

# Máster Semipresencial

## Programación Front End

### Full Stack Developer



## Máster Semipresencial

### Programación Front End

### Full Stack Developer

Modalidad: Semipresencial (Online + Prácticas)

Duración: 12 meses

Titulación: TECH Global University

Créditos: 60 + 4 ECTS

Acceso web: [www.techtitude.com/informatica/master-semipresencial/master-semipresencial-programacion-front-end-full-stack-developer](http://www.techtitude.com/informatica/master-semipresencial/master-semipresencial-programacion-front-end-full-stack-developer)

# Índice

01

Presentación

---

pág. 4

02

¿Por qué cursar este Máster Semipresencial?

---

pág. 8

03

Objetivos

---

pág. 12

04

Competencias

---

pág. 18

05

Dirección del curso

---

pág. 22

06

Estructura y contenido

---

pág. 26

07

Prácticas

---

pág. 38

08

¿Dónde puedo hacer las Prácticas?

---

pág. 44

09

Metodología de estudio

---

pág. 48

10

Titulación

---

pág. 58

# 01 Presentación

Las empresas buscan desarrolladores que no solo tengan la capacidad de crear interfaces de usuario atractivas y funcionales, sino que también dominen el desarrollo de servidores y bases de datos. Con el auge de tecnologías como React, Node.js y la creciente importancia de la experiencia de usuario (UX), se pueden integrar múltiples capas de desarrollo web son altamente valorados. De hecho, con la evolución constante de las metodologías ágiles y la automatización de Procesos, los *Full Stack Developers* deben estar siempre al día. En este contexto, TECH ha desarrollado este exhaustivo programa, que se divide en dos áreas, una primera totalmente online, basada en la innovadora metodología *Relearning*, y una estancia práctica de 3 semanas en una reconocida empresa.





“

*Gracias a este Máster Semipresencial, dominarás las técnicas de programación, adquiriendo también habilidades en gestión de proyectos, experiencia de usuario y despliegue continuo”*

La demanda de profesionales en Programación Front End Full Stack Developer sigue en aumento, impulsada por la transformación digital acelerada en múltiples sectores. Con la proliferación de aplicaciones web y móviles, las empresas buscan desarrolladores que puedan manejar tanto el *front-end* como el *back-end*.

Así nace este Máster Semipresencial, que se dividirá en dos secciones diferenciadas. Primero, en una parte teórica que proporcionará una visión general de las tecnologías y habilidades necesarias para el desarrollo completo de aplicaciones web, incluyendo un enfoque en el *front-end* en Programación, donde los informáticos se enfocarán en las técnicas y mejores prácticas para crear interfaces de usuario eficientes y atractivas.

Asimismo, el plan de estudios se adentrará en el lenguaje JavaScript aplicado a *Full Stack Developer*, crucial para todo desarrollador moderno, así como la maquetación web aplicada, donde los profesionales estructurarán y estilizarán páginas web de manera profesional. Además, las herramientas de JavaScript y las librerías específicas, como ReactJs, serán indispensables para el desarrollo de aplicaciones dinámicas y escalables.

Igualmente, se profundizará en la programación en lenguaje NodeJS, así como en las bases de datos para Desarrolladores *Full Stack*, incluyendo la lógica del servidor y la gestión de bases de datos. También se abordará la importancia de la experiencia del usuario, UX CX, y se concluirá con la integración continua y el despliegue de aplicaciones. Finalmente, la segunda parte de la capacitación consistirá en una estancia práctica exhaustiva, de 3 semanas de duración, en una de las mejores empresas informáticas, cuidadosamente seleccionadas para elevar la carrera del egresado al siguiente nivel.

De este modo, TECH ha diseñado un programa completamente en línea y totalmente adaptable, que solo precisará de un dispositivo electrónico con conexión a Internet para acceder a los materiales didácticos. Adicionalmente, el área teórica se basará en la revolucionaria metodología *Relearning*, pionera en esta universidad, y consistente de la repetición de conceptos clave para una asimilación óptima y orgánica de los contenidos.

Este **Máster Semipresencial en Programación Front End Full Stack Developer** contiene el programa más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ Desarrollo de más de 100 casos prácticos presentados por profesionales informáticos expertos en Programación Front End Full Stack Developer y profesores universitarios de amplia experiencia en el campo
- ♦ Sus contenidos gráficos, esquemáticos y eminentemente prácticos con los que están concebidos, recogen una información imprescindible sobre aquellas herramientas indispensables para el ejercicio profesional
- ♦ Todo esto se complementará con lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ♦ Disponibilidad de los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet
- ♦ Además, podrás realizar una estancia de prácticas en una de las mejores empresas



*Profundizarás en la creación de interfaces de usuario, con técnicas avanzadas de HTML, CSS y JavaScript para diseñar experiencias de usuario intuitivas y responsivas. ¿A qué esperas para matricularte?”*

“

*Te centrarás en el lenguaje JavaScript aplicado a Full Stack Developer, dominando JavaScript, la columna vertebral de la mayoría de las aplicaciones web modernas. ¡Con todas las garantías de calidad!”*

En esta propuesta de Máster, de carácter profesionalizante y modalidad semipresencial, el programa está dirigido a la actualización de profesionales de la informática que desarrollan sus funciones en empresas especializadas en desarrollo de software, y que requieren un alto nivel de cualificación. Los contenidos están basados en la última evidencia científica, y orientados de manera didáctica para integrar el saber teórico en la práctica informática, y los elementos teórico-prácticos facilitarán la actualización del conocimiento y permitirán la toma de decisiones.

Gracias a su contenido multimedia elaborado con la última tecnología educativa, permitirán al profesional de la programación un aprendizaje situado y contextualizado, es decir, un entorno simulado que proporcionará un aprendizaje inmersivo programado para entrenarse ante situaciones reales. El diseño de este programa está basado en el Aprendizaje Basado en Problemas, mediante el cual deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del mismo. Para ello, contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.

*Realizarás una estancia intensiva de 3 semanas en una empresa de prestigio y adquirirás todo el conocimiento para crecer personal y profesionalmente.*

*Analizarás los principios del diseño centrado en el usuario, mejorando la experiencia del cliente, a través de los mejores materiales didácticos del mercado académico, a la vanguardia tecnológica y educativa.*







“

*Tendrás la oportunidad de aprender a través de una metodología mixta, que incluye clases en línea y una estancia práctica innovadora, facilitando una experiencia de aprendizaje más dinámica y práctica”*

### 1. Actualizarse a partir de la última tecnología disponible

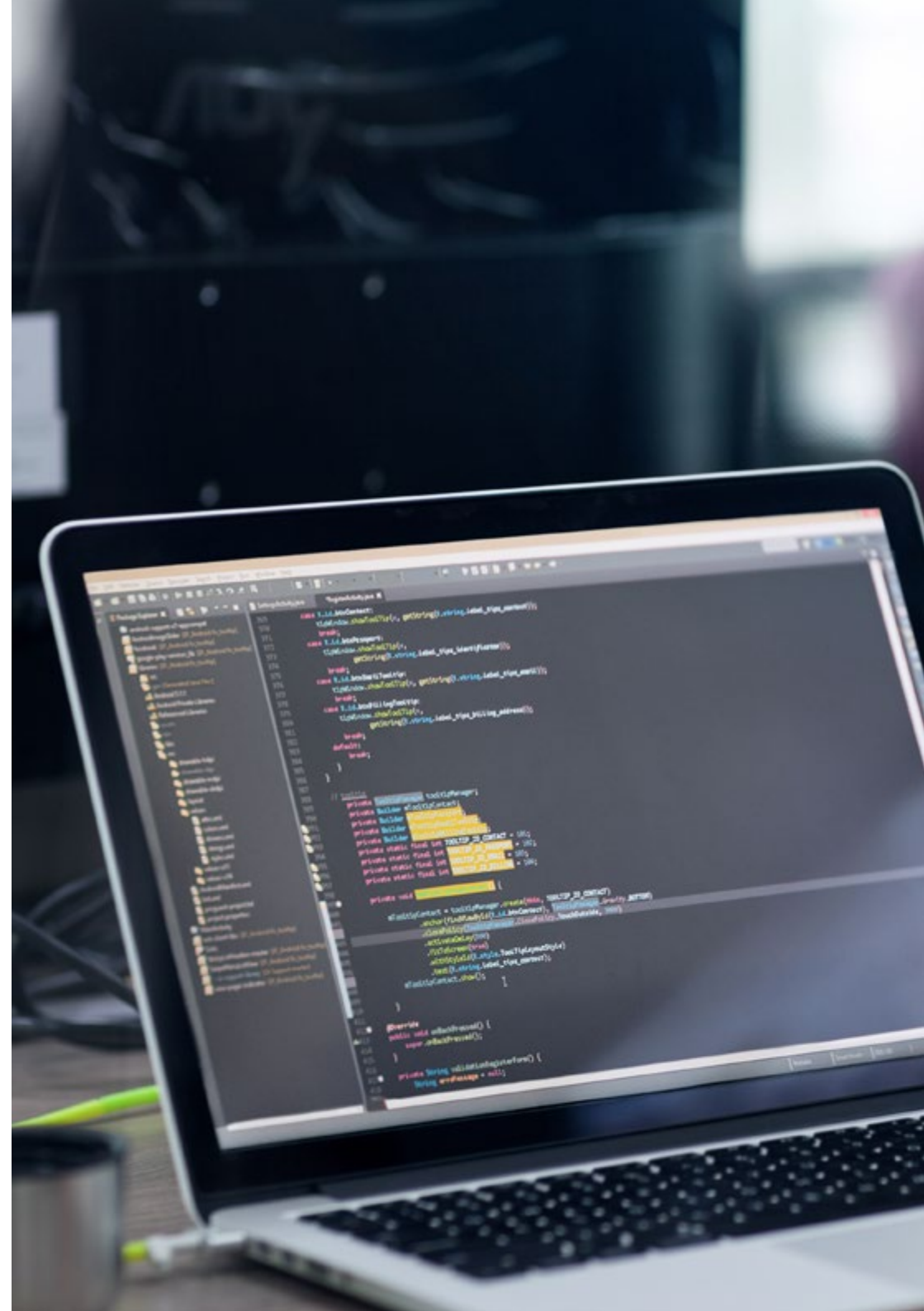
La última tecnología en Programación Front End Full Stack Developer se caracteriza por la integración de herramientas y *frameworks* avanzados que facilitan el desarrollo de aplicaciones web más rápidas, seguras y escalables. En el *front-end*, *frameworks* como React y Angular siguen evolucionando, ofreciendo mejoras en rendimiento y funcionalidad. Por su lado, en el *back-end*, Node.js continúa siendo una herramienta fundamental. Estas innovaciones no solo mejoran la productividad de los desarrolladores, sino que también elevan el estándar de las aplicaciones web en términos de rendimiento y usabilidad.

### 2. Profundizar a partir de la experiencia de los mejores especialistas

El amplio equipo de profesionales que acompañará al especialista a lo largo de todo el periodo práctico supone un aval de primera y una garantía de actualización sin precedentes. Con un tutor designado específicamente, el informático podrá ver pacientes reales en un entorno de vanguardia, lo que le permitirá incorporar en su práctica diaria los procedimientos y herramientas más efectivas en Programación Front End Full Stack Developer.

### 3. Adentrarse en entornos profesionales de primera

TECH selecciona minuciosamente todos los centros disponibles para las Capacitaciones Prácticas. Gracias a ello, el especialista tendrá garantizado el acceso a un entorno clínico de prestigio en el área de la Programación Front End Full Stack Developer. De esta manera, podrá comprobar el día a día de un área de trabajo exigente, rigurosa y exhaustiva, aplicando siempre las últimas tesis y postulados científicos en su metodología de trabajo.



#### 4. Combinar la mejor teoría con la práctica más avanzada

El mercado académico está plagado de programas pedagógicos poco adaptados al quehacer diario del especialista y que exigen largas horas de carga lectiva, muchas veces poco compatibles con la vida personal y profesional. TECH ofrece un nuevo modelo de aprendizaje, 100% práctico, que permite ponerse al frente de procedimientos de última generación en el campo de la Programación Front End Full Stack Developer y, lo mejor de todo, llevarlo a la práctica profesional en tan solo 3 semanas.

#### 5. Abrir la puerta a nuevas oportunidades

Los desarrolladores *Full Stack*, con su capacidad para manejar tanto el *front-end* como el *back-end* de una aplicación, son altamente valorados por su versatilidad y eficiencia en la creación de soluciones integrales. Este dominio completo permite trabajar en una variedad de proyectos, desde startups innovadoras, hasta grandes corporaciones tecnológicas, y adaptarse a diferentes roles, ya sea como desarrollador, arquitecto de software o líder técnico. Así, con el creciente enfoque en la digitalización y la transformación digital, los desarrolladores *Full Stack* están en una posición privilegiada para aprovechar estas oportunidades y avanzar en sus carreras.



*Tendrás una inmersión práctica total en el centro que tú mismo elijas*

# 03

## Objetivos

Este programa universitario buscará dotar a los informáticos de un conocimiento profundo y práctico de las tecnologías más avanzadas en programación web, como HTML, CSS, JavaScript, React, Angular, Node.js y bases de datos relacionales y no relacionales. Además, se desarrollarán habilidades en metodologías ágiles, gestión de proyectos, y herramientas de integración y despliegue continuo, garantizando que los egresados puedan adaptarse a las demandas dinámicas de la industria tecnológica. Otro objetivo clave será inculcar un enfoque centrado en el usuario, a través de los principios de UX y CX para crear experiencias digitales excepcionales.





“

*Al finalizar este Máster Semipresencial, estarás preparado para afrontar con éxito los retos del mercado laboral, destacando por tu versatilidad, capacidad de innovación y habilidades técnicas avanzadas”*



### Objetivo general

---

- ♦ El objetivo general que tiene el Máster Semipresencial en Programación Front End Full Stack Developer será proporcionar un conocimiento especializado en aspectos clave de la programación, enfocado en los fundamentos esenciales de la web, incluyendo la correcta sintaxis de los lenguajes HTML y CSS. Además, se capacitará a los profesionales para desarrollar cualquier tipo de aplicación utilizando JavaScript, incluyendo el uso de la librería Bootstrap y la implementación de proyectos de maquetación con SaSS. Asimismo, se desarrollarán habilidades avanzadas en el uso de *frameworks* modernos, como React y Angular, explorando el potencial del lenguaje NodeJS



*Gestionarás bases de datos relacionales y no relacionales, esenciales para el almacenamiento y manipulación de datos, de la mano de la mejor universidad digital del mundo, según Forbes: TECH"*





## Objetivos específicos

---

### **Módulo 1. Desarrollo *Full Stack Developer***

- ♦ Desarrollar conocimiento avanzado en Programación
- ♦ Fomentar el uso de sistemas de control de versiones y plataformas de alojamiento de código
- ♦ Promover el uso de metodologías ágiles
- ♦ Profundizar en los conceptos clave y funcionamiento del Internet
- ♦ Aumentar la destreza en la línea de comandos

### **Módulo 2. *Front-end* en Programación**

- ♦ Identificar y comprender la sintaxis correcta de HTML y CSS
- ♦ Explorar los distintos elementos del HTML
- ♦ Determinar el enfoque de diseño adaptativo
- ♦ Emplear formato de presentación a las páginas web aplicando hojas estilo cascadas
- ♦ Incorporar preprocesador de CSS
- ♦ Establecer los beneficios de la utilización de un preprocesador
- ♦ Generar conocimiento especializado sobre sistemas de diseño
- ♦ Establecer criterio de utilización de sistema de diseño

### Módulo 3. Lenguaje JavaScript aplicado a *Full Stack Developer*

- ♦ Establecer los tipos básicos y complejos que ofrece JavaScript
- ♦ Analizar las diferentes formas de programar con el lenguaje y hacer un uso correcto en cada situación
- ♦ Actualizar los conocimientos a las últimas versiones
- ♦ Descubrir la programación funcional
- ♦ Examinar la programación asíncrona y sus características

### Módulo 4. Maquetación Web aplicada a *Full Stack Developer*

- ♦ Evaluar un diseño web para saber situarlo temporalmente
- ♦ Examinar las principales reglas CSS
- ♦ Presentar diferentes metodologías CSS para obtener diseños *responsive*
- ♦ Fundamentar los principios de desarrollo en cascada CSS
- ♦ Identificar la tecnología *Bootstrap* en cualquier diseño web
- ♦ Analizar los principios de *Bootstrap*
- ♦ Desarrollar una maqueta web usando *Bootstrap*
- ♦ Determinar los principios de desarrollo en un proyecto SaSS

### Módulo 5. Herramientas de JavaScript. Librería ReactJs

- ♦ Determinar las funcionalidades de React
- ♦ Configurar un proyecto utilizando Create React App
- ♦ Analizar el ciclo de vida de los componentes en React
- ♦ Generar conocimiento especializado sobre las funcionalidades modernas de React, como *Hooks* y *Context*
- ♦ Setear estados globales utilizando contexto
- ♦ Crear y renderizar listas y crear formularios con React
- ♦ Implantar validación de campos en los formularios
- ♦ Estilizar componentes y elementos
- ♦ Debuggear, testear y desplegar aplicaciones React

### Módulo 6. *Framework* de JavaScript. Angular

- ♦ Desarrollar conocimiento especializado sobre la arquitectura del *framework*
- ♦ Profundizar en la metodología de angular
- ♦ Analizar el concepto de componentes
- ♦ Organizar el código correctamente



### Módulo 7. Programación en lenguaje NodeJS

- ♦ Generar conocimiento especializado sobre los tipos de JavaScript y sus operadores
- ♦ Analizar las mejores maneras de programar con el lenguaje
- ♦ Actualizar los conocimientos a las últimas versiones
- ♦ Explorar la programación funcional
- ♦ Desarrollar la programación asíncrona y su motivación
- ♦ Adquirir la capacidad de realizar una aplicación con Node.js

### Módulo 8. Bases de datos para Desarrolladores Full Stack

- ♦ Determinar por qué usar una base de datos en el desarrollo de aplicaciones
- ♦ Examinar los tipos de bases de datos disponibles y sus diferencias
- ♦ Desarrollar una idea clara de para qué usar cada tipo de base de datos
- ♦ Analizar el uso de base de datos en los actuales paradigmas de desarrollo

### Módulo 9. UX CX. Experiencia de Cliente

- ♦ Analizar la importancia del usuario en la actualidad y profundizar en la cultura de *feedback*
- ♦ Concretar estrategias de omnicanalidad y personalización basada en microinteracciones
- ♦ Estudiar la evolución de la analítica web hasta la analítica del comportamiento

- ♦ Determinar cómo la Inteligencia Artificial ha llevado la CX al siguiente nivel
- ♦ Establecer las técnicas más importantes de análisis de experiencia web, movilidad y accesibilidad
- ♦ Presentar la metodología *Design Thinking* y el proceso de creación de experiencia de usuario
- ♦ Plantear herramientas concretas de prototipado y *wireframing*, así como *frameworks* de desarrollo de *front-end*

### Módulo 10. Integración continua y despliegue de Aplicaciones

- ♦ Concretar los beneficios de adoptar un modelo automatizado de despliegue de aplicaciones
- ♦ Establecer las diferencias entre integración continua, entrega y despliegue continuos
- ♦ Determinar las principales características de DevOps
- ♦ Evaluar algunas de las herramientas fundamentales para implementar pipelines de CI/CD
- ♦ Desarrollar los factores esenciales para desarrollar aplicaciones preparadas para soportar los procesos CI/CD
- ♦ Examinar las tecnologías de contenedores como pilar fundamental de la práctica de CI/CD

# 04

## Competencias

Entre las competencias clave de esta titulación académica, se incluirán la capacidad para diseñar y desarrollar aplicaciones web completas, utilizando tecnologías *front-end* y *back-end* avanzadas, como HTML, CSS, JavaScript, React, Angular y Node.js. Así, los informáticos también adquirirán habilidades en la implementación de prácticas UX/UI para mejorar la experiencia del usuario y en la integración de bases de datos para la gestión eficiente de datos en aplicaciones. Además, se enfatizará en la utilización de metodologías ágiles y herramientas de CI/CD, para asegurar la entrega continua y la mejora constante de los productos digitales.

```
...ings.js
...l.js
JS command.js
JS editor.js
JS fileManager.js
JS main.js
JS readme.rst
JS sequences.js
```

```
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
```

```
self
input.on
//escap
if (e.
// th
self.
retur
}
if (e.
e.sto
e.pro
self.
self.
retur
}
//up/d
if (e.
e.pro
e.sto
if (
//
if
```

```
searchHistory;  
= this;  
("keydown", function(e) {  
  if (e.keyCode == 27) {  
    // this is a new line  
    .deactivate(true);  
    return; // this is a modified line
```

```
keyCode == 13) {  
  .stopImmediatePropagation();  
  .eventDefault();  
  .search();  
  .deactivate();  
  return;
```

```
own  
keyCode == 38 || e.keyCode == 40  
eventDefault();  
stopImmediatePropagation();  
e.keyCode == 38 || e.keyCode == 40  
show
```

“

*Indagarás en las prácticas de DevOps, asegurando la implementación y el mantenimiento de las aplicaciones de manera eficiente y segura, gracias a una amplia biblioteca de recursos multimedia”*



## Competencias generales

---

- ♦ Reconocer de forma correcta la sintaxis de los lenguajes HTML y CSS
- ♦ Desarrollar criterio de buenas prácticas de desarrollo web
- ♦ Generar conocimiento especializado sobre el lenguaje JavaScript
- ♦ Ser capaz de desarrollar cualquier tipo de aplicación con JavaScript
- ♦ Analizar la librería Bootstrap
- ♦ Llevar a cabo proyectos de maquetación con SaSS (*Syntactically Awesome Stylesheets*)
- ♦ Identificar la sintaxis de React y cómo programar usándola
- ♦ Aplicar buenas prácticas al lenguaje
- ♦ Examinar el proceso de carga y acceso en cada uno de los tipos de base de datos líderes en su área
- ♦ Evaluar las herramientas y técnicas más importantes en el análisis de CX y el 'stack tecnológico' habitual en las empresas







## Competencias específicas

---

- ♦ Analizar las diferentes estructuras de datos
- ♦ Examinar las técnicas de diseño e interpretación de algoritmos
- ♦ Preparar el entorno de desarrollo
- ♦ Clonar un sitio web
- ♦ Generar un sitio web con Bootstrap
- ♦ Compilar código CSS con SaSS
- ♦ Desarrollar un *framework* CSS propio basado en Bootstrap mediante SaSS
- ♦ Generar un proyecto y ponerlo en marcha
- ♦ Establecer cómo conectarse y cargar/extraer datos desde diferentes tipos de bases de datos
- ♦ Identificar prácticas, casos de uso, tecnologías y herramientas del ecosistema de CI/CD, imprescindibles para dar soporte al proceso en su conjunto

# 05

## Dirección del curso

El equipo docente detrás del Máster Semipresencial en Programación Front End Full Stack Developer son profesionales expertos y con una amplia experiencia en el campo del desarrollo web. De hecho, están comprometidos con proporcionar una enseñanza de alta calidad y actualizada, basada en su conocimiento profundo de tecnologías como HTML, CSS, JavaScript, React, Angular, Node.js, y diversas bases de datos. Además de su experiencia técnica, son hábiles en la aplicación de metodologías ágiles y en la gestión de proyectos, asegurando que los egresados adquieran, tanto habilidades prácticas, como una comprensión sólida de los principios fundamentales del desarrollo *Full Stack*.



“

*La dedicación del claustro no solo radica en transmitir conocimientos teóricos y prácticos, sino también en guiarte hacia la excelencia profesional, preparándote para sobresalir en el entorno laboral”*



## Dirección



### D. Olalla Bonal, Martín

- ♦ Gerente Senior de Práctica de *Blockchain* en EY
- ♦ Especialista Técnico Cliente *Blockchain* para IBM
- ♦ Director de Arquitectura para Blocknitive
- ♦ Coordinador de Equipo en Bases de Datos Distribuidas no Relacionales para WedoIT, Subsidiaria de IBM
- ♦ Arquitecto de Infraestructuras en Bankia
- ♦ Responsable del Departamento de Maquetación en T-Systems
- ♦ Coordinador de Departamento para Bing Data España SL

## Profesores

### D. Calzada Martínez, Jesús

- ♦ Ingeniero de Software Senior en Devo
- ♦ Desarrollador *Full Stack* en Blocknitive
- ♦ Responsable de *Frontend* en Infinia
- ♦ Desarrollador *Full Stack* en Resem
- ♦ Desarrollador Java en Hitec
- ♦ Graduado en Ingeniería Informática por la Universidad Complutense de Madrid

### D. Frias Favero, Pedro Luis

- ♦ Arquitecto Líder de Blockchain en EY
- ♦ Cofundador y Director Técnico de Swear IT Technologies
- ♦ Director de Soporte IT para Key Business One. México, Colombia y España
- ♦ Licenciado en Ingeniería Industrial por la Universidad Yacambú
- ♦ Experto en Blockchain y Aplicaciones Descentralizadas por la Universidad de Alcalá



**D. Reyes Oliva, Luis**

- ♦ Arquitecto de *Software y Cloud*
- ♦ Promotor de desarrollo y Arquitecto *Cloud* en IBM
- ♦ Responsable Técnico de Clientes para cuentas integradas de BBVA en IBM
- ♦ Arquitecto de *Cloud y DevOps* en IBM
- ♦ Arquitecto de *Software* para clientes en Telefónica
- ♦ Arquitecto de soluciones técnicas para Rational
- ♦ *Software Engineering Manager* en Borland
- ♦ *Project Manager* en Altana Consulting
- ♦ Licenciado en Ingeniería Informática por la Universidad Pontificia de Salamanca

**D. Pintado San Claudio, Bruno**

- ♦ Coordinador de Desarrollo en iDavinci
- ♦ Desarrollador Java en la Biblioteca Nacional de España
- ♦ Desarrollador de Apoyo y Técnico de Redes N1 en Sanitas
- ♦ Técnico de Apoyo de Sistemas en el Ayuntamiento de Alcobendas
- ♦ Técnico de comunicaciones N1 para ADIF en el Centro de Telecomunicaciones de Atocha
- ♦ Graduado en Ingeniería Técnica de Telecomunicaciones con Especialidad en Sistemas Electrónicos por la Universidad Politécnica de Valencia
- ♦ Graduado en Ingeniería de Electrónica de Comunicaciones por la Universidad Politécnica de Madrid

**D. Gómez Rodríguez, Antonio**

- ♦ Ingeniero Principal de Soluciones Cloud para Oracle
- ♦ Coorganizador de Málaga Developer Meetup
- ♦ Consultor Especialista para Sopra Group y Everis
- ♦ Líder de equipos en System Dynamics
- ♦ Desarrollador de Softwares en SGO Software
- ♦ Máster en E-Business por la Escuela de Negocios de La Salle
- ♦ Postgrado en Tecnologías y Sistemas de Información por el Instituto Catalán de Tecnología
- ♦ Licenciado en Ingeniería Superior de Telecomunicación por la Universidad Politécnica de Cataluña

**D. Guerrero Díaz-Pintado, Arturo**

- ♦ Director de Experiencia de Clientes para IBM
- ♦ Ingeniero Técnico de Preventa a través de *Watson Customer Engagement* portfolio
- ♦ Ingeniero de redes de I+D en Telefónica
- ♦ Licenciado en Ingeniería Superior de Telecomunicación por la Universidad de Alcalá y la *Danish Technical University*

06

# Estructura y contenido

Los contenidos del programa abarcarán, desde los fundamentos esenciales, como HTML, CSS y JavaScript, hasta tecnologías avanzadas, como React, Angular y Node.js, para el desarrollo *front-end* y *back-end* de aplicaciones web. Así, se profundizará en la maquetación web, el diseño de interfaces y la implementación de prácticas de UX/UI para optimizar la experiencia del usuario. Además, se analizarán técnicas de integración continua y despliegue de aplicaciones (CI/CD), así como la gestión eficiente de bases de datos y la aplicación de metodologías ágiles para la gestión de proyectos.

```
...se if (settings[0]  
if (name.compare  
name += "  
}  
name += Date  
if (set
```

“

*El Máster Semipresencial en Programación Front End Full Stack Developer abordará una amplia gama de contenidos, diseñados para proporcionarte una capacitación completa en desarrollo web moderno”*

## Módulo 1. Desarrollo *Full Stack Developer*

- 1.1. Desarrollo *Full Stack Developer* I. Programación y lenguajes
  - 1.1.1. Programación
  - 1.1.2. Roles de programación
  - 1.1.3. Lenguajes y *framework*
  - 1.1.4. Algoritmo
  - 1.1.5. Características de un algoritmo
- 1.2. Desarrollo *Full Stack Developer* II. Tipología
  - 1.2.1. Variables y constantes
  - 1.2.2. Tipos
  - 1.2.3. Operadores
  - 1.2.4. Declaraciones
  - 1.2.5. Bucles
  - 1.2.6. Funciones y objetos
- 1.3. Estructura de datos en el desarrollo
  - 1.3.1. Tipos de estructura lineales
  - 1.3.2. Tipos de estructura funcionales
  - 1.3.3. Tipos de estructuras de árbol
- 1.4. Diseño e interpretación de algoritmos
  - 1.4.1. Paralelismo en desarrollo. Divide y vencerás
  - 1.4.2. Algoritmos voraces
  - 1.4.3. Programación dinámica
- 1.5. Entorno y herramientas para el desarrollo orientado a *Full Stack Developer*
  - 1.5.1. Preparación del entorno para Mac OS
  - 1.5.2. Preparación del entorno para Linux
  - 1.5.3. Preparación del entorno para Windows
- 1.6. Línea de comando. Tipología y funcionamiento
  - 1.6.1. La Terminal
  - 1.6.2. Emuladores
  - 1.6.3. Intérprete de Comandos
  - 1.6.4. Primeros comandos
  - 1.6.5. Navegación
  - 1.6.6. Gestión de archivos y carpetas utilizando la Interfaz de Línea de Comandos
  - 1.6.7. *Secure Shell*. SSH
  - 1.6.8. Comandos avanzados

- 1.7. Git. Repositorio de software
  - 1.7.1. Git Repositorio de software
  - 1.7.2. Uso de Git
  - 1.7.3. Repositorios de software
  - 1.7.4. Ramas
  - 1.7.5. Ciclo de trabajo
  - 1.7.6. Comandos
- 1.8. Servicio de alojamiento de versionado de código
  - 1.8.1. Servicio de alojamiento de versionado de código
  - 1.8.2. Proveedores
  - 1.8.3. Repositorios
- 1.9. Internet
  - 1.9.1. Internet
  - 1.9.2. Protocolos utilizados en WWW
  - 1.9.3. Protocolo HTTP
- 1.10. Metodologías en desarrollo Full Stack
  - 1.10.1. Scrum
  - 1.10.2. XP
  - 1.10.3. *Design sprint*

## Módulo 2. *Front-end* en Programación

- 2.1. Lenguaje HTML
  - 2.1.1. Documento HTML
  - 2.1.2. Elemento *Head*
  - 2.1.3. Elemento *Body*
  - 2.1.4. Texto
  - 2.1.5. Hipervínculos
  - 2.1.6. Imágenes
  - 2.1.7. Primer *Site*
- 2.2. Lenguaje HTML. *Layouts*
  - 2.2.1. Lenguaje HTML. Elementos
  - 2.2.2. *Layout* tradicional
  - 2.2.3. *Layout* semántico



## 2.3. Hojas de Estilo en Cascada CSS (*Cascading Style Sheets*)

### 2.3.1. Inclusión de CSS en un documento HTML

### 2.3.2. Comentarios

### 2.3.3. Selectores

### 2.3.4. Selectores avanzados

## 2.4. Propiedades CSS (*Cascading Style Sheets*)

### 2.4.1. Color

### 2.4.2. Texto

### 2.4.3. Pseudo clases

### 2.4.4. Transiciones

### 2.4.5. Animaciones

### 2.4.6. Animación de elementos

### 2.4.7. Animación avanzada

## 2.5. Modelo de caja

### 2.5.1. Alto y ancho

### 2.5.2. Margen

### 2.5.3. Relleno

## 2.6. Posicionamiento

### 2.6.1. Posicionamiento estático

### 2.6.2. Posicionamiento relativo

### 2.6.3. Posicionamiento absoluto

### 2.6.4. Posicionamiento fijo

### 2.6.5. *Floats*

## 2.7. Diseño adaptable

### 2.7.1. *Viewport*

### 2.7.2. *Media queries*

### 2.7.3. Unidades CSS

### 2.7.4. *Images*

### 2.7.5. *Frameworks*

## 2.8. *Layout* moderno

### 2.8.1. *Flex*

### 2.8.2. *Grid*

### 2.8.3. *Flex* vs. *Grid*

## 2.9. Preprocesador

### 2.9.1. Sass

### 2.9.2. Variables

### 2.9.3. Mixins

### 2.9.4. Bucles

### 2.9.5. Funciones

## 2.10. Sistema de diseño

### 2.10.1. Bootstrap

### 2.10.2. Grilla de Bootstrap

### 2.10.3. *Header* y *footer* de nuestro sitio

### 2.10.4. Formularios

### 2.10.5. *Cards*

### 2.10.6. Modales

## Módulo 3. Lenguaje JavaScript aplicado a *Full Stack Developer*

## 3.1. Tipos primitivos y operadores

### 3.1.1. Lenguaje JavaScript

### 3.1.2. Números y sus operadores

### 3.1.3. Cadenas de texto y sus operadores

### 3.1.4. Valores booleanos

### 3.1.5. Conversión entre tipos

## 3.2. Controladores de flujo y estructura

### 3.2.1. Expresiones y sentencias

### 3.2.2. Variables y constantes

### 3.2.3. Sentencia *if*

### 3.2.4. Sentencias *for*, *while*

## 3.3. Funciones

### 3.3.1. Funciones

### 3.3.2. Parámetros

### 3.3.3. Funciones como parámetros

### 3.3.4. Ámbito de las variables

### 3.3.5. *Scopes* anidados

### 3.3.6. *Hoisting*

### 3.3.7. *Closures*

### 3.3.8. Recursividad

- 3.4. Estructuras de datos: Objetos
  - 3.4.1. Tipo *Object*
  - 3.4.2. Creación de objetos
  - 3.4.3. Acceso a los valores de un objeto
  - 3.4.4. Añadido o eliminación de propiedades
  - 3.4.5. Objetos anidados
  - 3.4.6. *Destructuring* de objetos
  - 3.4.7. Métodos del tipo *Object*
  - 3.4.8. *Spread operator*
  - 3.4.9. Inmutabilidad
- 3.5. Estructuras de datos: *Array*
  - 3.5.1. Estructura de Datos. *Array*
  - 3.5.2. *Array*. Tipología
  - 3.5.3. *Arrays* anidados
  - 3.5.4. Métodos de un *Array*
- 3.6. POO: *Prototype* y Clases
  - 3.6.1. POO. Programación Orientada a Objetos
  - 3.6.2. Prototipos
  - 3.6.3. Clases
  - 3.6.4. Datos privados
  - 3.6.5. Subclases
  - 3.6.6. *Call* y *apply*
- 3.7. Tipos de JavaScript
  - 3.7.1. *Set*
  - 3.7.2. *WeakSet*
  - 3.7.3. *Map*
  - 3.7.4. *WeakMap*
  - 3.7.5. Expresiones regulares
- 3.8. Utilidades de JavaScript
  - 3.8.1. *Date*
  - 3.8.2. *Math*
  - 3.8.3. *Symbol*
  - 3.8.4. JSON





- 3.9. JavaScript en el Browser
  - 3.9.1. Inclusión de JavaScript en una web
  - 3.9.2. DOM
  - 3.9.3. Eventos
  - 3.9.4. *Storage* en el navegador
- 3.10. Programación Asíncrona
  - 3.10.1. La Programación Asíncrona
  - 3.10.2. *Event loop*
  - 3.10.3. *Callbacks*
  - 3.10.4. *Promises*
  - 3.10.5. *Async / Await*

#### Módulo 4. Maquetación Web aplicado a *Full Stack Developer*

- 4.1. CSS y maquetación
  - 4.1.1. Maquetación con tablas
  - 4.1.2. Diseño fluido
  - 4.1.3. La era del *Responsive*
  - 4.1.4. *Mobile First* vs. *Desktop First*
- 4.2. CSS y las reglas del diseño web
  - 4.2.1. Selectores
  - 4.2.2. Pseudo clases
  - 4.2.3. Pseudo elementos
- 4.3. Maquetación con CSS
  - 4.3.1. Reglas del *Box Model*
  - 4.3.2. Tipografías
  - 4.3.3. Colores
  - 4.3.4. Imágenes
  - 4.3.5. Fondos
  - 4.3.6. Tablas
  - 4.3.7. Formularios
  - 4.3.8. Mostrando y ocultando elementos
  - 4.3.9. Variables CSS

- 4.4. Diseño *Responsive* y diseño fluido
  - 4.4.1. Elementos flotantes
  - 4.4.2. *Grid CSS*
  - 4.4.3. *Media Queries*
  - 4.4.4. *Flex Box*
- 4.5. La Cascada de CSS
  - 4.5.1. Prioridad de reglas CSS
  - 4.5.2. Sobrescribiendo reglas
  - 4.5.3. Clases vs identificadores
- 4.6. SaSS
  - 4.6.1. Software como Servicio (SaSS)
  - 4.6.2. Instalación de SaSS
  - 4.6.3. Ejecutando y compilando SaSS
  - 4.6.4. Estructura de un directorio SaSS
- 4.7. Uso de SaSS
  - 4.7.1. Variables en Sass
  - 4.7.2. Modularizado de nuestro proyecto
  - 4.7.3. Sintaxis de SaSS
- 4.8. Lógica de SaSS
  - 4.8.1. *Mixins*
  - 4.8.2. *Maps*
  - 4.8.3. Funciones y estructuras de control
- 4.9. Maquetación con Bootstrap
  - 4.9.1. Bootstrap
  - 4.9.2. *Layout* de Bootstrap
  - 4.9.3. Formularios
  - 4.9.4. *Box Model* con Bootstrap
  - 4.9.5. Colores y tipografías
  - 4.9.6. Enlaces y botones
  - 4.9.7. Mostrar y ocultar elementos con Bootstrap
  - 4.9.8. *Flex Box* con Bootstrap
  - 4.9.9. Componentes

- 4.10. *Theming* Bootstrap
  - 4.10.1. Reescribiendo Bootstrap con SaSS (*Software as a Service*)
  - 4.10.2. Estructura de archivos
  - 4.10.3. Creación de nuestro propio *framework* CSS (*Cascading Style Sheets*)

## Módulo 5. Herramientas de Javascript. Librería Reactjs

- 5.1. Herramienta de Javascript ReactJS
  - 5.1.1. La herramienta ReactJS
  - 5.1.2. Create React App
  - 5.1.3. *JavaScript Syntax Extension*
- 5.2. Componentes de ReactJS
  - 5.2.1. Componentes
  - 5.2.2. *Props*
  - 5.2.3. Renderizado
- 5.3. Eventos en la librería ReactJS
  - 5.3.1. Manejo de eventos
  - 5.3.2. Manejo de eventos en línea
  - 5.3.3. Eventos en la librería ReactJS
- 5.4. Configurando *Hooks* de ReactJS
  - 5.4.1. Estado de un componente
  - 5.4.2. *Hook* de estado
  - 5.4.3. *Hook* de efecto
  - 5.4.4. *Custom hooks*
  - 5.4.5. Otros *hooks*
- 5.5. Componente *Context* en ReactJS
  - 5.5.1. Componente *Context* en ReactJS
  - 5.5.2. Uso de *Context*
  - 5.5.3. Estructura del *Context*
  - 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. *Aplicación Práctica de Uso de Context*
- 5.6. Enrutado en ReactJs
  - 5.6.1. *Router*
  - 5.6.2. *React router*
  - 5.6.3. *Instalación*
  - 5.6.4. *Enrutado básico*
  - 5.6.5. *Enrutado dinámico*
  - 5.6.6. *Componentes primarios*
  - 5.6.7. *React router hooks*
- 5.7. *Uso de listas y formularios con ReactJS*
  - 5.7.1. *Listas y bucles*
  - 5.7.2. *Formularios y validaciones*
  - 5.7.3. *React hook forms*
- 5.8. *Uso de estilos en ReactJS*
  - 5.8.1. *Estilizado tradicional*
  - 5.8.2. *Estilos en línea*
  - 5.8.3. *Agregado de librería de sistema de diseño*
- 5.9. *Realización de pruebas en Javascript. Herramientas*
  - 5.9.1. *Testing*
  - 5.9.2. *Jest JavaScript Testing Framework*
  - 5.9.3. *Visual testing and Documentation*
- 5.10. *Despliegue de código con ReactJS*
  - 5.10.1. *Hosting*
  - 5.10.2. *Proveedores*
  - 5.10.3. *Preparación del proyecto*
  - 5.10.4. *Despliegue en Heroku*

## Módulo 6. Framework de JavaScript. Angular

- 6.1. *El Framework Angular y su Arquitectura*
  - 6.1.1. *Angular CLI*
  - 6.1.2. *Arquitectura*
  - 6.1.3. *Workspace y estructura*
  - 6.1.4. *Entorno*
- 6.2. *Componentes del Framework Angular*
  - 6.2.1. *Ciclo de vida*
  - 6.2.2. *Encapsulación de la vista*
  - 6.2.3. *Interacción entre componentes*
  - 6.2.4. *Proyección de contenido*
- 6.3. *Plantillas del Framework Angular*
  - 6.3.1. *Interpolación de texto*
  - 6.3.2. *Declaraciones*
  - 6.3.3. *Property binding*
  - 6.3.4. *Class, style y attribute binding*
  - 6.3.5. *Event binding y two-way binding*
  - 6.3.6. *Pipes*
- 6.4. *Directivas del Framework Angular*
  - 6.4.1. *Directivas de Angular*
  - 6.4.2. *Directivas de atributo*
  - 6.4.3. *Directivas de estructura*
- 6.5. *Servicios e inyección de dependencias*
  - 6.5.1. *Servicios*
  - 6.5.2. *Inyección de dependencias*
  - 6.5.3. *Service providers*
- 6.6. *Routing y navegación*
  - 6.6.1. *Aplicación con routing*
  - 6.6.2. *Enrutamiento básico*
  - 6.6.3. *Rutas anidadas*
  - 6.6.4. *Parámetros*
  - 6.6.5. *Acceso y autorización*
  - 6.6.6. *Lazy loading de módulos*

- 6.7. RxJS
  - 6.7.1. Observables
  - 6.7.2. Observers
  - 6.7.3. Subscripciones
  - 6.7.4. Operadores
- 6.8. Formularios y HTTP
  - 6.8.1. Formularios reactivos
  - 6.8.2. Validación de campos
  - 6.8.3. Formularios dinámicos
  - 6.8.4. Peticiones
  - 6.8.5. *Interceptors*
  - 6.8.6. Seguridad
- 6.9. Animaciones
  - 6.9.1. Transiciones y *Triggers*
  - 6.9.2. Transiciones de rutas
  - 6.9.3. Diferencias entre Transiciones
- 6.10. Testing en el *Framework* Angular
  - 6.10.1. Testeo de servicios
  - 6.10.2. Testeo de componentes
  - 6.10.3. Testeo de directivas y *pipes*

## Módulo 7. Programación en Lenguaje NodeJS

- 7.1. NodeJS y su Arquitectura
  - 7.1.1. NPM y manejo de paquetes
  - 7.1.2. Ejecución de un programa
  - 7.1.3. Módulos
  - 7.1.4. Creación de un módulo
  - 7.1.5. *Loop* de eventos
- 7.2. Servidor Backend, HTTP, *Express* y *Sockets*
  - 7.2.1. Módulo HTTP
  - 7.2.2. *Express*
  - 7.2.3. *Socket.io*
- 7.3. Bases de datos y caché
  - 7.3.1. MongoDB
  - 7.3.2. Mongoose
  - 7.3.3. SQL
  - 7.3.4. *Sequelize*
  - 7.3.5. Redis
- 7.4. Sistema de ficheros y *Os*
  - 7.4.1. *File System Module*
  - 7.4.2. *Os Module*
  - 7.4.3. *Cluster module*
- 7.5. Eventos, *Buffers* y *Streams*
  - 7.5.1. Eventos
  - 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. Arquitectura y Buenas Prácticas
  - 7.7.1. DRY
  - 7.7.2. SOLID
  - 7.7.3. CRUD
  - 7.7.4. MVC
  - 7.7.5. Monolitos
  - 7.7.6. Microservicios
  - 7.7.7. Arquitecturas hexagonales
- 7.8. *Typescript*
  - 7.8.1. Tipos, interfaces y clases
  - 7.8.2. Funciones y módulos
  - 7.8.3. Genéricos
  - 7.8.4. *Namespaces*
  - 7.8.5. Decoradores

- 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. Construcción de una API REST con Express
- 7.10. Creación y Contenerización de una Aplicación con NestJS
  - 7.10.1. Nest CLI
  - 7.10.2. *Docker*
  - 7.10.3. Construcción de una aplicación

## Módulo 8. Base de Datos para Desarrolladores *Full Stack*

- 8.1. Base de datos para desarrolladores *Full Stack*
  - 8.1.1. Base de datos dentro del desarrollo de aplicaciones
  - 8.1.2. Capacidades de bases de datos
  - 8.1.3. SQL (*Structured Query Language*)
- 8.2. Elección de base de datos
  - 8.2.1. Aplicación o servicio a considerar
  - 8.2.2. Categorías de bases de datos
  - 8.2.3. Panorama de base de datos
- 8.3. Desarrollo con MySQL
  - 8.3.1. Desarrollo con MySQL
  - 8.3.2. Despliegue de modelo relacional con MySQL
  - 8.3.3. Conexión a MySQL
- 8.4. Desarrollo con Oracle Database
  - 8.4.1. Desarrollo con Oracle DB
  - 8.4.2. Despliegue del modelo
  - 8.4.3. Conexión a Oracle Database
- 8.5. Desarrollo con Oracle SQL Server
  - 8.5.1. Oracle SQL Server
  - 8.5.2. Despliegue del modelo
  - 8.5.3. Conexión a SQL Server

- 8.6. Desarrollo con NoSQL
  - 8.6.1. Comparación con bases de datos SQL
  - 8.6.2. Creación de base de datos en MongoDB
  - 8.6.3. Conexión a MongoDB
- 8.7. Desarrollo con Grafos
  - 8.7.1. Desarrollo con Grafos
  - 8.7.2. Creación de base de datos con Neo4j
  - 8.7.3. Conexión con Neo4j
- 8.8. Desarrollo con base de datos Clave-Valor
  - 8.8.1. Desarrollo con base de datos k-v
  - 8.8.2. Creación de base de datos con Redis
  - 8.8.3. Conexión con Redis
- 8.9. Bases de datos con otros tipos de datos
  - 8.9.1. *Elastic Search*
  - 8.9.2. *Inmemory database*
  - 8.9.3. Desarrollo con datos espaciales
- 8.10. Base de datos. Aspectos avanzados
  - 8.10.1. Bases de datos en desarrollos *cloud native*
  - 8.10.2. Bases de datos en arquitectura microservicios
  - 8.10.3. CI/CD y las bases de datos

## Módulo 9. UX CX. Experiencia de cliente

- 9.1. *Customer Experience*
  - 9.1.1. *Customer Experience (CX)*
  - 9.1.2. Nuevas necesidades en el consumo
  - 9.1.3. *Feedback* en *Customer Experience*
- 9.2. Tecnologías innovadoras
  - 9.2.1. Máquinas que piensan
  - 9.2.2. Nuevas formas de compartir información
  - 9.2.3. Medición de lo que no se puede medir
- 9.3. Canales de interacción con el usuario
  - 9.3.1. Análisis del cliente
  - 9.3.2. Personalización
  - 9.3.3. Múltiples canales de interacción con el usuario

- 9.4. Analítica de usuario
  - 9.4.1. Estructura de la web
  - 9.4.2. Analítica de usuario
  - 9.4.3. Analítica de usuario avanzada
- 9.5. Nielsen y su impacto en la CX
  - 9.5.1. Nielsen y su Impacto en la CX
  - 9.5.2. Técnicas de *User Testing*
- 9.6. Herramientas de *Customer Experience*
  - 9.6.1. Herramientas avanzadas
  - 9.6.2. Movilidad
  - 9.6.3. Accesibilidad
- 9.7. Nuevas metodologías
  - 9.7.1. El reto del usuario
  - 9.7.2. Proceso de UX
  - 9.7.3. Investigación de usuario
- 9.8. Comunicación de un diseño
  - 9.8.1. *Wireframing*
  - 9.8.2. Herramientas de comunicación de un diseño
  - 9.8.3. Herramientas avanzadas de comunicación de un diseño
- 9.9. Diseño de UI
  - 9.9.1. Diseño de UI
  - 9.9.2. Interfaces web y móvil
  - 9.9.3. Componentes web y móvil
- 9.10. Elaboración de una CX
  - 9.10.1. Elaboración de una CX
  - 9.10.2. Diseño de nuevas experiencias
  - 9.10.3. Interfaces

## Módulo 10. Integración Continua y Despliegue de Aplicaciones

- 10.1. Integración Continua y Despliegue Continuo: CI/CD
  - 10.1.1. Uso de Integración Continua y Despliegue Continuo (CI/CD)
  - 10.1.2. Diferencias entre Integración Continua y Despliegue Continuo (CI/CD)
  - 10.1.3. Integración Continua y Despliegue Continuo. Beneficios de CI/CD
- 10.2. Los nuevos paradigmas de desarrollo
  - 10.2.1. De Waterfall a DevOps
  - 10.2.2. Guía de estilo: los 12 factores
  - 10.2.3. Cloud Native, microservicios y serverless
- 10.3. DevOps, más allá de CI/CD
  - 10.3.1. DevOps
  - 10.3.2. DevOps. *Continuous Everything*
  - 10.3.3. DevOps vs SRE
- 10.4. Tecnología de contenedores I - Docker
  - 10.4.1. Los contenedores. Aportación
  - 10.4.2. Docker. Arquitectura
  - 10.4.3. Proceso de despliegue con Docker
- 10.5. Tecnología de contenedores II - Kubernetes
  - 10.5.1. Orquestación
  - 10.5.2. Kubernetes
  - 10.5.3. El ecosistema de Kubernetes
- 10.6. Configuración de Infraestructura con GitOps
  - 10.6.1. Infraestructura inmutable
  - 10.6.2. GitOps
  - 10.6.3. Herramientas GitOps
- 10.7. Pipelines y automatización. Casos de Uso de CI/CD
  - 10.7.1. Integración continua
  - 10.7.2. Despliegue y entrega continua
  - 10.7.3. Validaciones automáticas
  - 10.7.4. Buenas prácticas en CI/CD



- 10.8. CI/CD con Jenkins. Referencia
  - 10.8.1. CI/CD con Jenkins
  - 10.8.2. Pipelines Jenkins
  - 10.8.3. Buenas prácticas con Jenkins
- 10.9. Ecosistema CI/CD
  - 10.9.1. Organización del ecosistema
  - 10.9.2. Herramientas avanzadas
  - 10.9.3. *Dagger*. El futuro
- 10.10. Fases finales del ciclo de software orientado a CI/CD
  - 10.10.1. Aplicación de IA al proceso de CI/CD
  - 10.10.2. DevSecOps
  - 10.10.3. *Chaos Engineering*

“

*Este enfoque integral y práctico te preparará para enfrentar con éxito los desafíos del mercado laboral actual, dotándote de las habilidades para desarrollar y mantener aplicaciones web sofisticadas y funcionales”*

# 07 Prácticas

Tras superar el periodo teórico online, el programa contemplará un periodo de 3 semanas de capacitación práctica en una empresa de referencia, especializada en programación y desarrollo web. De esta forma, el egresado tendrá a su disposición el apoyo de un tutor personal, que le acompañará durante todo el proceso y le dedicará el 100% de su tiempo, tanto en la preparación como en el desarrollo de las prácticas.



“

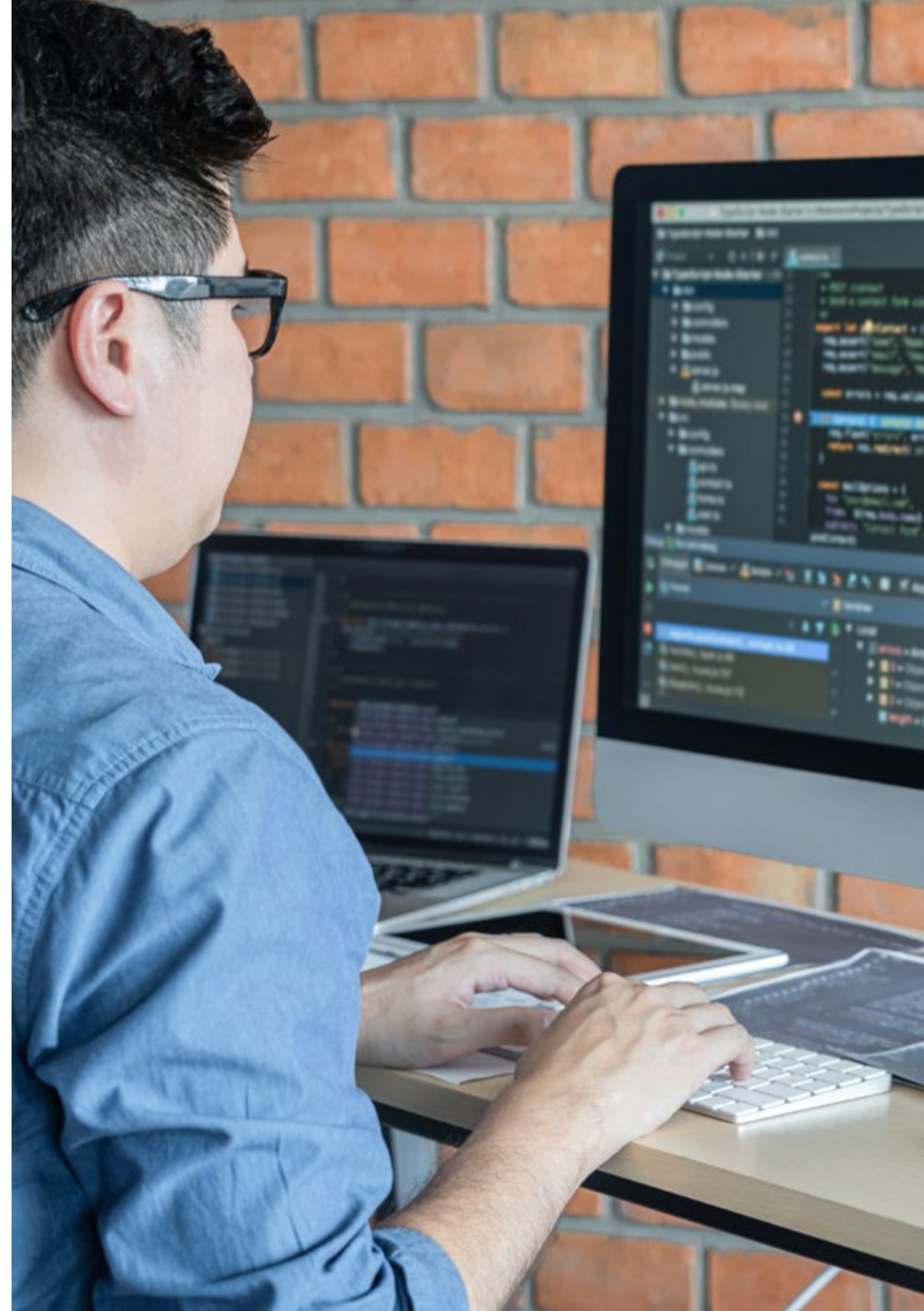
*Las prácticas te ofrecerán la oportunidad de colaborar en proyectos reales, tanto individuales como en equipo, bajo la supervisión y orientación de profesionales experimentados del sector”*



El periodo de Capacitación Práctica de este programa de Programación Front End Full Stack Developer está conformado por una estancia práctica en una empresa informática de gran prestigio, de 3 semanas de duración, de lunes a viernes y con jornadas de 8 horas consecutivas de capacitación práctica, al lado de un especialista adjunto. Esta estancia permitirá trabajar en proyectos de programación reales, junto a un equipo de profesionales de referencia en el área de la enfermería de los cuidados intensivos, aplicando los procedimientos diagnósticos más innovadores y planificando la terapéutica de última generación en cada patología.

En esta propuesta de capacitación, cada actividad está diseñada para fortalecer y perfeccionar las competencias clave que exige el ejercicio especializado de esta área. De este modo, se potenciará al máximo el perfil profesional, impulsando un desempeño sólido, eficiente y altamente competitivo.

La parte práctica se realizará con la participación activa del estudiante desempeñando las actividades y procedimientos de cada área de competencia (aprender a aprender y aprender a hacer), con el acompañamiento y guía de los profesores y demás compañeros de entrenamiento que faciliten el trabajo en equipo y la integración multidisciplinar como competencias transversales para la praxis de la informática (aprender a ser y aprender a relacionarse).





Los procedimientos descritos a continuación serán la base de la parte práctica de la capacitación, y su realización estará sujeta a la disponibilidad propia del centro y su volumen de trabajo, siendo las actividades propuestas las siguientes:

Módulo	Actividad Práctica
<b>Diseño y desarrollo de interfaces</b>	Crear interfaces de usuario intuitivas y atractivas
	Implementar diseños responsivos que se adapten a distintos dispositivos y tamaños de pantalla
	Utilizar herramientas de diseño y prototipado como Adobe XD, Figma o Sketch
	Realizar pruebas de usabilidad para asegurar una experiencia de usuario óptima
<b>Desarrollo de aplicaciones web</b>	Desarrollar aplicaciones web utilizando tecnologías <i>front-end</i> como HTML, CSS y JavaScript
	Utilizar <i>frameworks</i> y librerías como React, Angular o Vue.js
	Integrar APIs y servicios externos para enriquecer las funcionalidades de las aplicaciones
	Optimizar el rendimiento de las aplicaciones web para una mejor experiencia del usuario
<b>Gestión de proyectos y trabajo en equipo</b>	Utilizar metodologías ágiles como Scrum o Kanban para gestionar proyectos
	Colaborar con diseñadores, desarrolladores y otros miembros del equipo
	Participar en revisiones de código y aportar <i>feedback</i> constructivo
	Gestionar el control de versiones utilizando herramientas como Git y GitHub
<b>Desarrollo <i>back-end</i> y bases de datos</b>	Desarrollar y mantener servidores y bases de datos utilizando Node.js y Express
	Diseñar y gestionar bases de datos relacionales y no relacionales como MySQL, PostgreSQL, y MongoDB
	Implementar y asegurar API RESTful para la comunicación entre el <i>front-end</i> y el <i>back-end</i>
	Gestionar la autenticación y autorización de usuarios en aplicaciones web

Módulo	Actividad Práctica
<b>Testing y depuración</b>	Escribir y ejecutar pruebas unitarias y de integración utilizando herramientas como Jest, Mocha o Jasmine
	Identificar y solucionar errores en el código de manera eficiente
	Automatizar pruebas y despliegues utilizando CI/CD pipelines
	Realizar revisiones de código para mantener estándares de calidad y buenas prácticas
<b>Seguridad y optimización</b>	Implementar medidas de seguridad para proteger las aplicaciones web de amenazas y vulnerabilidades
	Optimizar el rendimiento de las aplicaciones web, reduciendo tiempos de carga y mejorando la eficiencia del código
	Implementar técnicas de optimización de SEO para mejorar la visibilidad en motores de búsqueda
	Realizar auditorías de seguridad y rendimiento de las aplicaciones web



*Te enfrentarás a desafíos específicos del desarrollo web, desde la implementación de diseños y funcionalidades interactivas hasta la integración de bases de datos y la optimización de rendimiento”*

## Seguro de responsabilidad civil

La máxima preocupación de esta institución es garantizar la seguridad tanto de los profesionales en prácticas como de los demás agentes colaboradores necesarios en los procesos de capacitación práctica en la empresa. Dentro de las medidas dedicadas a lograrlo, se encuentra la respuesta ante cualquier incidente que pudiera ocurrir durante todo el proceso de enseñanza-aprendizaje.

Para ello, esta entidad educativa se compromete a contratar un seguro de responsabilidad civil que cubra cualquier eventualidad que pudiera surgir durante el desarrollo de la estancia en el centro de prácticas.

Esta póliza de responsabilidad civil de los profesionales en prácticas tendrá coberturas amplias y quedará suscrita de forma previa al inicio del periodo de la capacitación práctica. De esta forma el profesional no tendrá que preocuparse en caso de tener que afrontar una situación inesperada y estará cubierto hasta que termine el programa práctico en el centro.



## Condiciones generales de la capacitación práctica

Las condiciones generales del acuerdo de prácticas para el programa serán las siguientes:

**1. TUTORÍA:** durante el Máster Semipresencial el alumno tendrá asignados dos tutores que le acompañarán durante todo el proceso, resolviendo las dudas y cuestiones que pudieran surgir. Por un lado, habrá un tutor profesional perteneciente al centro de prácticas que tendrá como fin orientar y apoyar al alumno en todo momento. Por otro lado, también tendrá asignado un tutor académico cuya misión será la de coordinar y ayudar al alumno durante todo el proceso resolviendo dudas y facilitando todo aquello que pudiera necesitar. De este modo, el profesional estará acompañado en todo momento y podrá consultar las dudas que le surjan, tanto de índole práctica como académica.

**2. DURACIÓN:** el programa de prácticas tendrá una duración de tres semanas continuadas de formación práctica, distribuidas en jornadas de 8 horas y cinco días a la semana. Los días de asistencia y el horario serán responsabilidad del centro, informando al profesional debidamente y de forma previa, con suficiente tiempo de antelación para favorecer su organización.

**3. INASISTENCIA:** en caso de no presentarse el día del inicio del Máster Semipresencial, el alumno perderá el derecho a la misma sin posibilidad de reembolso o cambio de fechas. La ausencia durante más de dos días a las prácticas sin causa justificada/ médica, supondrá la renuncia las prácticas y, por tanto, su finalización automática. Cualquier problema que aparezca durante el transcurso de la estancia se tendrá que informar debidamente y de forma urgente al tutor académico.

**4. CERTIFICACIÓN:** el alumno que supere el Máster Semipresencial recibirá un certificado que le acreditará la estancia en el centro en cuestión.

**5. RELACIÓN LABORAL:** el Máster Semipresencial no constituirá una relación laboral de ningún tipo.

**6. ESTUDIOS PREVIOS:** algunos centros podrán requerir certificado de estudios previos para la realización del Máster Semipresencial. En estos casos, será necesario presentarlo al departamento de prácticas de TECH para que se pueda confirmar la asignación del centro elegido.

**7. NO INCLUYE:** el Máster Semipresencial no incluirá ningún elemento no descrito en las presentes condiciones. Por tanto, no incluye alojamiento, transporte hasta la ciudad donde se realicen las prácticas, visados o cualquier otra prestación no descrita.

No obstante, el alumno podrá consultar con su tutor académico cualquier duda o recomendación al respecto. Este le brindará toda la información que fuera necesaria para facilitarle los trámites.



# 08

## ¿Dónde puedo hacer las Prácticas?

Este programa de Máster Semipresencial contempla en su itinerario una estancia práctica en una empresa informática de prestigio donde el alumnado pondrá en práctica todo lo aprendido en materia de Programación Front End Full Stack Developer. En este sentido, y para acercar este título a más profesionales, TECH ofrece al egresado la oportunidad de cursarlo en diferentes instituciones alrededor de la geografía nacional. De esta manera, afianza su compromiso con la calidad y la educación asequible para todos.



“

*Además de fortalecer tus habilidades técnicas, las prácticas fomentarán el desarrollo de tus competencias en comunicación, gestión de proyectos y resolución de problemas”*



## tech 46 | ¿Dónde puedo hacer las Prácticas?



El alumno podrá cursar la parte práctica de este Máster Semipresencial en los siguientes centros:



Informática

### Captia Ingeniería

País	Ciudad
España	Madrid

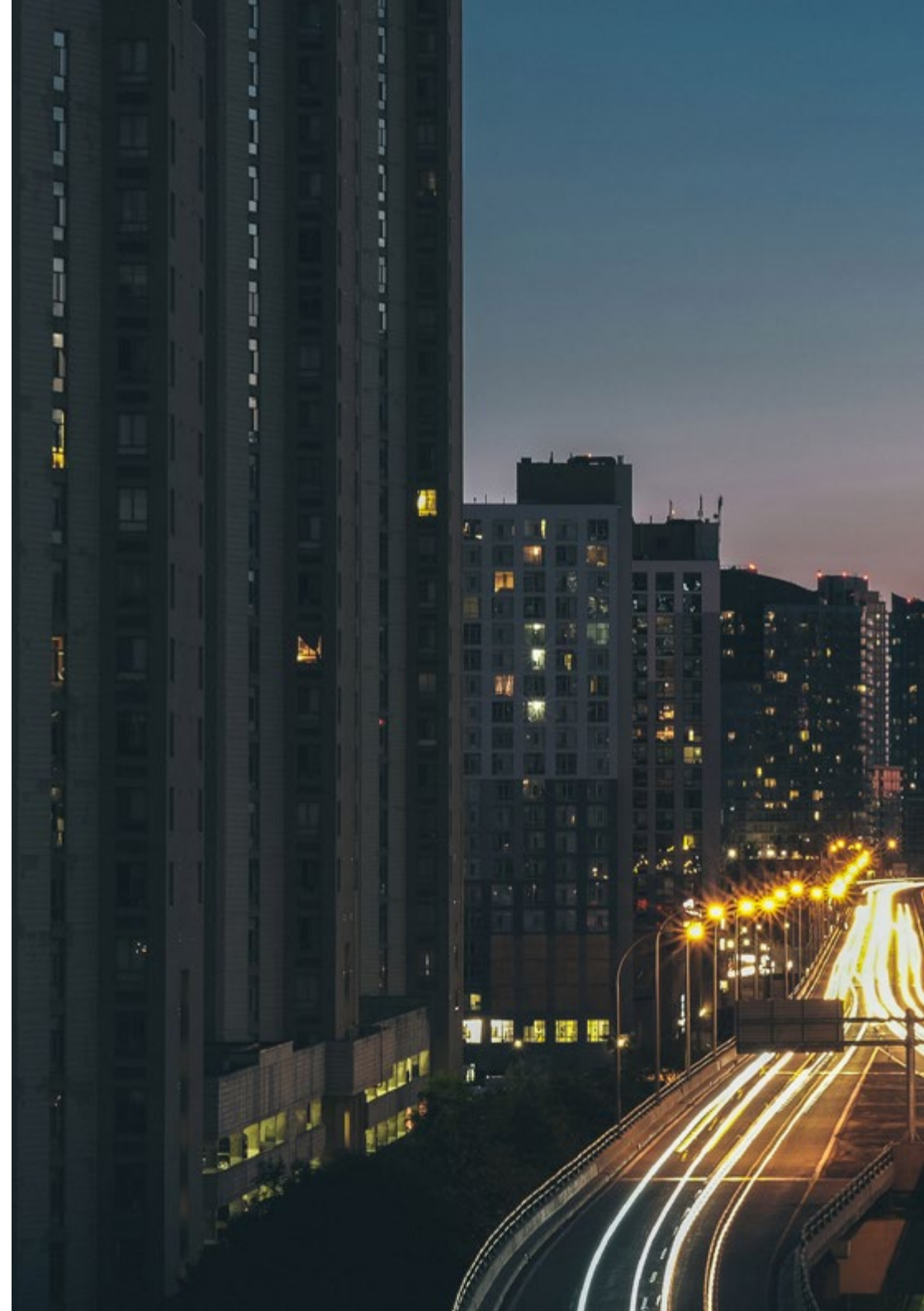
Dirección: Av. de las Nieves, 37, Bloque A Planta 1  
Oficina E, 28935, Móstoles, Madrid

Empresa informática dedicada a proporcionar soluciones tecnológicas avanzadas a las industrias

---

**Capacitaciones prácticas relacionadas:**

- Visual Analytics y Big Data
- Desarrollo de Software





“

*Impulsa tu trayectoria profesional con una enseñanza holística, que te permite avanzar tanto a nivel teórico como práctico”*



09

# Metodología de estudio

TECH es la primera universidad en el mundo que combina la metodología de los **case studies** con el **Relearning**, un sistema de aprendizaje 100% online basado en la reiteración dirigida.

Esta disruptiva estrategia pedagógica ha sido concebida para ofrecer a los profesionales la oportunidad de actualizar conocimientos y desarrollar competencias de un modo intensivo y riguroso. Un modelo de aprendizaje que coloca al estudiante en el centro del proceso académico y le otorga todo el protagonismo, adaptándose a sus necesidades y dejando de lado las metodologías más convencionales.



“

*TECH te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera”*

## El alumno: la prioridad de todos los programas de TECH

En la metodología de estudios de TECH el alumno es el protagonista absoluto. Las herramientas pedagógicas de cada programa han sido seleccionadas teniendo en cuenta las demandas de tiempo, disponibilidad y rigor académico que, a día de hoy, no solo exigen los estudiantes sino los puestos más competitivos del mercado.

Con el modelo educativo asincrónico de TECH, es el alumno quien elige el tiempo que destina al estudio, cómo decide establecer sus rutinas y todo ello desde la comodidad del dispositivo electrónico de su preferencia. El alumno no tendrá que asistir a clases en vivo, a las que muchas veces no podrá acudir. Las actividades de aprendizaje las realizará cuando le venga bien. Siempre podrá decidir cuándo y desde dónde estudiar.

“

*En TECH NO tendrás clases en directo  
(a las que luego nunca puedes asistir)”*





### Los planes de estudios más exhaustivos a nivel internacional

TECH se caracteriza por ofrecer los itinerarios académicos más completos del entorno universitario. Esta exhaustividad se logra a través de la creación de temarios que no solo abarcan los conocimientos esenciales, sino también las innovaciones más recientes en cada área.

Al estar en constante actualización, estos programas permiten que los estudiantes se mantengan al día con los cambios del mercado y adquieran las habilidades más valoradas por los empleadores. De esta manera, quienes finalizan sus estudios en TECH reciben una preparación integral que les proporciona una ventaja competitiva notable para avanzar en sus carreras.

Y además, podrán hacerlo desde cualquier dispositivo, pc, tableta o smartphone.

“

*El modelo de TECH es asincrónico, de modo que te permite estudiar con tu pc, tableta o tu smartphone donde quieras, cuando quieras y durante el tiempo que quieras”*

## Case studies o Método del caso

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de negocios del mundo. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, su función era también presentarles situaciones complejas reales. Así, podían tomar decisiones y emitir juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Con este modelo de enseñanza es el propio alumno quien va construyendo su competencia profesional a través de estrategias como el *Learning by doing* o el *Design Thinking*, utilizadas por otras instituciones de renombre como Yale o Stanford.

Este método, orientado a la acción, será aplicado a lo largo de todo el itinerario académico que el alumno emprenda junto a TECH. De ese modo se enfrentará a múltiples situaciones reales y deberá integrar conocimientos, investigar, argumentar y defender sus ideas y decisiones. Todo ello con la premisa de responder al cuestionamiento de cómo actuaría al posicionarse frente a eventos específicos de complejidad en su labor cotidiana.



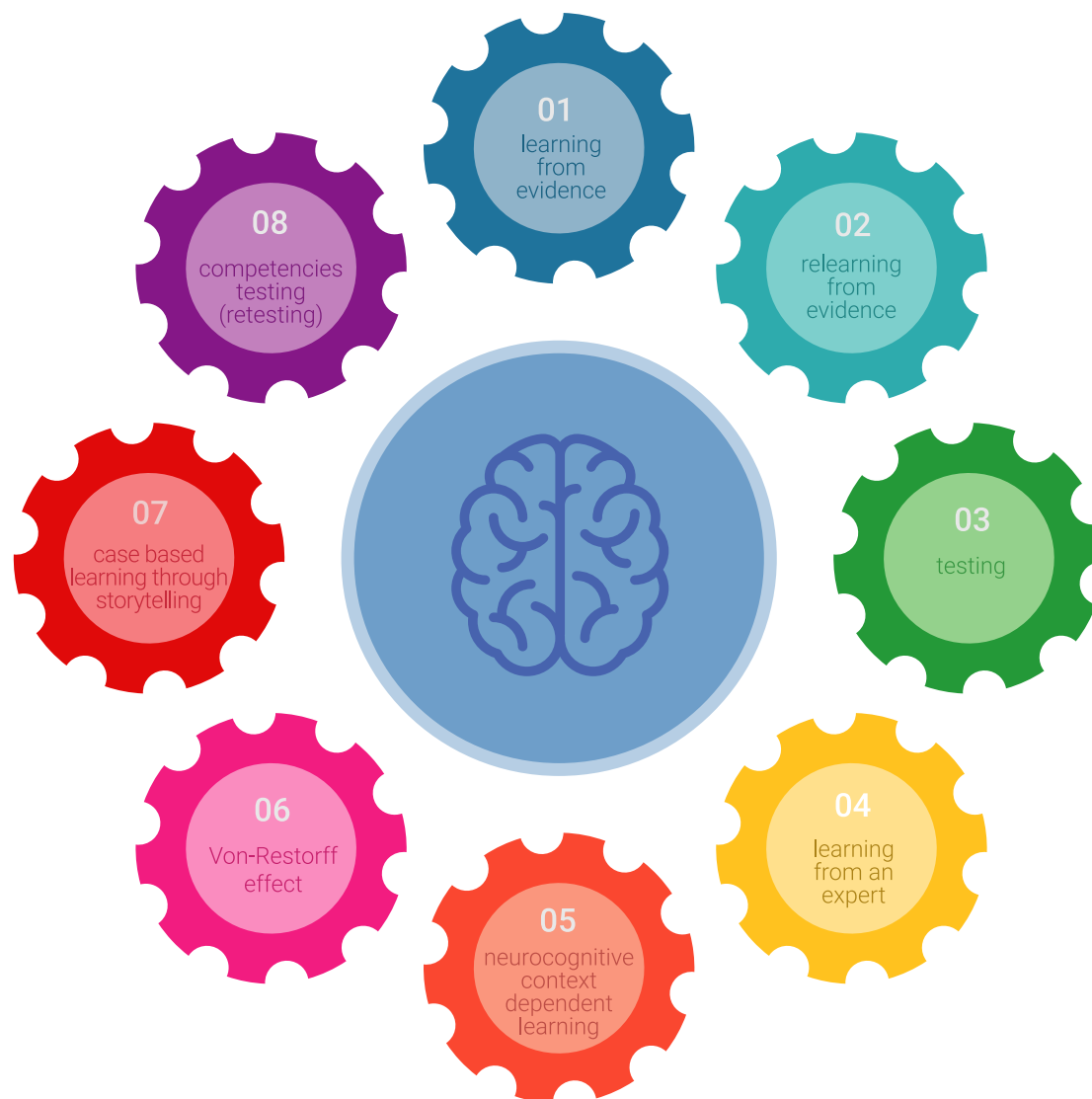
## Método Relearning

En TECH los *case studies* son potenciados con el mejor método de enseñanza 100% online: el *Relearning*.

Este método rompe con las técnicas tradicionales de enseñanza para poner al alumno en el centro de la ecuación, proveyéndole del mejor contenido en diferentes formatos. De esta forma, consigue repasar y reiterar los conceptos clave de cada materia y aprender a aplicarlos en un entorno real.

En esta misma línea, y de acuerdo a múltiples investigaciones científicas, la reiteración es la mejor manera de aprender. Por eso, TECH ofrece entre 8 y 16 repeticiones de cada concepto clave dentro de una misma lección, presentada de una manera diferente, con el objetivo de asegurar que el conocimiento sea completamente afianzado durante el proceso de estudio.

*El Relearning te permitirá aprender con menos esfuerzo y más rendimiento, implicándote más en tu especialización, desarrollando el espíritu crítico, la defensa de argumentos y el contraste de opiniones: una ecuación directa al éxito.*



## Un Campus Virtual 100% online con los mejores recursos didácticos

Para aplicar su metodología de forma eficaz, TECH se centra en proveer a los egresados de materiales didácticos en diferentes formatos: textos, vídeos interactivos, ilustraciones y mapas de conocimiento, entre otros. Todos ellos, diseñados por profesores cualificados que centran el trabajo en combinar casos reales con la resolución de situaciones complejas mediante simulación, el estudio de contextos aplicados a cada carrera profesional y el aprendizaje basado en la reiteración, a través de audios, presentaciones, animaciones, imágenes, etc.

Y es que las últimas evidencias científicas en el ámbito de las Neurociencias apuntan a la importancia de tener en cuenta el lugar y el contexto donde se accede a los contenidos antes de iniciar un nuevo aprendizaje. Poder ajustar esas variables de una manera personalizada favorece que las personas puedan recordar y almacenar en el hipocampo los conocimientos para retenerlos a largo plazo. Se trata de un modelo denominado *Neurocognitive context-dependent e-learning* que es aplicado de manera consciente en esta titulación universitaria.

Por otro lado, también en aras de favorecer al máximo el contacto mentor-alumno, se proporciona un amplio abanico de posibilidades de comunicación, tanto en tiempo real como en diferido (mensajería interna, foros de discusión, servicio de atención telefónica, email de contacto con secretaría técnica, chat y videoconferencia).

Asimismo, este completísimo Campus Virtual permitirá que el alumnado de TECH organice sus horarios de estudio de acuerdo con su disponibilidad personal o sus obligaciones laborales. De esa manera tendrá un control global de los contenidos académicos y sus herramientas didácticas, puestas en función de su acelerada actualización profesional.



*La modalidad de estudios online de este programa te permitirá organizar tu tiempo y tu ritmo de aprendizaje, adaptándolo a tus horarios"*

### La eficacia del método se justifica con cuatro logros fundamentales:

1. Los alumnos que siguen este método no solo consiguen la asimilación de conceptos, sino un desarrollo de su capacidad mental, mediante ejercicios de evaluación de situaciones reales y aplicación de conocimientos.
2. El aprendizaje se concreta de una manera sólida en capacidades prácticas que permiten al alumno una mejor integración en el mundo real.
3. Se consigue una asimilación más sencilla y eficiente de las ideas y conceptos, gracias al planteamiento de situaciones que han surgido de la realidad.
4. La sensación de eficiencia del esfuerzo invertido se convierte en un estímulo muy importante para el alumnado, que se traduce en un interés mayor en los aprendizajes y un incremento del tiempo dedicado a trabajar en el curso.



## La metodología universitaria mejor valorada por sus alumnos

Los resultados de este innovador modelo académico son constatables en los niveles de satisfacción global de los egresados de TECH.

La valoración de los estudiantes sobre la calidad docente, calidad de los materiales, estructura del curso y sus objetivos es excelente. No en valde, la institución se convirtió en la universidad mejor valorada por sus alumnos según el índice global score, obteniendo un 4,9 de 5.

*Accede a los contenidos de estudio desde cualquier dispositivo con conexión a Internet (ordenador, tablet, smartphone) gracias a que TECH está al día de la vanguardia tecnológica y pedagógica.*

*Podrás aprender con las ventajas del acceso a entornos simulados de aprendizaje y el planteamiento de aprendizaje por observación, esto es, Learning from an expert.*



Así, en este programa estarán disponibles los mejores materiales educativos, preparados a conciencia:



#### Material de estudio

Todos los contenidos didácticos son creados por los especialistas que van a impartir el curso, específicamente para él, de manera que el desarrollo didáctico sea realmente específico y concreto.

Estos contenidos son aplicados después al formato audiovisual que creará nuestra manera de trabajo online, con las técnicas más novedosas que nos permiten ofrecerte una gran calidad, en cada una de las piezas que pondremos a tu servicio.



#### Prácticas de habilidades y competencias

Realizarás actividades de desarrollo de competencias y habilidades específicas en cada área temática. Prácticas y dinámicas para adquirir y desarrollar las destrezas y habilidades que un especialista precisa desarrollar en el marco de la globalización que vivimos.



#### Resúmenes interactivos

Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

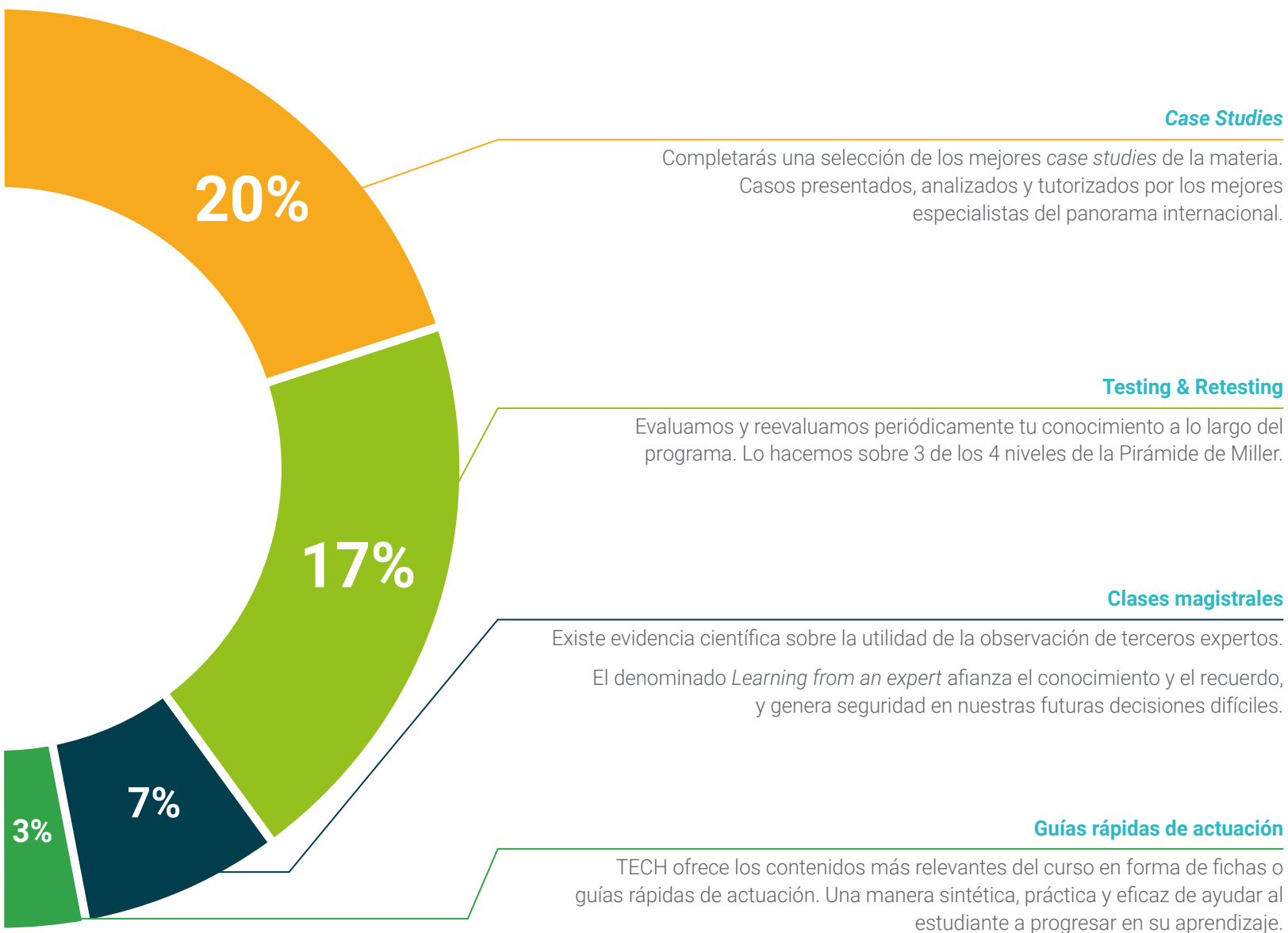
Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



#### Lecturas complementarias

Artículos recientes, documentos de consenso, guías internacionales... En nuestra biblioteca virtual tendrás acceso a todo lo que necesitas para completar tu capacitación.







# 10 Titulación

El Máster Semipresencial en Programación Front End Full Stack Developer garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Semipresencial expedido por TECH Global University.





“

*Supera con éxito este programa y recibe tu titulación universitaria sin desplazamientos ni farragosos trámites”*

Este programa te permitirá obtener el título propio de **Máster Semipresencial en Programación Front End Full Stack Developer** avalado por **TECH Global University**, la mayor Universidad digital del mundo.

**TECH Global University**, es una Universidad Oficial Europea reconocida públicamente por el Gobierno de Andorra ([boletín oficial](#)). Andorra forma parte del Espacio Europeo de Educación Superior (EEES) desde 2003. El EEES es una iniciativa promovida por la Unión Europea que tiene como objetivo organizar el marco formativo internacional y armonizar los sistemas de educación superior de los países miembros de este espacio. El proyecto promueve unos valores comunes, la implementación de herramientas conjuntas y fortaleciendo sus mecanismos de garantía de calidad para potenciar la colaboración y movilidad entre estudiantes, investigadores y académicos.

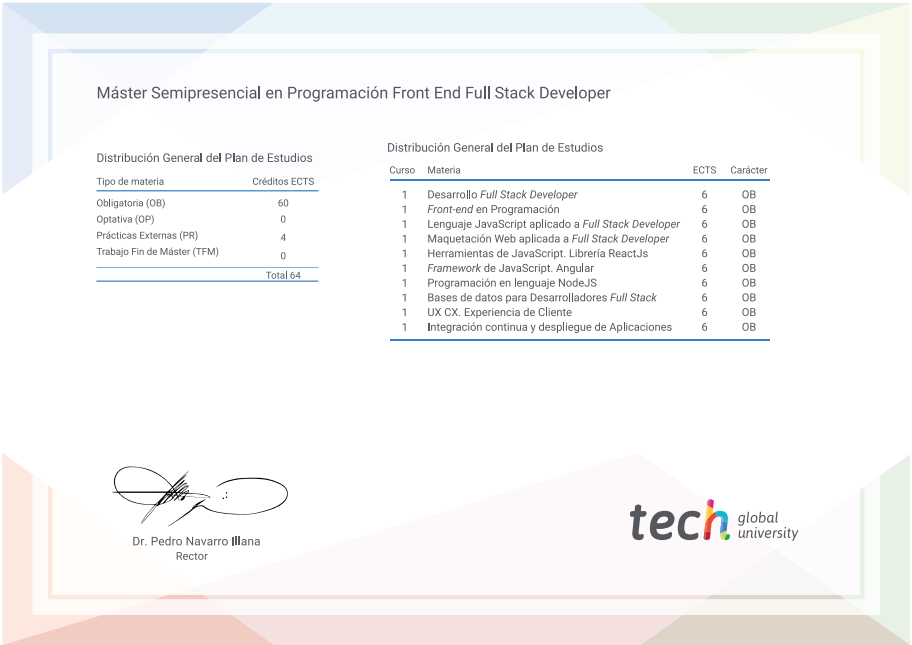
Este título propio de **TECH Global University**, es un programa europeo de formación continua y actualización profesional que garantiza la adquisición de las competencias en su área de conocimiento, confiriendo un alto valor curricular al estudiante que supere el programa.

Título: **Máster Semipresencial en Programación Front End Full Stack Developer**

Modalidad: **Semipresencial (Online + Prácticas)**

Duración: **12 meses**

Créditos: **60 + 4 ECTS**



\*Apostilla de La Haya. En caso de que el alumno solicite que su título en papel recabe la Apostilla de La Haya, TECH Global University realizará las gestiones oportunas para su obtención, con un coste adicional.



## Máster Semipresencial

### Programación Front End Full Stack Developer

Modalidad: Semipresencial (Online + Prácticas)

Duración: 12 meses

Titulación: TECH Global University

Créditos: 60 + 4 ECTS

# Máster Semipresencial

## Programación Front End

## Full Stack Developer