Professional Master's Degree

Full Front-End Programming Stack Developer

tech technological university

Professional Master's Degree
Full Front-End Programming Stack Developer

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

Website: www.techtitute.com/in/information-technology/professional-master-degree/master-full-front-end-programming-stack-developer

Index

01

Introduction

Front-End developers are an essential part of any web development Their nature of union between the graphic design part and the Back End programming itself requires them to have a very specific and developed knowledge, which in turn must be updated to meet the increasingly demanding challenges of the present. Being a specialty that offers multiple opportunities at the professional level, TECH has assembled the best possible teaching team to develop an extensive, comprehensive and methodological university program. Focusing on the most advanced tools and methodologies in the development of web solutions, this program offers the necessary impulse to take the step towards Full Stack Front-End programming.

Become an essential part of any web development, leading and coordinating IT teams thanks to the perfected work and leadership methodology of this Professional Master's Degree"

Full Stack Development is a particularly interesting option for all IT professionals looking to significantly boost their careers. The skills required to stand outin the industry are extensive, meaning the opportunities to thrive and even lead development teams are abundant.

Thanks to the exhaustiveness with which all the contents of this program have been developed, the graduates will be able to direct their career towards Front-End Web Development, page layout, customer experience specialist or DevOps. With a 360º vision of the entire process of building an application/web, the computer scientist will be able to tackle any type of project, also providing a development in the latest advances in all processes of the life cycle of a software.

A unique academic opportunity to access knowledge that combines the latest computer theory with top-level professional practice, provided by a teaching team of the highest quality. Their experience at the head of numerous relevant projects in the field of digital banking or telecommunications enriches the didactic content, providing a large number of real cases and complementary readings.

The great flexibility of this teaching is another of its most outstanding characteristics. There are no fixed schedules or face-to-face classes, being the student himself the one who decides when, where and how to assume the entire teaching load. The entire content of the virtual classroom is available for download and can be studied from any device with an internet connection.

This **Professional Master's Degree in Full Front-End Programming Stack Developer** contains the most complete and up-to-date educational program on the market. Its most notable features are:

- The development of case studies presented by experts in Front-End Full Stack Developer Programming
- 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 for experts and individual reflection work
- Content that is accessible from any fixed or portable device with an Internet connection

Sign up now and don't miss the opportunity that will take you to the top in leadership and development of the most ambitious IT projects”

You will reach an advanced level of specialization, being able to build any web solution required with a modern Customer Experience perspective and adapted to the current market"

Students will delve into agile methodologies and how they can be implemented in the development process, increasing their skills and cross-cutting competencies.

You will have access to extensive learning material, ranging from the Javascript programming language to tools such as CSS, Angular and ReactJS.

The program includes, in its teaching staff, professionals from the sector who bring to this program the experience of their work, in addition to recognized specialists from prestigious reference societies and 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 immersive education programmed to learn in real situations.

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

02

Objectives

The objective of this Professional Master's Degree in Full Front-End Programming Stack Developer, taking into account the multiple options offered by development, is none other than to provide the most advanced knowledge and techniques in this field. In this way, the computer scientists can even start developing their own projects or boost their professional career before finishing the program, thanks to the eminently practical approach of all the contents provided.

You will achieve your most ambitious professional goals thanks to the distinctive approach of this program, which will take you through all stages of Full Stack Front-End development"

General Objectives

- Generate specialized knowledge on key aspects of programming
- Encourage algorithmic thinking
- Provide the necessary tools and skills for development
- Promote the adoption of agile methodologies for project execution
- Develop specialized knowledge on the fundamentals of the Web
- Encourage the use of modern Front-End development tools and techniques
- Develop a web design to approach the layout correctly
- Assess the knowledge acquired

Specific Objectives

Module 1. Full Stack Developer

- Develop advanced programming knowledge
- Promote the use of version control systems and code hosting platforms
- Promote the use of Agile Methodologies
- Delve into the key concepts and operation of the internet
- Increase command line proficiency

Module 2. Front-End Programming

- Identify and understand the correct syntax of HTML and CSS
- Explore the various elements of HTML
- Determine the adaptive design approach
- Employ presentation formatting to web pages by applying cascading style sheets
- Incorporate CSS pre-processor
- Establish the benefits of using a pre-processor
- Generate specialized knowledge about design systems
- Establish design system usage criteria

Module 3. JavaScript Language Applied to Full Stack Developer

- Establish the basic and complex types offered by JavaScript
- Analyze the different ways of programming with the language and make a correct use in each situation
- Update knowledge to the latest versions
- Discover functional programming
- Examine asynchronous programming and its characteristics

Module 4. Web Layout Applied to Full Stack Developer

- Evaluate a web design to know how to place it temporally
- Examine the main CSS rules
- Present different CSS methodologies to obtain Responsive layouts
- Fundamentals of CSS waterfall development principles
- Identify the Bootstrap technology in any web design
- Analyze Bootstrap principles
- Develop a web mockup using Bootstrap
- Determine the development principles in a SaSS project

Module 5. JavaScript Tools. ReactJs Library

- Determine React functionalities
- Configure a project using Create React App
- Analyze the life cycle of components in React
- Generate specialized knowledge about modern React features such as Hooks and Context
- Set global states using Context
- Create and render lists and create forms with React
- Implement field validation in forms
- Styling components and elements
- Debugging, testing and deploying React applications

Module 6. JavaScript Framework. Angular

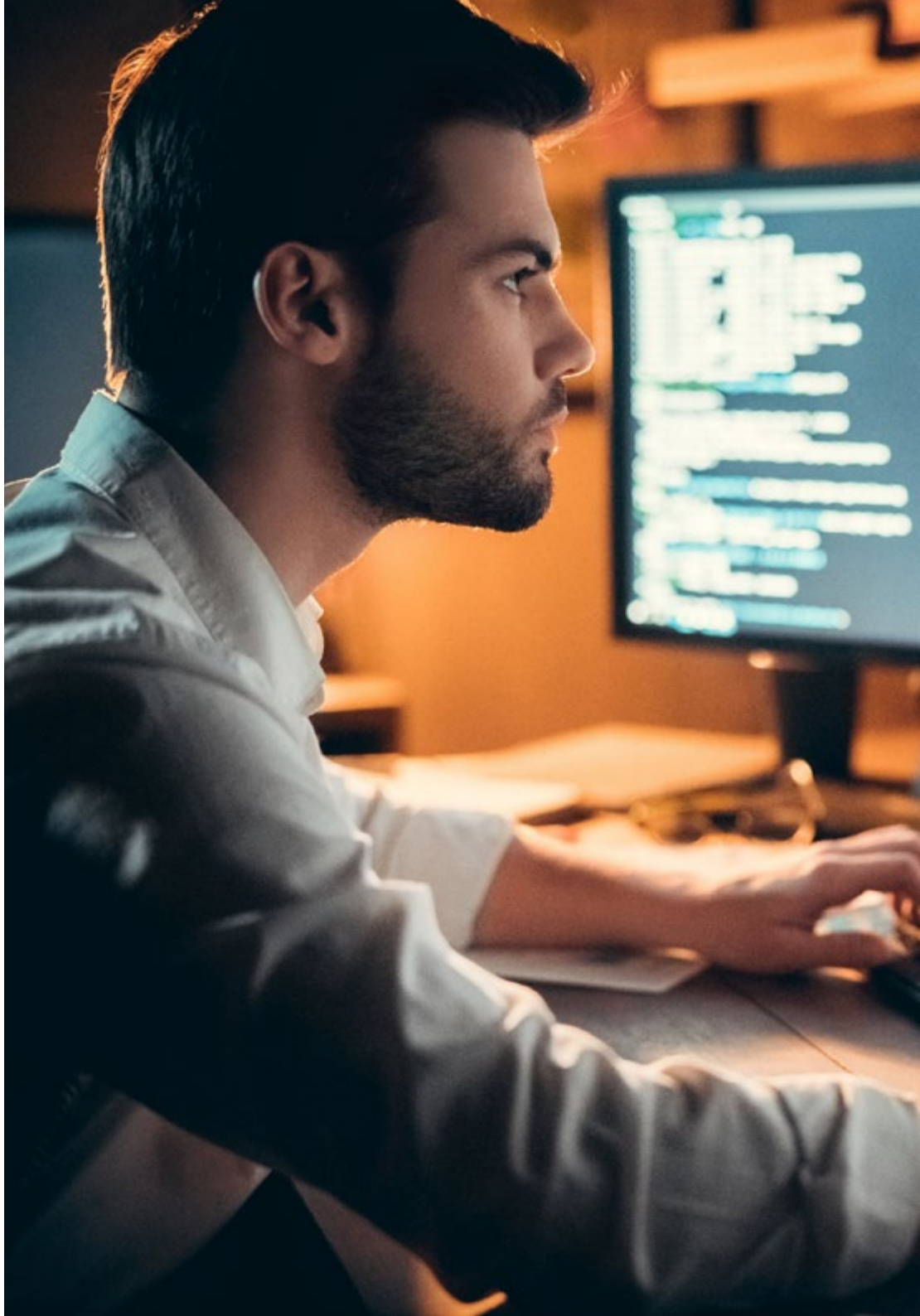
- Develop specialized knowledge about the architecture of the Framework
- Delve into Angular methodology
- Analyze the concept of components
- Organize the code correctly

Module 7. Programming in NodeJs Language

- Generate specialized knowledge about JavaScript types and their operators
- Analyze the best ways to program with the language
- Update knowledge to the latest versions
- Explore functional programming
- Develop asynchronous programming and its motivation
- Acquire the ability to build an application with NodeJSIndex

Module 8. Databases for Full Stack Developers

- Determine why to use a database in application development
- Examine the types of databases available and their differences
- Develop a clear idea of what to use each type of database for
- Analyze the use of database in current development paradigms



Module 9. UX CX. Customer Experience

- Analyze the importance of the user today and delve into the feedback culture
- Specify omnichannel strategies and personalization based on micro-interactions
- Study the evolution of web analytics to behavioral analytics
- Determine how Artificial Intelligence has taken CX to the next level
- Establish the most important web experience, mobility and accessibility analytics techniques
- Present the Design Thinking methodology and the user experience creation process
- Present specific prototyping and WireFraming tools, as well as Front-End development frameworks

Module 10. Continuous Integration and Application Deployment

- Realize the benefits of adopting an automated application deployment model
- Establish the differences between continuous integration, continuous delivery and continuous deployment
- Determine the main features of DevOps
- Assess some of the fundamental tools for implementing CI/CD pipelines
- Develop the essential factors for developing applications ready to support CI/CD processes
- Examine container technologies as a fundamental pillar of CI/CD practice

You will improve your skills and competences progressively, through 10 modules elaborated from the most solid knowledge and the proven experience of all the teachers"

03

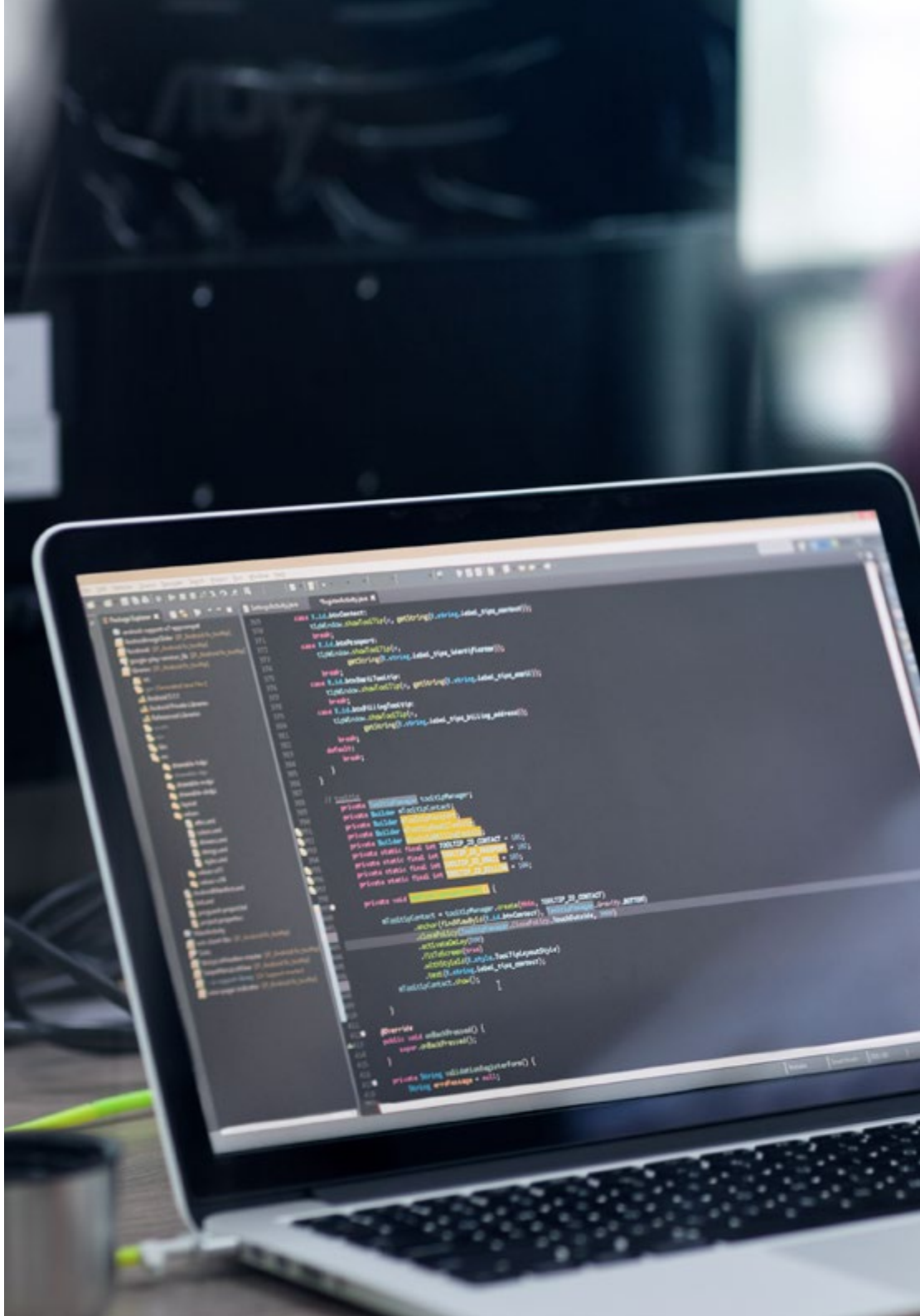
Skills

The skills that a Full Stack Front-End programmer must develop are multiple, especially in a field as disputed and specialized as IT. For this reason, the syllabus covers the different types of language most common in this field, as well as the tools and work philosophy that the computer scientist must follow to stand out with good professional distinction. This is possible thanks to the multidisciplinary nature of the teaching team itself, which has pooled all its knowledge regarding various areas of Front-End development.

You will give a high-quality boost to your CV by incorporating a multitude of skills and competencies required in the higher echelons of software development"

General Skills

- Correctly recognize the syntax of HTML and CSS languages
- Develop best practice criteria for web development
- Generate specialized knowledge about the JavaScript language
- Be able to develop any type of application with JavaScript
- Analyze the Bootstrap library
- Carry out layout projects with SaSS (Syntactically Awesome Stylesheets)
- Identify React syntax and how to program using it
- Apply best practices to the language
- Examine the loading and accessing process in each of the leading database types in your field
- Assess the most important tools and techniques in CX analysis and the common enterprise technology stack



Specific Skills

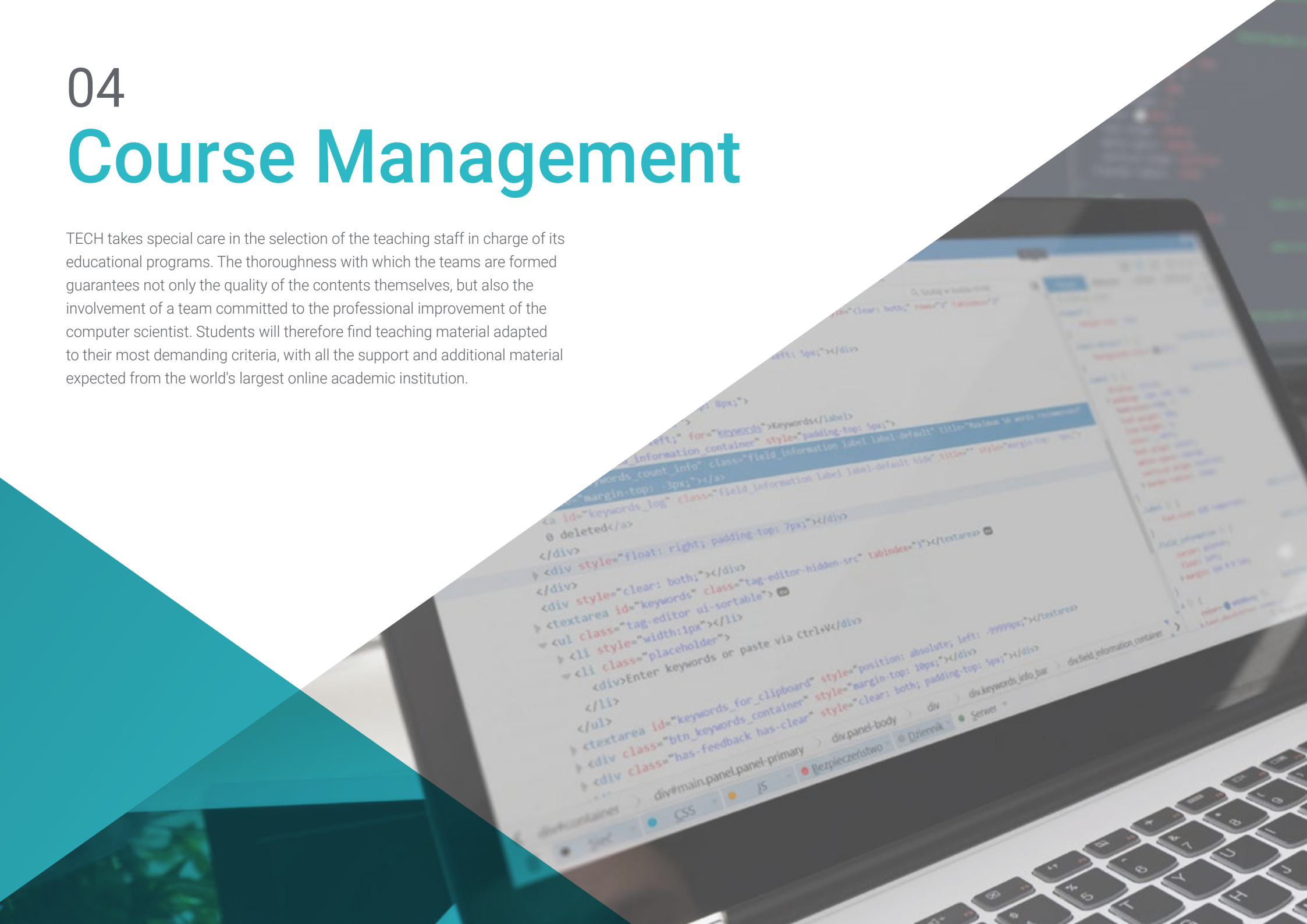
- Analyze different data structures
- Examine algorithm design and interpretation techniques
- Prepare the development environment
- Clone a website
- Generate a website with Bootstrap
- Compile CSS code with SaSS
- Develop your own Bootstrap-based CSS Framework using SaSS
- Generate a project and get it up and running
- Establish how to connect to and load/extract data from different database types
- Identify practices, use cases, technologies and tools from the CI/CD ecosystem, essential to support the overall process

You will develop throughout the course the necessary skills to succeed in Front-End development, enhancing not only your own knowledge but also your transversal skills"

04

Course Management

TECH takes special care in the selection of the teaching staff in charge of its educational programs. The thoroughness with which the teams are formed guarantees not only the quality of the contents themselves, but also the involvement of a team committed to the professional improvement of the computer scientist. Students will therefore find teaching material adapted to their most demanding criteria, with all the support and additional material expected from the world's largest online academic institution.



Students will be able to consult all their doubts directly with the teaching staff, obtaining a personalized tutoring adapted to their needs"

Management

D. Olalla Bonal, Martín

- Blockchain Technical Specialist at IBM SPGI
- Blockchain Technical Sales Specialist. IBM
- Director of Architecture. Blocknative
- Digital Electronics Technician
- Blockchain Architect -IT Infrastructure Architect - IT Project Manager. Business Areas: Software, Infrastructure, Telecommunications

Professors

D. Calzada Martínez, Jesús

- Senior Software Engineer at Devo
- Full Stack Developer at Blocknitive
- Front-End Manager at Infinia
- Full Stack Developer at Resem
- Java Developer at Hitec
- Graduate in Computer Engineering

Mr. Guerrero Díaz-Pintado, Arturo

- Technical Presales Engineer across Watson Customer Engagement portfolio (Marketing and Customer Experience solutions) within Spain, Portugal, Greece and Israel at IBM
- R&D Network Engineer at Telefónica
- Professional services consultant working with leading organizations in Europe, Middle East and Latin America since IBM
- Graduated in Telecommunication Engineering from the University of Alcalá and the Danish Technical University
- Outstanding collaborations in renowned universities and higher education centers in technology-related subjects such as Artificial Intelligence, Internet of Things, Cloud, Customer Experience and Digital Transformation

Mr. Pintado San Claudio, Bruno

- Development Coordinator at IDavinci
- Java Developer at the National Library of Spain
- Support developer and N1 network technician in Sanitas
- Systems support technician at Alcobendas City Council
- Communications Technician N1 for ADIF at the Atocha Telecommunications Center
- Graduated in Telecommunications Technical Engineering with specialization in Electronic Systems at the Polytechnic University of Madrid
- Graduate in Communications Electronics Engineering at the Polytechnic University of Madrid

Mr. Reyes Oliva, Luis

- Development developer and cloud architect at IBM
- Technical client manager for integrated accounts for BBVA at IBM
- Cloud Executive Selling at IBM
- Cloud and DevOps Architect at IBM
- Customer Software Architect at Telefónica
- Technical Solutions Architect at Rational
- Software Engineering Manager at Borland
- Software Development and Quality Assurance Manager at Altana Consulting
- Degree in Computer Engineering from Pontificia University of Salamanca, in Madrid

Mr. Frias Favero, Pedro Luis

- CTO - Swearit technologies
- COO - Key identification
- Expert in blockchain and decentralized applications - University of Alcala
- Full Stack Developer - Ironhack
- Industrial Engineer graduated from Yacambu University

D. Gómez Rodríguez, Antonio

- Cloud Solutions Engineer at Oracle
- Project Manager at Sopra Group
- Project Manager at Everis
- Project Manager at Public Company for the Management of Cultural Programs Department of Culture of Andalusia
- Information Systems Analyst. Sopra Group
- Degree in Telecommunications Engineering from the Polytechnic University of Catalonia
- Postgraduate Degree in Information Technologies and Systems, Catalan Institute of Technology
- E-Business Master, La Salle School of Business

05

Structure and Content

Given that the effort and investment made by the computer scientist in opting for a program of these characteristics are not minor, TECH is committed to a first-class pedagogical methodology, proven in efficiency and profitability. *Relearning* allows the student to acquire the most important knowledge of the program throughout the entire syllabus, therefore avoiding valuable study time that can be used in the multiple complementary materials that accompany each proposed topic.

Explore the different modules and knowledge topics through video summaries, detailed videos and motivational videos created by the teachers themselves"

Module 1. Full Stack Developer

1.1. Full Stack Developer I. Programming and Languages
- 1.1.1. Programming
- 1.1.2. Programming Roles
- 1.1.3. Languages and Framework
- 1.1.4. Algorithm
- 1.1.5. Characteristics of an Algorithm

1.2. Full Stack Developer II. Typology
- 1.2.1. Variables and Constants
- 1.2.2. Types
- 1.2.3. Operators
- 1.2.4. Declarations
- 1.2.5. Loops
- 1.2.6. Functions and Objects

1.3. Data Structure in Development
- 1.3.1. Linear Structure Types
- 1.3.2. Functional Structure Types
- 1.3.3. Tree Structure Types

1.4. Algorithm Design and Interpretation
- 1.4.1. Parallelism in Development. Divide and Conquer
- 1.4.2. Voracious Algorithms
- 1.4.3. Dynamic Programming

1.5. Environment and Tools for Full Stack Developer Oriented Development
- 1.5.1. Preparation of the Environment for Mac OS
- 1.5.2. Preparation of the Environment for Linux
- 1.5.3. Preparation of the Environment for Windows

1.6. Command Line. Typology and Operation
- 1.6.1. The Terminal
- 1.6.2. Emulators
- 1.6.3. Command Interpreter
- 1.6.4. First Commands
- 1.6.5. Navigation
- 1.6.6. Managing Files and Folders Using the Command Line Interface
- 1.6.7. Secure Shell. SSH
- 1.6.8. Advanced Commands

1.7. Git. Software Repository
- 1.7.1. Git Software Repository
- 1.7.2. Using Git
- 1.7.3. Software Repository
- 1.7.4. Branches
- 1.7.5. Duty Cycle
- 1.7.6. Commands

1.8. Code Versioning Hosting Service
- 1.8.1. Code Versioning Hosting Service
- 1.8.2. Suppliers
- 1.8.3. Repositories

1.9. Internet
- 1.9.1. Internet
- 1.9.2. Protocols Used in WWW
- 1.9.3. HTTP Protocol

1.10. Methodologies in Full Stack Development
- 1.10.1. Scrum
- 1.10.2. XP
- 1.10.3. Design Sprint

Module 2. Front-End Programming

2.1. HTML Language
- 2.1.1. HTML Document
- 2.1.2. Head Element
- 2.1.3. Body Element
- 2.1.4. Text:
- 2.1.5. Hyperlinks
- 2.1.6. Images
- 2.1.7. First Site

2.2. HTML Language. Layouts
- 2.2.1. HTML Language. Components
- 2.2.2. Traditional Layout
- 2.2.3. Semantic Layout

2.3. Cascading Style Sheets CSS
- 2.3.1. Inclusion of CSS in an HTML Document
- 2.3.2. Comments
- 2.3.3. Selectors
- 2.3.4. Advanced Selectors

2.4. CSS (Cascading Style Sheets) Properties
- 2.4.1. Color
- 2.4.2. Text:
- 2.4.3. Pseudo Classes
- 2.4.4. Transitions
- 2.4.5. Animations
- 2.4.6. Animation of Elements
- 2.4.7. Advanced Animation

2.5. Box Models
- 2.5.1. Height and Width
- 2.5.2. Margin
- 2.5.3. Filling

2.6. Positioning
- 2.6.1. Static Positioning
- 2.6.2. Relative Positioning
- 2.6.3. Absolute Positioning
- 2.6.4. Fixed Positioning
- 2.6.5. Floats

2.7. Adaptive Design
- 2.7.1. Viewport
- 2.7.2. Media Queries
- 2.7.3. CSS Units
- 2.7.4. Images
- 2.7.5. Frameworks

2.8. Modern Layout
- 2.8.1. Flex
- 2.8.2. Grid
- 2.8.3. Flex Vs. Grid

2.9. Pre-Processing
- 2.9.1. Sass
- 2.9.2. Variables
- 2.9.3. Mixins
- 2.9.4. Loops
- 2.9.5. Functions

2.10. System Design
- 2.10.1. Bootstrap
- 2.10.2. Bootstrap Grid
- 2.10.3. Header and Footer of Our Site
- 2.10.4. Forms
- 2.10.5. Cards
- 2.10.6. Modals

Module 3. JavaScript Language Applied to Full Stack Developer

3.1. Primitive Types and Operators
- 3.1.1. JavaScript Language
- 3.1.2. Numbers and Their Operators
- 3.1.3. Text Strings and Their Operators
- 3.1.4. Boolean Values
- 3.1.5. Conversion Between Types

3.2. Flow Controllers and Structure
- 3.2.1. Expressions and Statements
- 3.2.2. Variables and Constants
- 3.2.3. If Statement
- 3.2.4. For, While Statements

3.3. Functions
- 3.3.1. Functions
- 3.3.2. Parameters
- 3.3.3. Functions as Parameters
- 3.3.4. Scope of Variables
- 3.3.5. Nested Scopes
- 3.3.6. Hoisting
- 3.3.7. Closures
- 3.3.8. Recursion

3.4. Data Structures: Objects
- 3.4.1. Object Type
- 3.4.2. Creation of Objects
- 3.4.3. Accessing the Values of an Object
- 3.4.4. Adding or Deleting Properties
- 3.4.5. Nested Objects
- 3.4.6. Destructuring Objects
- 3.4.7. Object Type Methods
- 3.4.8. Spread Operator
- 3.4.9. Immutability

3.5. Data Structures: Array
- 3.5.1. Data Structure Array
- 3.5.2. Array. Typology
- 3.5.3. Nested Arrays
- 3.5.4. Methods of an Array

3.6. OOP: Prototype and Classes
- 3.6.1. OOP: Object-Oriented Programming
- 3.6.2. Prototypes
- 3.6.3. Classes
- 3.6.4. Private Data
- 3.6.5. Subclasses
- 3.6.6. Call and Apply

3.7. JavaScript Types
- 3.7.1. Set
- 3.7.2. WeakSet
- 3.7.3. Map
- 3.7.4. WeakMap
- 3.7.5. Common Expressions

3.8. JavaScript Utilities
- 3.8.1. Date
- 3.8.2. Math
- 3.8.3. Symbol
- 3.8.4. JSON

3.9. JavaScript in the Browser
- 3.9.1. Inclusion of JavaScript in a Website
- 3.9.2. DOM
- 3.9.3. Events
- 3.9.4. Browser Storage

3.10. Asynchronous Programming
- 3.10.1. The Asynchronous Programming
- 3.10.2. Event Loop
- 3.10.3. Call-backs
- 3.10.4. Promises
- 3.10.5. Async/Await

Module 4. Web Layout Applied to Full Stack Developer

4.1. CSS and Layout
- 4.1.1. Layout with Tables
- 4.1.2. Fluid Layout
- 4.1.3. The Responsive Era
- 4.1.4. Mobile First Vs. Desktop First

4.2. CSS and the Rules of Web Design
- 4.2.1. Selectors
- 4.2.2. Pseudo Classes
- 4.2.3. Pseudo Elements

4.3. Layout with CSS
- 4.3.1. Box Model Rules
- 4.3.2. Typographies
- 4.3.3. Colors
- 4.3.4. Images
- 4.3.5. Backgrounds
- 4.3.6. Tables
- 4.3.7. Forms
- 4.3.8. Showing and Hiding Elements
- 4.3.9. CSS Variables

4.4. Responsive Design and Fluid Design
- 4.4.1. Floating Elements
- 4.4.2. CSS Grid
- 4.4.3. Media Queries
- 4.4.4. Flex Box

4.5. The CSS Cascade
- 4.5.1. Priority of CSS Rules
- 4.5.2. Overwriting Rules
- 4.5.3. Classes Vs. Identifiers

4.6. SaSS
- 4.6.1. Software as a Service (SaSS)
- 4.6.2. SaSS Installation
- 4.6.3. Running and Compiling SaSS
- 4.6.4. Structure of a SaSS Directory

4.7. Using SaSS
- 4.7.1. Variables in Sass
- 4.7.2. Modularization of Our Project
- 4.7.3. SaSS Syntax

4.8. SaSS Logic
- 4.8.1. Mixins
- 4.8.2. Maps
- 4.8.3. Functions and Control Structures

4.9. Layout with Bootstrap
- 4.9.1. Bootstrap
- 4.9.2. Bootstrap Layout
- 4.9.3. Forms
- 4.9.4. Box Model with Bootstrap
- 4.9.5. Colors and Fonts
- 4.9.6. Links and Buttons
- 4.9.7. Showing and Hiding Elements with Bootstrap
- 4.9.8. Flex Box with Bootstrap
- 4.9.9. Components

4.10. Theming Bootstrap
- 4.10.1. Rewriting Bootstrap with SaSS (Software as a Service)
- 4.10.2. File Structure
- 4.10.3. Creating our Own CSS Framework (Cascading Style Sheets)

Module 5. Javascript Tools. ReactJS Library

5.1. ReactJS Javascript Tool
- 5.1.1. The ReactJS Tool
- 5.1.2. Create React App
- 5.1.3. JavaScript Syntax Extension

5.2. ReactJS Components
- 5.2.1. Components
- 5.2.2. Props
- 5.2.3. Rendering

5.3. Events in the ReactJS Library
- 5.3.1. Event Handling
- 5.3.2. Inline Event Handling
- 5.3.3. Events in the ReactJS Library

5.4. Configuring ReactJS Hooks
- 5.4.1. Status of a Component
- 5.4.2. Status Hook
- 5.4.3. Hook of Effect
- 5.4.4. Custom Hooks
- 5.4.5. Other Hooks

5.5. Context Component in ReactJS
- 5.5.1. Context Component in ReactJS
- 5.5.2. Using Context
- 5.5.3. Context Structure
- 5.5.4. React Create Context
- 5.5.5. Context. Provider
- 5.5.6. Class. Context Type
- 5.5.7. Context. Consumer
- 5.5.8. Context.displayName
- 5.5.9. Practical Application of Context

5.6. Routing in ReactJs
- 5.6.1. Router
- 5.6.2. React Router
- 5.6.3. Installation
- 5.6.4. Basic Routing
- 5.6.5. Dynamic Routing
- 5.6.6. Primary Components
- 5.6.7. React Router Hooks

5.7. Using Lists and Forms with ReactJS
- 5.7.1. Lists and Loops
- 5.7.2. Forms and Validations
- 5.7.3. Hook React Forms

5.8. Using Styles in ReactJS
- 5.8.1. Traditional Styling
- 5.8.2. Inline Styling
- 5.8.3. Addition of Design System Library

5.9. Performing Tests in JavaScript. Tools
- 5.9.1. Testing
- 5.9.2. Jest JavaScript Testing Framework
- 5.9.3. Visual Testing and Documentation

5.10. Deploying Code with ReactJS
- 5.10.1. Hosting
- 5.10.2. Suppliers
- 5.10.3. Project Preparation
- 5.10.4. Deployment on Heroku

Module 6. JavaScript Framework. Angular

6.1. The Angular Framework and its Architecture
- 6.1.1. Angular CLI
- 6.1.2. Architecture
- 6.1.3. Workspace and Structure
- 6.1.4. Environment

6.2. Angular Framework Components
- 6.2.1. Life Cycle
- 6.2.2. View Encapsulation
- 6.2.3. Interaction Between Components
- 6.2.4. Content Projection

6.3. Angular Framework Templates
- 6.3.1. Text Interpolation
- 6.3.2. Declarations
- 6.3.3. Property Binding
- 6.3.4. Class, Style and Attribute Binding
- 6.3.5. Event Binding and Two-Way Binding
- 6.3.6. Pipes

6.4. Angular Framework Directives
- 6.4.1. Angular Directives
- 6.4.2. Attribute Directives
- 6.4.3. Structure Directives

6.5. Services and Dependency Injection
- 6.5.1. Services
- 6.5.2. Dependency Injection
- 6.5.3. Service Providers

6.6. Routing and Navigation
- 6.6.1. Application with Routing
- 6.6.2. Basic Routing
- 6.6.3. Nested Routes
- 6.6.4. Parameters
- 6.6.5. Access and Authorization
- 6.6.6. Lazy Loading of Modules

6.7. RxJS
- 6.7.1. Observables
- 6.7.2. Observers
- 6.7.3. Subscriptions
- 6.7.4. Operators

6.8. Forms and HTTP
- 6.8.1. Reactive Forms
- 6.8.2. Field Validation
- 6.8.3. Dynamic Forms
- 6.8.4. Requests
- 6.8.5. Interceptors
- 6.8.6. Security

6.9. Animations
- 6.9.1. Transitions and Triggers
- 6.9.2. Path Transitions
- 6.9.3. Differences Between Transitions

6.10. Testing in the Angular Framework
- 6.10.1. Testing Services
- 6.10.2. Component Testing
- 6.10.3. Testing of Directives and Pipelines

Module 7. Programming in NodeJs Language

7.1. NodeJS and its Architecture
- 7.1.1. NPM and Package Management
- 7.1.2. Executing a Program
- 7.1.3. Modules
- 7.1.4. Creating a Module
- 7.1.5. Loop of Events

7.2. Backend, HTTP, Express and Sockets Server
- 7.2.1. Module HTTP
- 7.2.2. Express
- 7.2.3. Socket.io

7.3. Database and Cache
- 7.3.1. MongoDB
- 7.3.2. Mongoose
- 7.3.3. SQL
- 7.3.4. Sequelize
- 7.3.5. Redis

7.4. File System and Os
- 7.4.1. File System Module
- 7.4.2. Os Module
- 7.4.3. Cluster Module

7.5. Events, Buffers and Streams
- 7.5.1. Events
- 7.5.2. Buffers
- 7.5.3. Streams

7.6. Testing
- 7.6.1. Jest
- 7.6.2. Mocha
- 7.6.3. TDD - Cucumber

7.7. Architecture and Good Practices
- 7.7.1. DRY
- 7.7.2. SOLID
- 7.7.3. CRUD
- 7.7.4. MVC
- 7.7.5. Monoliths
- 7.7.6. Microservices
- 7.7.7. Hexagonal Architecture

7.8. Typescript
- 7.8.1. Types, Interfaces and Classes
- 7.8.2. Functions and Modules
- 7.8.3. Generics
- 7.8.4. Namespaces
- 7.8.5. Decorators

7.9. API REST
- 7.9.1. Get
- 7.9.2. Post
- 7.9.3. Put
- 7.9.4. Delete
- 7.9.5. Swagger
- 7.9.6. Building a REST API with Express

7.10. Building and Containerizing an Application with NestJS
- 7.10.1. Nest CLI
- 7.10.2. Docker
- 7.10.3. Building an Application

Module 8. Databases for Full Stack Developers

8.1. Databases for Full Stack Developers
- 8.1.1. Database within Application Development
- 8.1.2. Database Capabilities
- 8.1.3. SQL (Structured Query Language)

8.2. Choice of Database
- 8.2.1. Application or Service to be Considered
- 8.2.2. Database Categories
- 8.2.3. Database Overview

8.3. Development with MySQL
- 8.3.1. Development with MySQL
- 8.3.2. Deployment of Relational Model with MySQL
- 8.3.3. Connection to MySQL

8.4. Development with Oracle Database
- 8.4.1. Development with Oracle DB
- 8.4.2. Model Deployment
- 8.4.3. Connection to Oracle Database

8.5. Development with Oracle SQL Server
- 8.5.1. Oracle SQL Server
- 8.5.2. Model Deployment
- 8.5.3. Connection to SQL Server

8.6. Development with NoSQL
- 8.6.1. Comparison with SQL Databases
- 8.6.2. Database Creation in MongoDB
- 8.6.3. Connection to MongoDB

8.7. Development with Networks
- 8.7.1. Development with Networks
- 8.7.2. Database Creation with Neo4j
- 8.7.3. Connection with Neo4j

8.8. Key-value Database Development
- 8.8.1. Development with K-V Database
- 8.8.2. Database Creation with Redis
- 8.8.3. Connection with Redis

8.9. Databases with Other Data Types
- 8.9.1. Elastic Search
- 8.9.2. In Memory Database
- 8.9.3. Development with Spatial Data

8.10. Database. Advanced Aspects
- 8.10.1. Databases in Cloud Native Development
- 8.10.2. Databases in Microservices Architecture
- 8.10.3. CI/CD and Databases

Module 9. UX CX. Customer Experience

9.1. Customer Experience
- 9.1.1. Customer Experience(CX)
- 9.1.2. New Consumer Needs
- 9.1.3. Feedback in Customer Experience

9.2. Innovative Technologies
- 9.2.1. Thinking Machines
- 9.2.2. New Ways of Sharing Information
- 9.2.3. Measuring What Cannot Be Measured

9.3. Channels of Interaction with the User
- 9.3.1. Customer Analysis
- 9.3.2. Personalization
- 9.3.3. Multiple User Interaction Channels

9.4. User Analytics
- 9.4.1. Web Structure
- 9.4.2. User Analytics
- 9.4.3. Advanced User Analytics

9.5. Nielsen and its Impact on CX
- 9.5.1. Nielsen and its Impact on CX
- 9.5.2. User Testing Techniques

9.6. Customer Experience Tools
- 9.6.1. Advanced Tools
- 9.6.2. Mobility
- 9.6.3. Accessibility

9.7. New Methodologies
- 9.7.1. The User's Challenge
- 9.7.2. UX Process
- 9.7.3. User Research

9.8. Communication of a Design
- 9.8.1. Wireframing
- 9.8.2. Design Communication Tools
- 9.8.3. Advanced Design Communication Tools

9.9. UI design
- 9.9.1. UI design
- 9.9.2. Web and Mobile Interfaces
- 9.9.3. Web and Mobile Components

9.10. Elaboration of a CX
- 9.10.1. Elaboration of a CX
- 9.10.2. Design of New Experiences
- 9.10.3. Interfaces

Module 10. Continuous Integration and Application Deployment

10.1. Continuous Integration and Continuous Deployment: CI/CD
- 10.1.1. Use of Continuous Integration and Continuous Deployment (CI/CD)
- 10.1.2. Differences Between Continuous Integration and Continuous Deployment (CI/CD)
- 10.1.3. Continuous Integration and Continuous Deployment. Benefits of CI/CD

10.2. New Development Paradigms
- 10.2.1. From Waterfall to DevOps
- 10.2.2. Style Guide: The 12 Factors
- 10.2.3. Cloud Native, Microservices and Serverless

10.3. DevOps, Beyond CI/CD
- 10.3.1. DevOps
- 10.3.2. DevOps. Continuous Everything
- 10.3.3. DevOps vs SRE

10.4. Container Technology I - Docker
- 10.4.1. Containers Contribution
- 10.4.2. Docker. Architecture
- 10.4.3. Deployment Process with Docker

10.5. Container Technology II - Kubernetes
- 10.5.1. Orchestration
- 10.5.2. Kubernetes
- 10.5.3. The Kubernetes Ecosystem

10.6. Infrastructure Configuration with GitOps
- 10.6.1. Immutable Infrastructure
- 10.6.2. GitOps
- 10.6.3. GitOps Tools

10.7. Pipelines and Automation. CI/CD Use Cases
- 10.7.1. Continuous Integration
- 10.7.2. Continuous Deployment and Delivery
- 10.7.3. Automatic Validation
- 10.7.4. Best Practices in CI/CD

10.8. CI/CD with Jenkins. Reference
- 10.8.1. CI/CD with Jenkins
- 10.8.2. Jenkins Pipelines
- 10.8.3. Best Practices with Jenkins

10.9. CI/CD Ecosystem
- 10.9.1. Ecosystem Organization
- 10.9.2. Advanced Tools
- 10.9.3. Dagger. The Future

10.10. Final Phases of the CI/CD Oriented Software Cycle
- 10.10.1. Application of IA to the CI/CD Process
- 10.10.2. DevSecOps
- 10.10.3. Chaos Engineering

You will have at your disposal the most modern educational resources, with free access to the virtual classroom 24 hours a day"

06

Methodology

This academic program offers students a different way of learning. Our methodology follows a cyclical learning process: **Relearning.**
This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

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

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

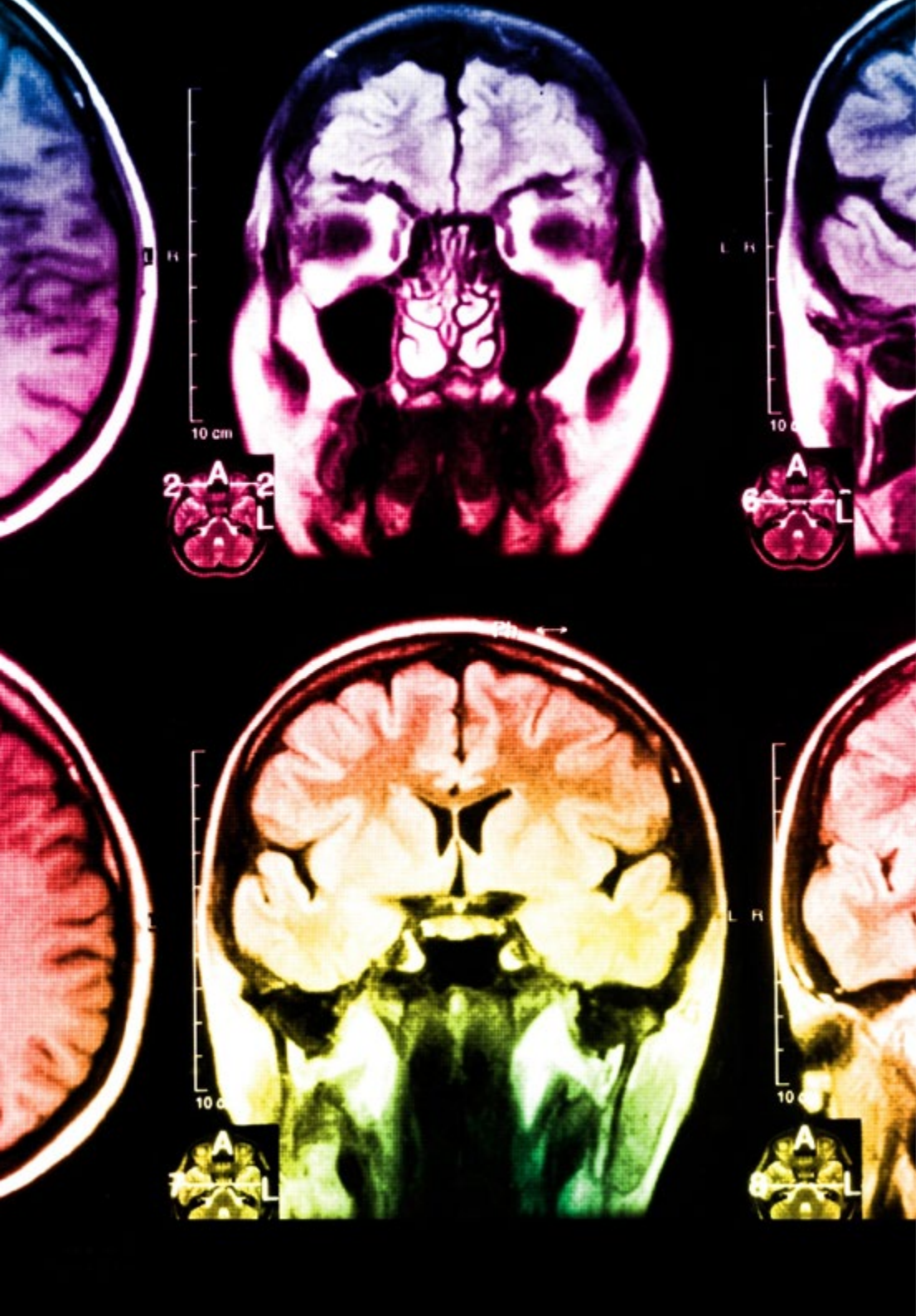
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

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

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

Study Material

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

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.

Classes

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

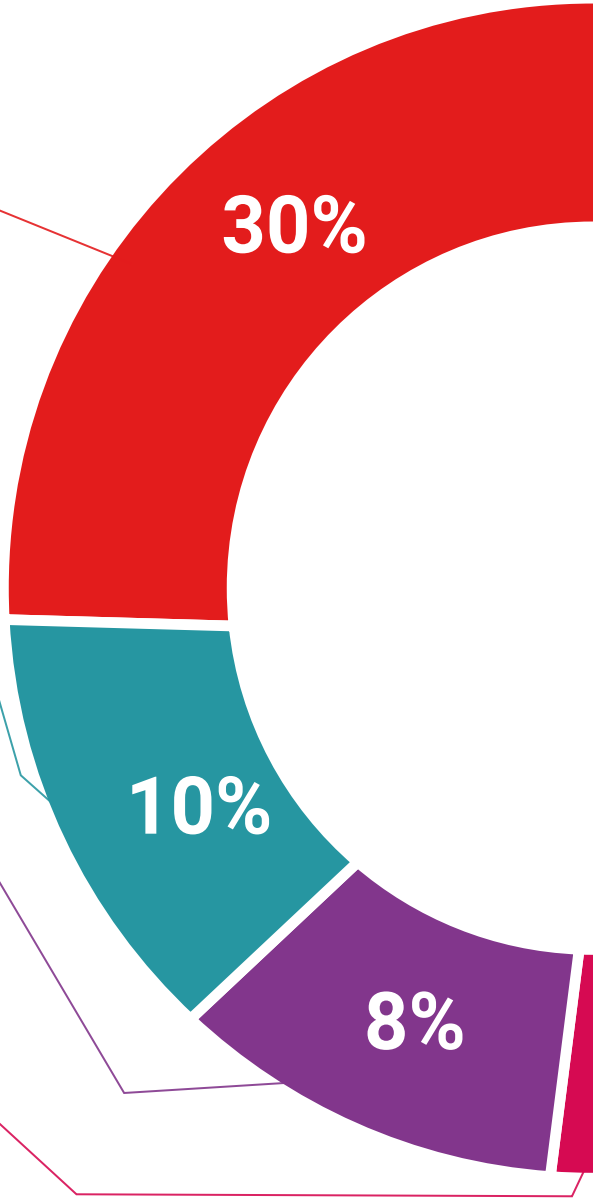
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

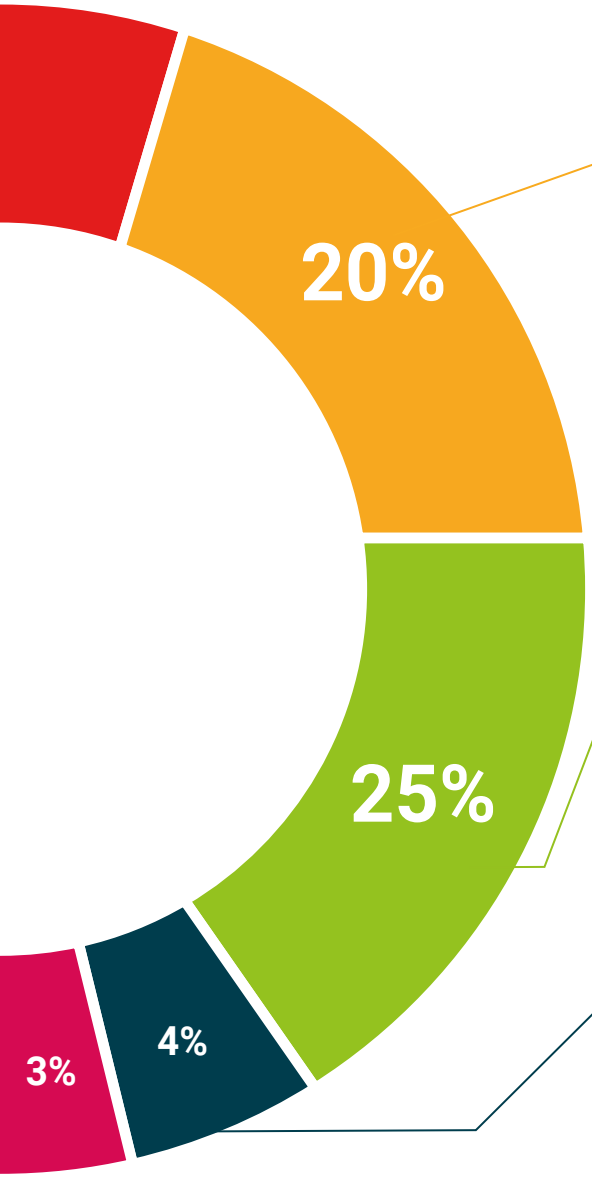
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

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

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

07

Certificate

The Professional Master's Degree in Full Front-End Programming Stack Developer guarantees students, in addition to the most rigorous and up-to-date education, access to a Professional Master's Degree issued by TECH Technological University.

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

This **Professional Master's Degree in Full Front-End Programming Stack Developer** contains the most complete and up-to-date program on the market.

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

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

Title: **Professional Master's Degree in Full Front-End Programming Stack Developer**

Official Nº of hours: **1,500 h.**


tech technological university

Awards the following

CERTIFICATE

to

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

For having successfully passed and accredited the following program

PROFESSIONAL MASTER'S DEGREE

in

Full Front-End Programming Stack Developer

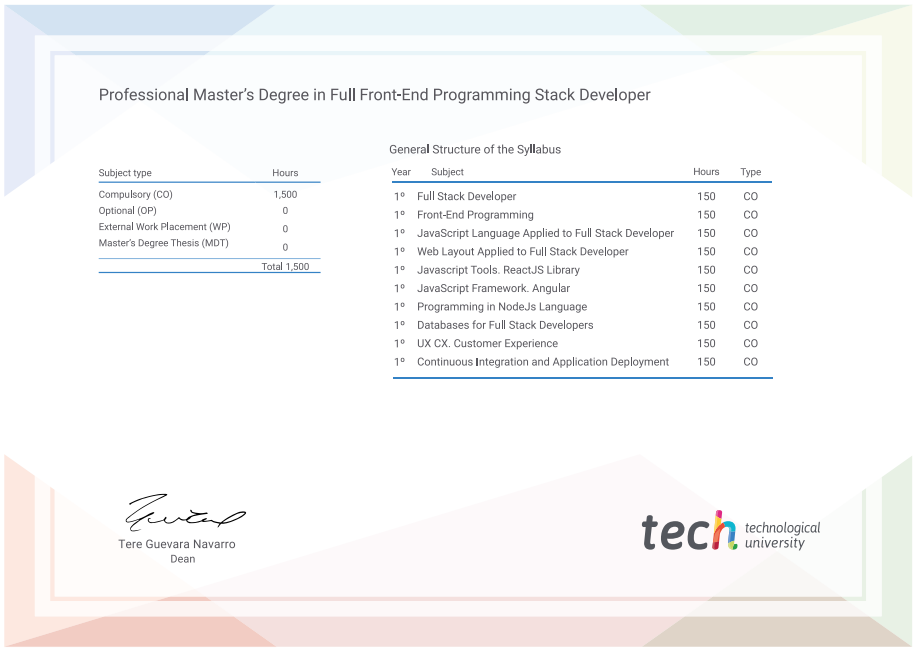
This is a qualification awarded by this University, equivalent to 1,500 hours, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

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

June 17, 2020

Tere Guevara Navarro
Dean

Unique TECH Code: AFWORD23S techtitute.com/certificates


Professional Master's Degree in Full Front-End Programming Stack Developer

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

General Structure of the Syllabus

Year	Subject	Hours	Type
1º	Full Stack Developer	150	CO
1º	Front-End Programming	150	CO
1º	JavaScript Language Applied to Full Stack Developer	150	CO
1º	Web Layout Applied to Full Stack Developer	150	CO
1º	Javascript Tools. ReactJS Library	150	CO
1º	JavaScript Framework. Angular	150	CO
1º	Programming in NodeJs Language	150	CO
1º	Databases for Full Stack Developers	150	CO
1º	UX CX. Customer Experience	150	CO
1º	Continuous Integration and Application Deployment	150	CO

Tere Guevara Navarro
Dean

tech technological university

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

tech technological university

Professional Master's Degree
Full Front-End Programming Stack Developer

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

Professional Master's Degree

Full Front-End Programming Stack Developer

tech technological university