlog, Sprint Backlog and Product Increment
- 10.9.4. Scrum Events: Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective

10.10. Agile Methodologies (III): Kanban and Metrics from Scratch
- 10.10.1. Kanban and its Visual Approach
- 10.10.2. Key Elements in Kanban: Columns, Cards and WIP Limits
- 10.10.3. Agile Metrics: Burnup, Burndown Charts, Velocity and Lead Time

You will learn from the comfort of your own home and at your own pace, with the innovative online methodology and flexibility you need. Don't wait any longer to boost your career in technology! Enroll today"

04

Teaching Objectives

The main goal of this Master's Degree is to provide professionals with the necessary skills to start their career in the world of programming in a solid and effective way. Throughout the program, they will develop a deep understanding of the fundamental concepts of programming, such as algorithms, data structures and programming logic, which form the basis on which they will build more advanced skills. In addition, they will acquire a mastery of programming languages widely used in the industry (Python, JavaScript and HTML), which will enable them to confidently tackle a variety of technology projects.

You will acquire the necessary skills to lead development projects, manage databases and optimize processes, increasing your professional competitiveness in the technology sector"

General Objectives

- Understand the fundamentals of programming and software development by identifying the essential elements of a program, control structures and the relationship between software and hardware
- Understand the fundamentals of Python and its basic syntax, developing a suitable environment for back-end development
- Acquire a solid understanding of fundamental algorithms and data structures
- Master the fundamentals of back-end development and its role within a software architecture
- Manage the principles of object-oriented programming and apply them in the construction of flexible, reusable and maintainable software, using Python as the programming language
- Develop the fundamental concepts of HTML and CSS to structure and style web pages
- Analyze the JavaScript language, from the basics to advanced techniques
- Introduce the fundamental concepts of React.js to build interactive web applications
- Manage database types and their query languages, both in relational and NoSQL environments
- Detail the automation of testing, integration and code deployment

Specific Objectives

Module 1. Programming and Software Development from Scratch

- Define and differentiate between software and hardware
- Understand the basic concepts of programming
- Understand the basic structure of a program
- Explore and analyze the different programming paradigms

Module 2. Back-End Development I: Python from Scratch

- Master the characteristics of Python
- Understand the structure and basic syntax of Python
- Develop skills in flow control using conditionals
- Apply loops to create repetition cycles in Python

Module 3. Back-End Development II - Algorithms and Data Structures with Python from Scratch

- Implement and compare types of search algorithms in data structures
- Analyze sorting algorithms such as bubble, insertion, selection, merge sort and quick sort
- Examine algorithmic complexity and efficiency measurement using Big O notation
- Represent graphs and perform depth-first (DFS) and breadth-first (BFS) searches

Module 4. Back-End Development III - Flask, API Creation and Basic Architecture from Scratch

- Develop RESTful APIs using Flask
- Integrate databases into Flask applications
- Implement authentication and security in APIs
- Design the basic architecture of a back-end application with Flask

Module 5. Object-Oriented Programming and Design Patterns from Scratch

- Define the key concepts of Object-Oriented Programming such as classes, objects, attributes, methods, encapsulation, abstraction, inheritance and polymorphism
- Understand the use of encapsulation and abstraction in classes using the Python programming language
- Examine the concept of polymorphism and overloading within the Python language, understanding its applications and advantages
- Determine the types of relationships between classes such as association, aggregation and composition

Module 6. Front End I - HTML and CSS from Scratch

- Identify the basic structure of an HTML document and its importance in web development
- Use HTML to organize and present content on the web in a semantic and accessible way: Web architecture
- Apply styles with CSS to improve the visual presentation of elements
- Use the CSS box model to structure and distribute elements in the interface

Module 7. Front End II - JavaScript from Scratch

- Understand syntax and data types in JavaScript
- Learn to structure code using functions and control structures
- Manipulate the DOM to interact with dynamic web pages
- Work with APIs and handle asynchrony using Promises and async/await

Module 8. Front End III - React.js from Scratch

- Understand the use of JSX to create declarative interfaces
- Learn to work with functional components, props and life cycles
- Manage local and global states using modern tools such as Context API and Redux Toolkit
- Implement routing to build single-page applications (SPAs)

Module 9. Database Management and Optimization from Scratch

- Identify the different types of databases and their characteristics
- Understand and apply the relational data model
- Develop SQL skills for database management
- Use advanced SQL queries

Module 10. Development Tools from Scratch: Linux, Version Control, CI/CD, Docker and Agile Methodologies

- Operate the Linux operating system at the command line level
- Master the use of Git for version control
- Implement Continuous Integration and Deployment (CI/CD) pipelines
- Create and manage Docker containers

Do you want to master the language of the future? This Master's Degree represents the first step towards the technological future you have always wanted. Enroll now!"

05

Career Opportunities

This degree opens up a wide range of career opportunities in a constantly expanding technology sector. As companies in all sectors become increasingly digital, the demand for professionals with programming skills is not only growing, but also diversifying, opening up opportunities in areas as varied as software development, database management, web design and process automation. Upon graduation, students will be perfectly qualified to take on roles in technology companies, innovative start-ups and organizations in any sector looking to integrate digital solutions into their operations.

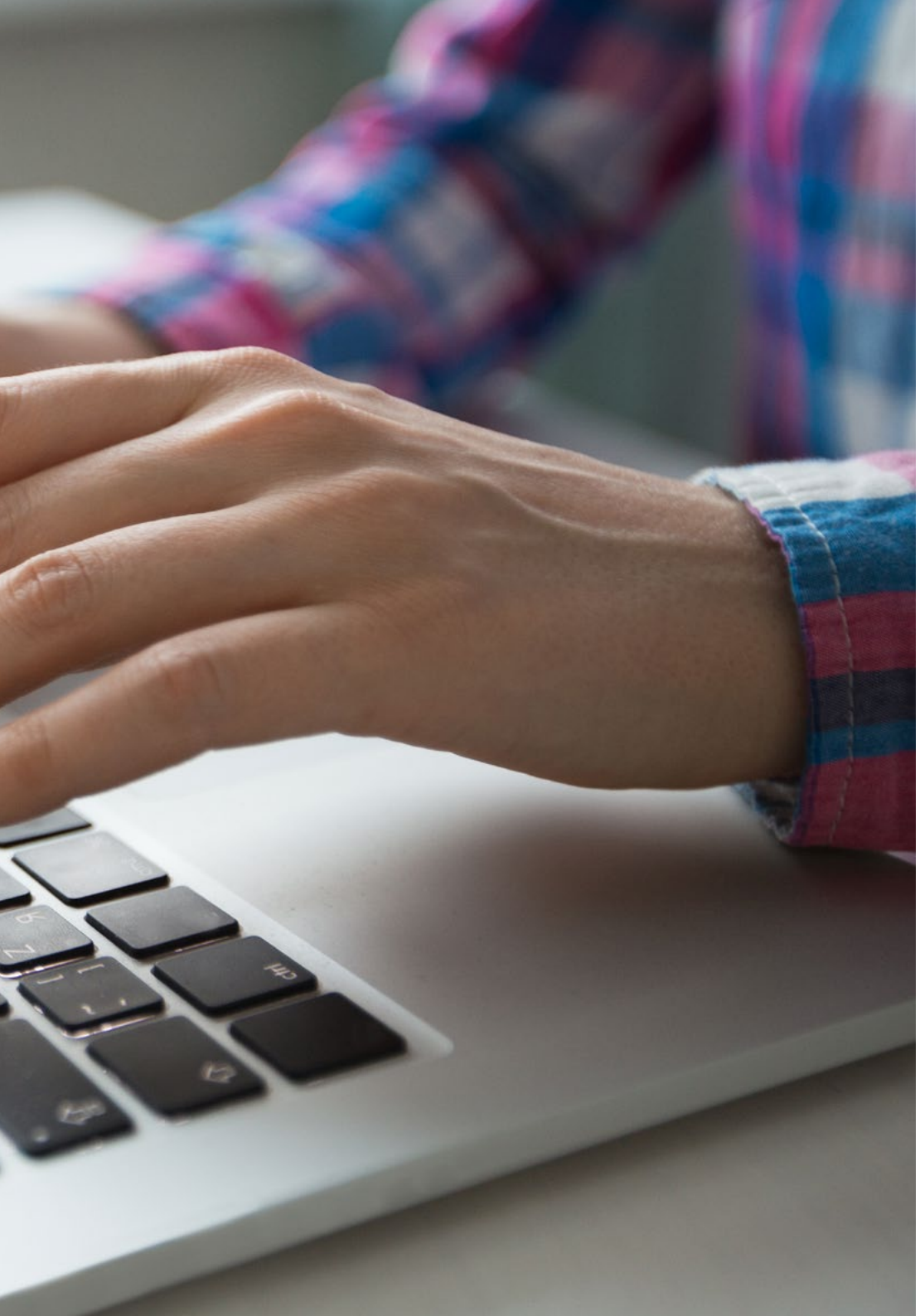
Not only will you become a high-level expert, but you will also address a wide variety of professional opportunities, ensuring your success in a labor market in continuous growth and transformation"

Graduate Profile

Graduates will be highly skilled professionals, ready to face the technological challenges of the future. They will not only have a firm grasp of the fundamental concepts of algorithms, data structures and programming languages, but will also have developed a deep understanding of the tools and technologies most in demand in the industry. With a solid foundation in programming and application development, these experts will be able to tackle technological projects autonomously and effectively. Furthermore, they will stand out for their ability to solve complex problems, apply innovative solutions and manage systems and databases.

> *You will become a versatile professional, with solid technical expertise and strategic vision, capable of integrating programming into technological solutions that respond to the needs of companies and society.*

- **Critical thinking and problem solving:** Analyze complex situations and find innovative and efficient solutions through programming
- **Teamwork and multidisciplinary collaboration:** Work in multidisciplinary teams, collaborating effectively with professionals from different areas, such as designers, engineers and data analysts, to achieve common goals in technology projects
- **Time management and learning autonomy:** Manage time effectively to meet tight deadlines and develop projects independently and responsibly
- **Adaptability to new technologies:** Manage new tools and programming languages, ensuring they are constantly kept up to date in a technological environment that is evolving at a fast pace



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

1. **Software Developer:** Responsible for designing, creating and maintaining computer applications and programs according to the needs of the client or company
2. **Web Developer:** Manages the creation, design and maintenance of websites, ensuring their functionality and optimization on different devices
3. **Database Engineer:** Responsible for designing, implementing and managing databases, guaranteeing their efficiency, security and availability
4. **Systems Analyst:** Developer of existing computer systems, proposing improvements to optimize the performance and efficiency of processes
5. **Front-End Programmer:** Responsible for the development of the visual part of applications and websites, improving the user experience
6. **Back-End Programmer:** Responsible for developing the server logic and databases of applications and websites, guaranteeing their correct functioning
7. **Network Administrator:** Responsible for managing and maintaining network infrastructures within a company, ensuring their stability and security
8. **Technology Consultant:** Advisor and consultant in companies, providing information on the implementation of technological solutions appropriate to their needs, improving their performance and competitiveness
9. **Mobile Application Developer:** Manager of the design, development and maintenance of applications for mobile devices, ensuring their usability and performance
10. **Automation Software Engineer:** Responsible for designing and developing automated solutions to improve the efficiency of business processes, using advanced programming and technology.

06

Study Methodology

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

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

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

The student: the priority of all TECH programs

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

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

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

The most comprehensive study plans at the international level

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

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

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

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

Case Studies and Case Method

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

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

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

Relearning Methodology

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

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

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

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

A 100% online Virtual Campus with the best teaching resources

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

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

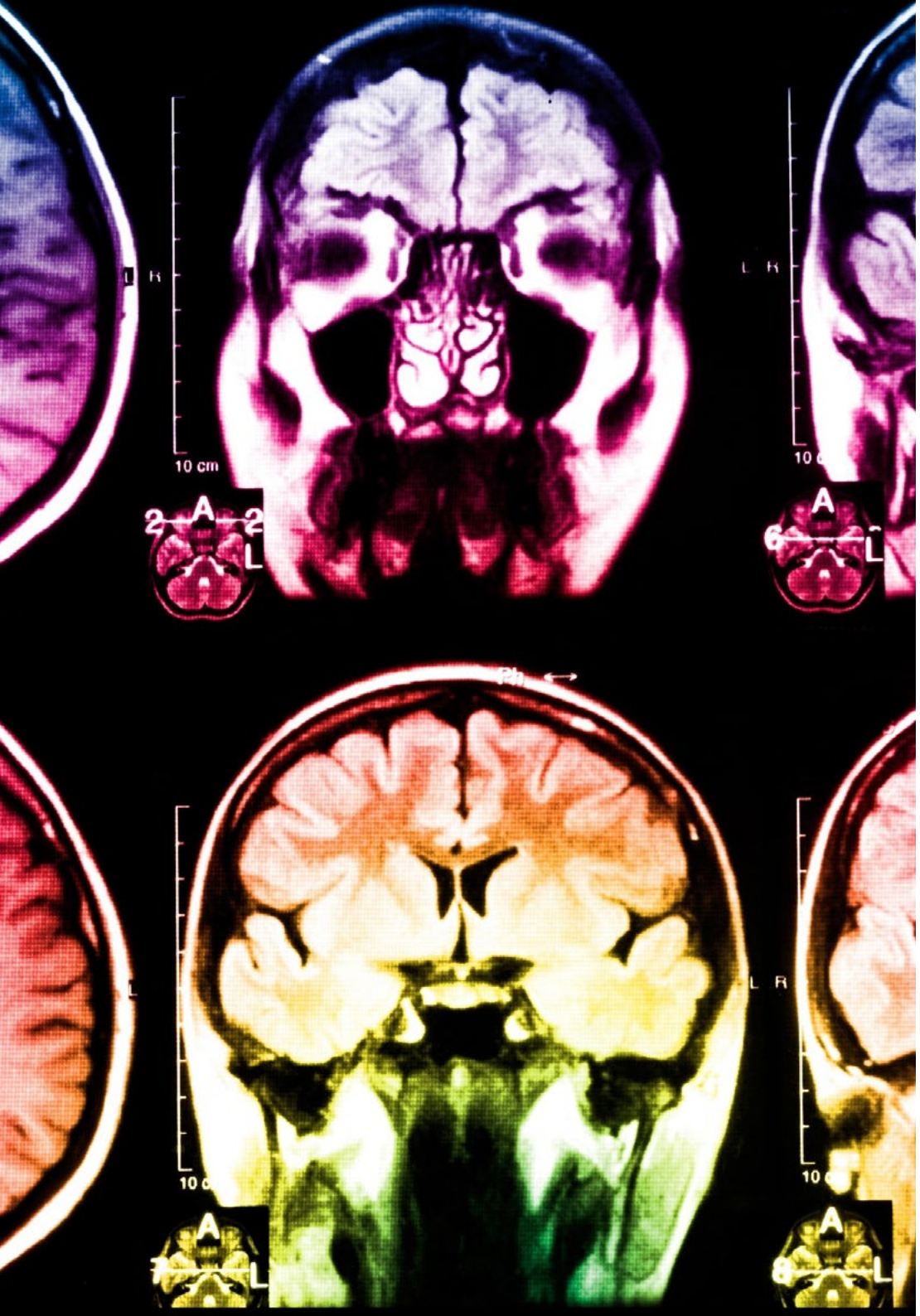
In order to facilitate tutor-student contact as much as possible, you will have a wide range of communication possibilities, both in real time and delayed (internal messaging, telephone answering service, email contact with the technical secretary, chat and videoconferences).

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

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

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

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



The university methodology top-rated by its students

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

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

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

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

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

Study Material

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

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

Practicing Skills and Abilities

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

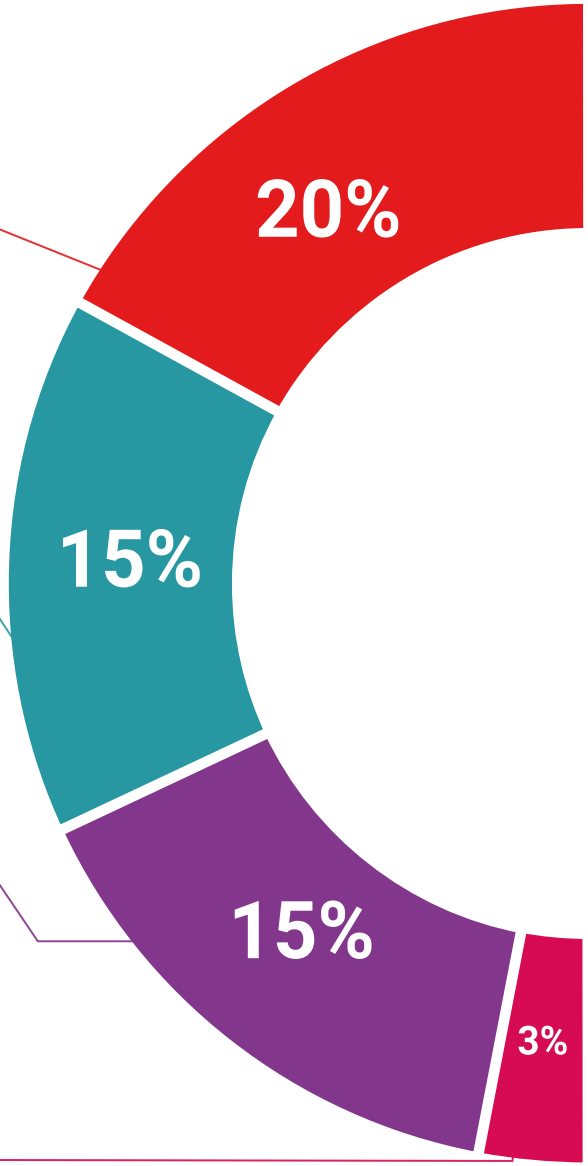
Interactive Summaries

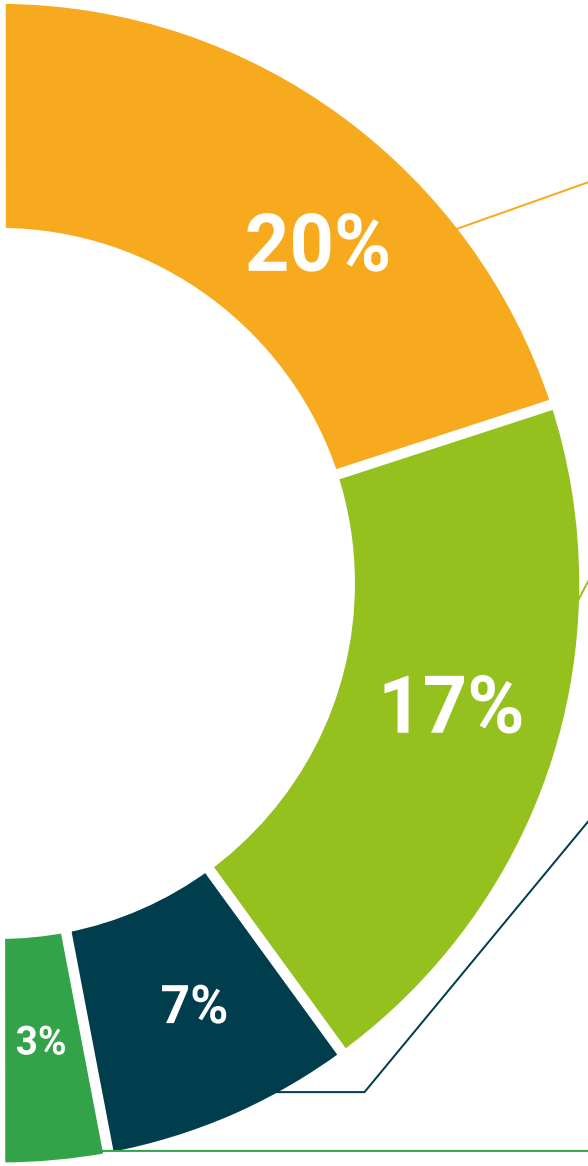
We present the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

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

Additional Reading

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





Case Studies

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

Testing & Retesting

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

Classes

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

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

Quick Action Guides

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

07

Teaching Staff

The teaching staff is made up of highly qualified professionals with extensive experience in the field of technology. Each one has been carefully selected to guarantee that the students receive quality education, based on up-to-date knowledge and real-world practice in the field. In this sense, their work experience in renowned companies and their academic background allow them to offer a comprehensive and applied perspective on Programming. In short, the mentors are experts in various areas of technology, which provides a multidisciplinary and enriching vision.

The teaching staff is a fundamental pillar in the success of the program. The quality and experience of the mentors will ensure a comprehensive education that will prepare you to face the technology sector with confidence and ability"

Management

Dr. Lucas Cuesta, Juan Manuel

- Senior Software Engineer and Analyst at Indizen – Believe in Talent
- Senior Software Engineer and Analyst at Krell Consulting and IMAGiNA Artificial Intelligence
- Software Engineer at Intel Corporation
- Software Engineer at Intelligent Dialog Systems
- PhD's Degree in Electronic Systems Engineering for Intelligent Environments from the Polytechnic University of Madrid
- Graduate in Telecommunications Engineering at the Polytechnic University of Madrid
- Master's Degree in Electronic Systems Engineering for Intelligent Environments from the Polytechnic University of Madrid

Mr. Márquez Ruiz de Lacanal, Juan Antonio

- Software Developer at GTD Defense & Security Solutions
- Software Developer at Solera Inc
- Development and Research Engineer at GRVC Sevilla
- Co-founder of Unmute
- Co-founder of VR Educa
- Academic Exchange in Engineering and Entrepreneurship at the University of California, Berkeley
- Degree in Industrial Engineering from the University of Sevilla

Professors

Dr. Luna Perejón, Francisco

- Specialist in Computer Architecture and Technology
- PhD in Computer Engineering from the University of Sevilla
- Master's Degree in Computer Engineering from the University of Sevilla
- Degree in Health Engineering from the University of Sevilla
- Degree in Computer Engineering and Computer Technology from the University of Sevilla
- Member of: Robotics and Computer Technology Research Group (TEP108)

Mr. Péris Millán, Eduardo

- Director of the Technological Consultancy Department
- Specialist in Computer Engineering
- Master's Degree in Strategic Management of Information and Knowledge in Organizations
- Master's Degree in Leadership and Public Management
- Expert in Public Management
- Expert in Computer Systems for Smart CITIES

Mr. Pi Morell, Oriol

- Functional Analyst at Fihoca
- Hosting and Mail Product Owner CDMON
- Functional Analyst and Software Engineer at Atmira and CapGemini
- Teacher at ORACLE Forms CapGemini and Atmira
- Degree in Technical Engineering in Computer Management from the Autonomous University of Barcelona
- Master's Degree in Artificial Intelligence from the Catholic University of Avila
- Professional's Degree in Business Administration and Management by IMF Smart Education
- Master's Degree in Information of Systems Management by IMF Smart Education
- Postgraduate Degree in Design in Patterns from the Open University of Catalonia

Mr. Grillo Hernández, José Enrique

- Application Developer and Technology Analyst
- Senior Mobile Applications Developer at Globant
- Android Developer at Plexus Tech
- Senior Android Developer at RoadStr
- Senior Mobile Developer at Avantgarde IT-Information Technology Services
- Project Leader at Smartdess
- Developer at Educatablet
- Technology Analyst at Corporate Mobile Solutions
- Master's Degree in System Engineering from the Simón Bolívar University

Ms. Domínguez Valderrama, Desirée

- *Lead Product & Growth Strategist*
- Master's Degree in Graphic Design and Creativity from the Business School of the Sevilla Chamber of Commerce
- Expert in UX/UI Designer from CoderHouse
- Expert in Technology and Entrepreneurship

Take this opportunity to learn about the latest advances in this field in order to apply it to your daily practice"

08

Certificate

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

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

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

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

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

Title: **Master's Degree in Programming from Scratch**

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 Programming from Scratch

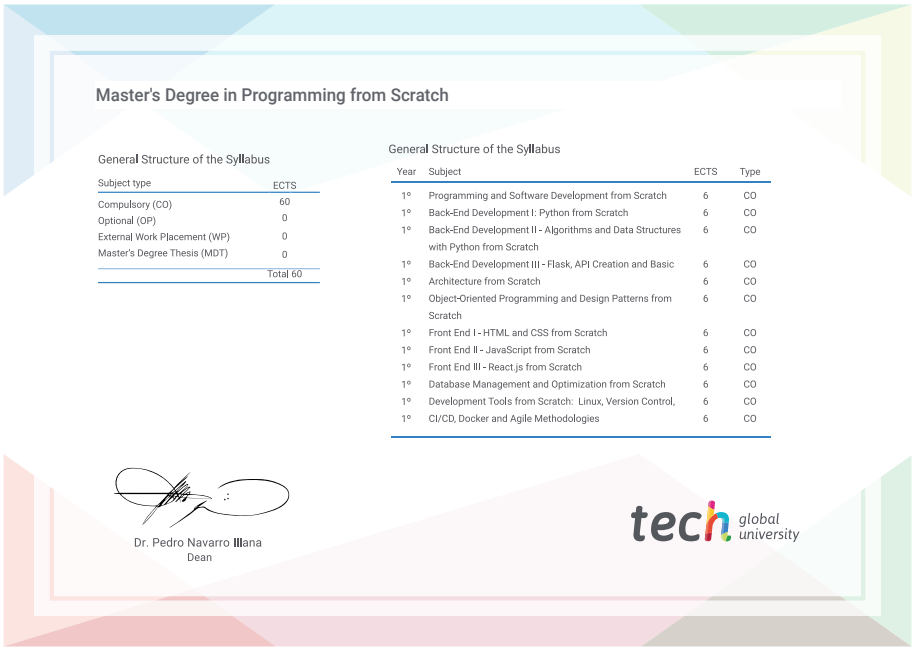
This is a private qualification of 1,800 hours of duration equivalent to 60 ECTS, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

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 Programming from Scratch

General Structure of the Syllabus

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

General Structure of the Syllabus

Year	Subject	ECTS	Type
1º	Programming and Software Development from Scratch	6	CO
1º	Back-End Development I: Python from Scratch	6	CO
1º	Back-End Development II - Algorithms and Data Structures with Python from Scratch	6	CO
1º	Back-End Development III - Flask, API Creation and Basic	6	CO
1º	Architecture from Scratch	6	CO
1º	Object-Oriented Programming and Design Patterns from Scratch	6	CO
1º	Front End I - HTML and CSS from Scratch	6	CO
1º	Front End II - JavaScript from Scratch	6	CO
1º	Front End III - React.js from Scratch	6	CO
1º	Database Management and Optimization from Scratch	6	CO
1º	Development Tools from Scratch: Linux, Version Control,	6	CO
1º	CI/CD, Docker and Agile Methodologies	6	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
Programming from Scratch

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

Master's Degree
Programming from Scratch

tech global university