Máster Título Propio

Programación Front End Full Stack Developer



tech universidad tecnológica



tech universidad tecnológica

Máster Título Propio
Programación Front End Full Stack Developer

Modalidad: **Online**
Duración: **12 meses**
Titulación: **TECH Universidad Tecnológica**
Horas lectivas: **1.500 h.**
Acceso web: **www.techtitute.com/informatica/master/master-programacion-front-end-full-stack-developer**

Índice

01 Presentación — pág. 4

02 Objetivos — pág. 8

03 Competencias — pág. 14

04 Dirección del curso — pág. 18

05 Estructura y contenido — pág. 22

06 Metodología — pág. 34

07 Titulación — pág. 42

01

Presentación

Los desarrolladores *Front End* conforman una parte imprescindible de cualquier desarrollo web. Su naturaleza de unión entre la parte de diseño gráfico y la propia programación *Back End* les obliga a tener unos conocimientos muy específicos y desarrollados, que a su vez deben estar actualizados para poder enfrentarse a los retos cada vez más exigentes del presente. Siendo una especialidad que ofrece múltiples oportunidades a nivel profesional, TECH ha reunido al mejor equipo docente posible para desarrollar un programa universitario extenso, completo y metodológico. Incidiendo en las herramientas y metodologías más avanzadas en el desarrollo de soluciones web, este título ofrece el impulso necesario para dar el paso hacia la programación *Front End Full Stack*.





Conviértete en una pieza imprescindible en cualquier desarrollo web, liderando y coordinando equipos informáticos gracias a la perfeccionada metodología de trabajo y liderazgo de este Máster Título Propio"

El desarrollo *Full Stack* es una opción especialmente interesante para todos los profesionales de la informática que busquen impulsar significativamente sus carreras. Los conocimientos que se han de tener para desarrollar un buen papel en el sector son amplios, lo que hace que las oportunidades para prosperar e incluso liderar equipos de desarrollo sean múltiples.

Gracias a la exhaustividad con la que han sido desarrollados todos los contenidos de este programa, el egresado podrá dirigir su carrera hacia el desarrollo del *Front End* web, maquetado de páginas, especialista en experiencia de clientes o DevOps. Con una visión de 360º de todo el proceso de construcción de una aplicación/web, el informático será capaz de abordar cualquier tipo de proyecto, aportando además un desarrollo en los últimos avances en todos los procesos de ciclo de vida de un software.

Una oportunidad académica única con la que poder acceder a un conocimiento que aúna la teoría informática más reciente con la práctica profesional de primer nivel, proporcionada por un equipo docente de la más alta calidad. Su experiencia al frente de numerosos proyectos relevantes en el ámbito de la banca digital o las telecomunicaciones hacen que el contenido didáctico se enriquezca, aportando gran cantidad de casos reales y lecturas complementarias.

La gran flexibilidad de esta enseñanza es otra de sus características más destacadas. No existen ni horarios prefijados ni clases presenciales, siendo el propio alumno el que decide cuando, donde y como asumir toda la carga lectiva. La totalidad del contenido del aula virtual está disponible para su descarga, pudiendo estudiarse desde cualquier dispositivo con conexión a internet.

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

- El desarrollo de casos prácticos presentados por expertos en Programación Front End Full Stack Developer
- Los contenidos gráficos, esquemáticos y eminentemente prácticos con los que está concebido recogen una información científica y práctica sobre aquellas disciplinas indispensables para el ejercicio profesional
- Los ejercicios prácticos donde realizar el proceso de autoevaluación para mejorar el aprendizaje
- Su especial hincapié en metodologías innovadoras
- Las lecciones teóricas, preguntas al experto y trabajos de reflexión individual
- La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet

Inscríbete ya y no dejes pasar la oportunidad que te llevará al cénit en liderazgo y desarrollo de los proyectos informáticos más ambiciosos”

Llegarás a un nivel de especialización avanzado, siendo capaz de construir cualquier solución web requerida con una perspectiva de Experiencia de Cliente moderna y adaptada al mercado actual"

Profundizarás en las metodologías ágiles y como estas se pueden implementar en el proceso de desarrollo, aumentando tus capacidades y competencias transversales.

Tendrás acceso a un material didáctico amplio, que trata desde el lenguaje de programación Javascript hasta las herramientas como CSS, Angular y ReactJS.

El programa incluye, en su cuadro docente, a profesionales del sector que vierten en esta capacitación la experiencia de su trabajo, además de reconocidos especialistas de sociedades de referencia y universidades de prestigio.

Su contenido multimedia, elaborado con la última tecnología educativa, permitirá al profesional un aprendizaje situado y contextual, es decir, un entorno simulado que proporcionará una capacitación inmersiva programada para entrenarse ante situaciones reales.

El diseño de este programa se centra en el Aprendizaje Basado en Problemas, mediante el cual el profesional deberá tratar de resolver las distintas situaciones de práctica profesional que se le planteen a lo largo del programa académico. Para ello, contará con la ayuda de un novedoso sistema de vídeo interactivo realizado por reconocidos expertos.



02

Objetivos

El objetivo de este Máster Título Propio en Programación Front End Full Stack Developer, teniendo en cuenta las múltiples opciones que ofrece el desarrollo, no es otro que el de proporcionar los conocimientos y técnicas más avanzados en este ámbito. De esta forma, el informático puede incluso empezar a desarrollar sus propios proyectos o impulsar su trayectoria profesional antes de finalizar la titulación, gracias al enfoque eminentemente práctico de todos los contenidos proporcionados.





Cumplirás tus objetivos profesionales más ambiciosos gracias al enfoque distintivo de este programa, que te llevará por todas las etapas del desarrollo Front End Full Stack"

Objetivos generales

- Generar conocimiento especializado sobre aspectos clave de programación
- Fomentar el pensamiento de forma algorítmica
- Dotar de las herramientas y habilidades necesarias para el desarrollo
- Promover la adopción de metodologías ágiles para la ejecución de proyecto
- Desarrollar conocimiento especializado sobre los fundamentos de la web
- Fomentar el uso de técnicas y herramientas de desarrollo *Front End* modernas
- Desarrollar un diseño web para abordar la maquetación correctamente
- Evaluar los conocimientos adquiridos





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 aplicado 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 *Context*
- 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 NodeJSÍndice

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

Mejorarás tus habilidades y competencias de forma progresiva, a lo largo de 10 módulos elaborados desde los conocimientos más sólidos y la experiencia contrastada de todos los docentes"

03

Competencias

Las competencias que debe desarrollar un programador *Front End Full Stack* son múltiples, especialmente en un campo tan disputado y especializado como lo es el de la informática. Por ello, el temario abarca los diferentes tipos de lenguaje más comunes en este ámbito, así como las herramientas y filosofía de trabajo que debe seguir el informático para destacar con buena distinción profesional. Esto es posible gracias a la propia naturaleza multidisciplinar del propio equipo docente, que ha puesto en común todo su conocimiento respecto a diversas áreas del desarrollo *Front End*.

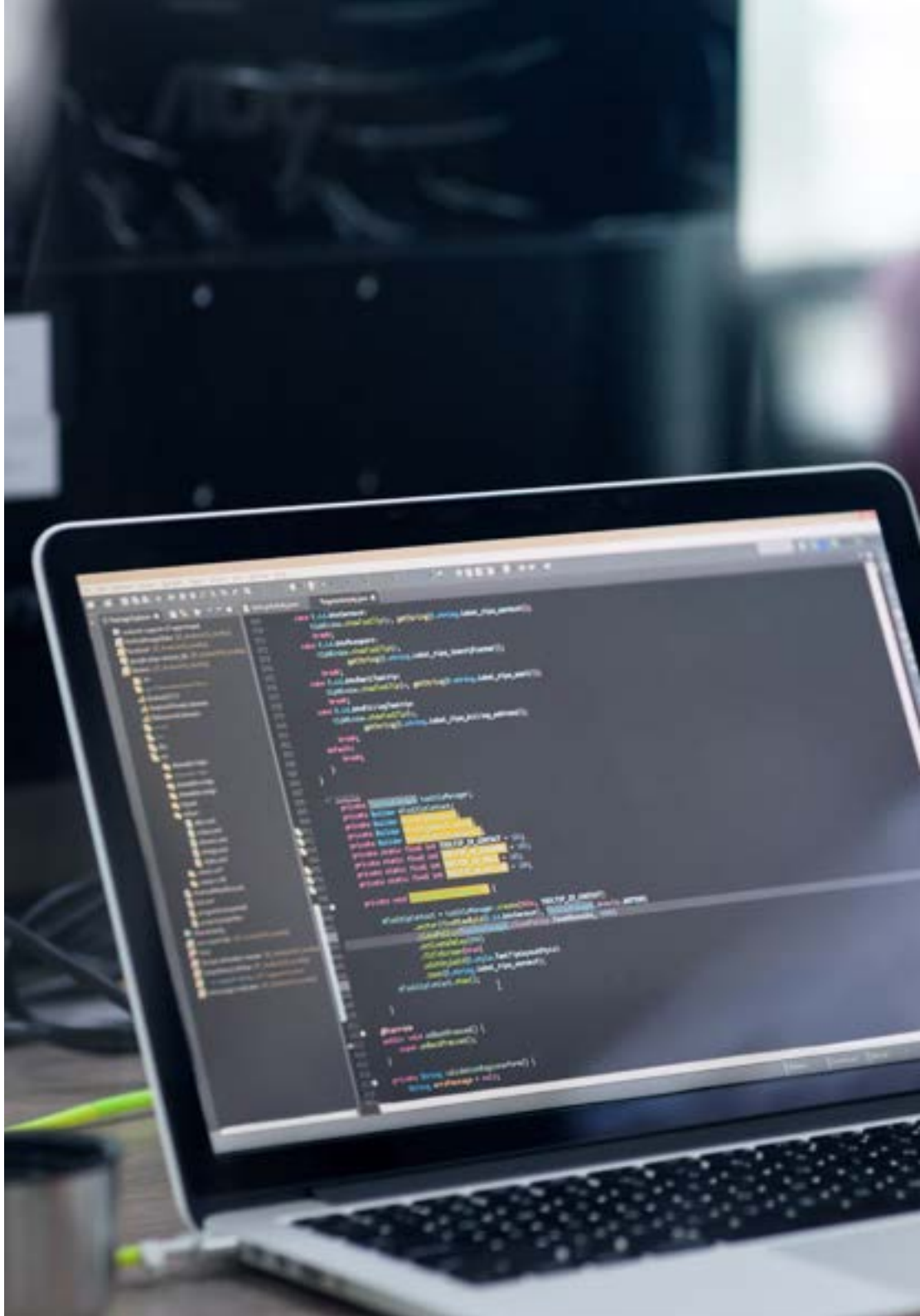




Le darás un impulso de gran calidad a tu CV incorporando multitud de habilidades y competencias requeridas en los ámbitos más elevados del desarrollo software”

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

Desarrollarás a lo largo de todo el temario las habilidades necesarias para triunfar en el desarrollo Front End, potenciando no solo tus propios conocimientos sino también tus competencias transversales"

04

Dirección del curso

TECH pone especial cuidado en la selección del personal docente a cargo de sus titulaciones. La exhaustividad con la que se conforman los equipos garantiza no solo la calidad de los propios contenidos, sino también la implicación de un equipo comprometido con la mejora profesional del informático. El alumno encontrará, por tanto, un material didáctico adaptado a sus criterios más exigentes, con todas las ayudas y material adicional esperables de la mayor institución académica online del mundo.





Podrás consultar todas tus dudas directamente con el personal docente, obteniendo una tutoría personalizada y adaptada a tus necesidades”

Dirección



D. Olalla Bonal, Martín

- Especialista técnico en *Blockchain* en IBM SPGI
- Especialista en ventas técnicas en *Blockchain*. IBM
- Director de Arquitectura. *Blocknitive*
- Técnico en electrónica digital
- *Blockchain Architect* -IT *Infrastructure Architect* - Gestor de proyectos IT. Ámbitos de negocio: Software, Infraestructura, Telecomunicaciones

Profesores

D. Calzada Martínez, Jesús

- Ingeniero de Software Senior en Devo
- Desarrollador *Full Stack* en Blocknitive
- Responsable de *Front End* en Infinia
- Desarrollador *Full Stack* en Resem
- Desarrollador Java en Hitec
- Graduado en Ingeniería de Ordenadores

D. Guerrero Díaz-Pintado, Arturo

- Ingeniero Técnico de Preventa a través de Watson *Customer Engagement portfolio* (soluciones de Marketing *y Customer Experience)* dentro de España, Portugal, Grecia e Israel en IBM
- Ingeniero de redes de I+D en Telefónica
- Consultor de servicios profesionales trabajando con organizaciones de vanguardia en Europa, Oriente Medio y América Latina desde IBM
- Graduado en Ingeniería Superior de Telecomunicación por la Universidad de Alcalá y la *Danish Technical University*
- Destacas colaboraciones en reconocidas universidades y centros de educación superior en materias relacionadas con la tecnología como Inteligencia Artificial, *Internet of Things, Cloud, Customer Experience y Digital Transformation*

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 en la Universidad Politécnica de Madrid
- Graduado en Ingeniería de Electrónica de Comunicaciones en la Universidad Politécnica de Madrid

D. Reyes Oliva, Luis

- Promotor de desarrollo y arquitecto cloud en IBM
- Responsable técnico de clientes para cuentas integradas del BBVA en IBM
- *Cloud Executive Selling* en IBM
- Arquitecto Cloud y DevOps en IBM
- Arquitecto de software para clientes en Telefónica
- Arquitecto de soluciones técnicas para Rational
- *Software Engineering Manager* en Borland
- Desarrollo Software y responsable de control de calidad en Altana Consulting
- Licenciado en Ingeniería Informática en la Universidad Pontificia de Salamanca, en Madrid

D. Frias Favero, Pedro Luis

- CTO - Swearit Technologies
- COO - Key Identification
- Experto en *blockchain* y aplicaciones descentralizadas - Universidad de Alcalá
- Full Stack Developer - Ironhack
- Ingeniero Industrial egresado de la universidad Yacambu

D. Gómez Rodríguez, Antonio

- Ingeniero de soluciones Cloud en Oracle
- Director de Proyectos en Sopra Group
- Director de Proyectos en Everis
- Jefe de Proyectos en Empresa pública de Gestión de Programas Culturales. Consejería de Cultura de Andalucía
- Analista de Sistemas de Información. Sopra Group
- Licenciado en Ingeniería Superior de Telecomunicación por la Universidad Politécnica de Cataluña
- Postgrado en Tecnologías y Sistemas de Información, Instituto Catalán de Tecnología
- *E-Business Master*, Escuela de Negocios La Salle

05

Estructura y contenido

Dado que el esfuerzo y la inversión que realiza el informático en optar por un programa de estas características no son menores, TECH apuesta por una metodología pedagógica de primer nivel, contrastada en eficacia y rentabilidad. El *Relearning* permite al alumno adquirir los conocimientos más importantes del programa a lo largo de todo el temario, evitando así un tiempo de estudio valioso que se puede aprovechar en los múltiples materiales complementarios que acompañan a cada tema propuesto.





Indaga en los diferentes módulos y temas de conocimiento a través de vídeo resúmenes, vídeos en detalle y vídeos motivacionales creados por los propios docentes"

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. *Calbacks*
- 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. *Component Context* en ReactJS
- 5.5.1. *Component 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. Creacion 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 AI al proceso de CI/CD
- 10.10.2. DevSecOps
- 10.10.3. *Chaos Engineering*

Dispondrás de los medios educativos más modernos, con un acceso libre al aula virtual las 24 horas del día”

06

Metodología

Este programa de capacitación ofrece una forma diferente de aprender. Nuestra metodología se desarrolla a través de un modo de aprendizaje de forma cíclica: ***el Relearning***.

Este sistema de enseñanza es utilizado, por ejemplo, en las facultades de medicina más prestigiosas del mundo y se ha considerado uno de los más eficaces por publicaciones de gran relevancia como el ***New England Journal of Medicine.***





Descubre el Relearning, un sistema que abandona el aprendizaje lineal convencional para llevarte a través de sistemas cíclicos de enseñanza: una forma de aprender que ha demostrado su enorme eficacia, especialmente en las materias que requieren memorización"

Estudio de Caso para contextualizar todo el contenido

Nuestro programa ofrece un método revolucionario de desarrollo de habilidades y conocimientos. Nuestro objetivo es afianzar competencias en un contexto cambiante, competitivo y de alta exigencia.

Con TECH podrás experimentar una forma de aprender que está moviendo los cimientos de las universidades tradicionales de todo el mundo"



Accederás a un sistema de aprendizaje basado en la reiteración, con una enseñanza natural y progresiva a lo largo de todo el temario.



El alumno aprenderá, mediante actividades colaborativas y casos reales, la resolución de situaciones complejas en entornos empresariales reales.

Un método de aprendizaje innovador y diferente

El presente programa de TECH es una enseñanza intensiva, creada desde 0, que propone los retos y decisiones más exigentes en este campo, ya sea en el ámbito nacional o internacional. Gracias a esta metodología se impulsa el crecimiento personal y profesional, dando un paso decisivo para conseguir el éxito. El método del caso, técnica que sienta las bases de este contenido, garantiza que se sigue la realidad económica, social y profesional más vigente.

> *Nuestro programa te prepara para afrontar nuevos retos en entornos inciertos y lograr el éxito en tu carrera"*

El método del caso ha sido el sistema de aprendizaje más utilizado por las mejores escuelas de Informática del mundo desde que éstas existen. Desarrollado en 1912 para que los estudiantes de Derecho no solo aprendiesen las leyes a base de contenidos teóricos, el método del caso consistió en presentarles situaciones complejas reales para que tomasen decisiones y emitiesen juicios de valor fundamentados sobre cómo resolverlas. En 1924 se estableció como método estándar de enseñanza en Harvard.

Ante una determinada situación, ¿qué debería hacer un profesional? Esta es la pregunta a la que te enfrentamos en el método del caso, un método de aprendizaje orientado a la acción. A lo largo del curso, los estudiantes se enfrentarán a múltiples casos reales. Deberán integrar todos sus conocimientos, investigar, argumentar y defender sus ideas y decisiones.

Relearning Methodology

TECH aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina elementos didácticos diferentes en cada lección.

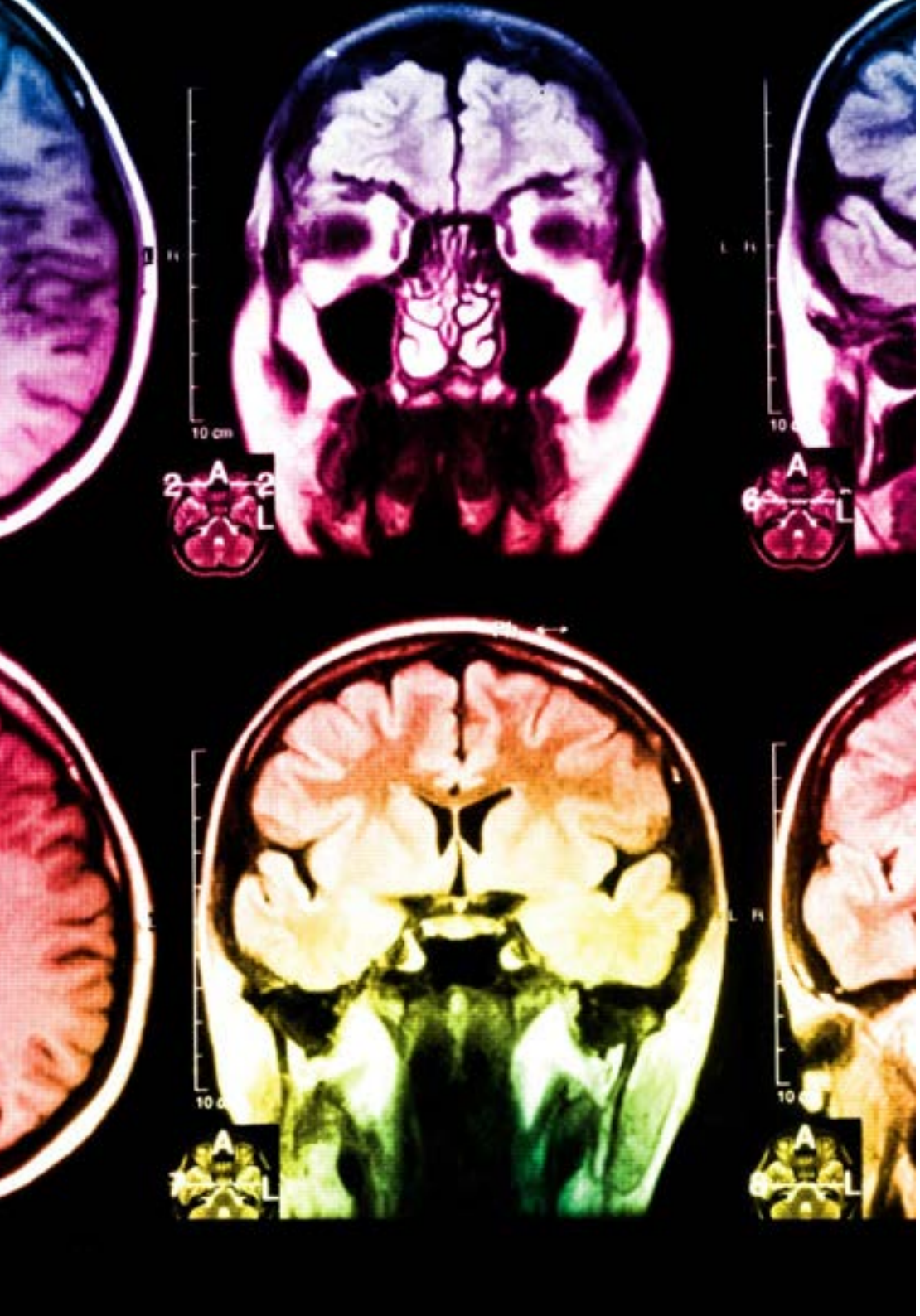
Potenciamos el Estudio de Caso con el mejor método de enseñanza 100% online: el Relearning.

En 2019 obtuvimos los mejores resultados de aprendizaje de todas las universidades online en español en el mundo.

En TECH aprenderás con una metodología vanguardista concebida para capacitar a los directivos del futuro. Este método, a la vanguardia pedagógica mundial, se denomina Relearning.

Nuestra universidad es la única en habla hispana licenciada para emplear este exitoso método. En 2019, conseguimos mejorar los niveles de satisfacción global de nuestros alumnos (calidad docente, calidad de los materiales, estructura del curso, objetivos…) con respecto a los indicadores de la mejor universidad online en español.





En nuestro programa, el aprendizaje no es un proceso lineal, sino que sucede en espiral (aprender, desaprender, olvidar y reaprender). Por eso, se combinan cada uno de estos elementos de forma concéntrica. Con esta metodología se han capacitado más de 650.000 graduados universitarios con un éxito sin precedentes en ámbitos tan distintos como la bioquímica, la genética, la cirugía, el derecho internacional, las habilidades directivas, las ciencias del deporte, la filosofía, el derecho, la ingeniería, el periodismo, la historia o los mercados e instrumentos financieros. Todo ello en un entorno de alta exigencia, con un alumnado universitario de un perfil socioeconómico alto y una media de edad de 43,5 años.

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

A partir de la última evidencia científica en el ámbito de la neurociencia, no solo sabemos organizar la información, las ideas, las imágenes y los recuerdos, sino que sabemos que el lugar y el contexto donde hemos aprendido algo es fundamental para que seamos capaces de recordarlo y almacenarlo en el hipocampo, para retenerlo en nuestra memoria a largo plazo.

De esta manera, y en lo que se denomina Neurocognitive context-dependent e-learning, los diferentes elementos de nuestro programa están conectados con el contexto donde el participante desarrolla su práctica profesional.

Este programa ofrece los mejores materiales educativos, preparados a conciencia para los profesionales:

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, para crear el método de trabajo online de TECH. Todo ello, con las técnicas más novedosas que ofrecen piezas de gran calidad en todos y cada uno los materiales que se ponen a disposición del alumno.

Clases magistrales

Existe evidencia científica sobre la utilidad de la observación de terceros expertos.

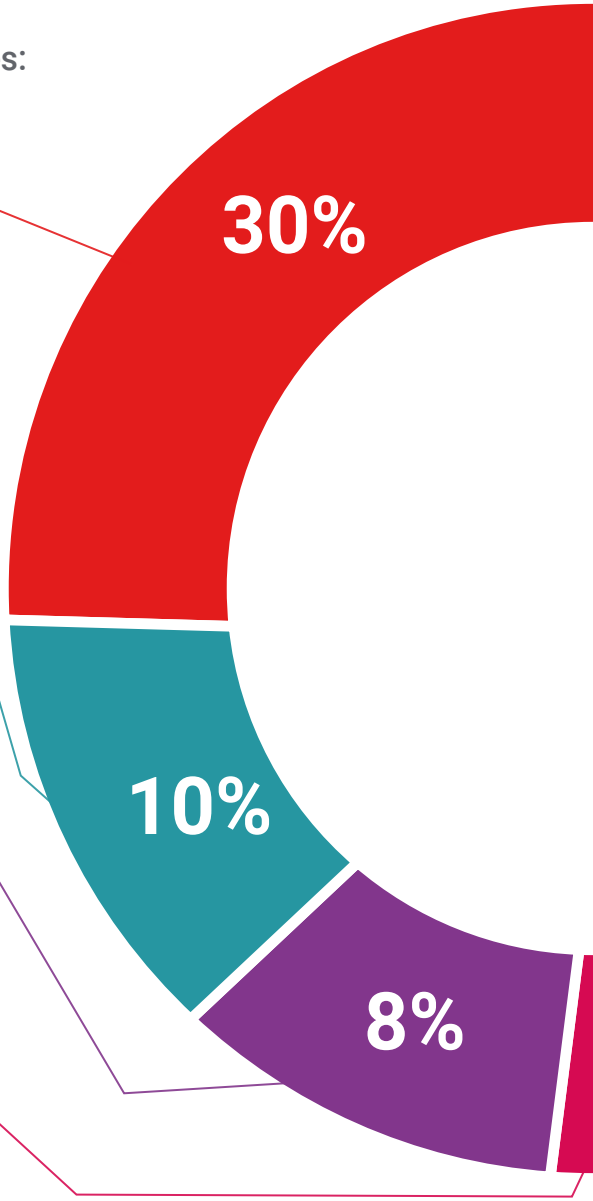
El denominado Learning from an Expert afianza el conocimiento y el recuerdo, y genera seguridad en las futuras decisiones difíciles.

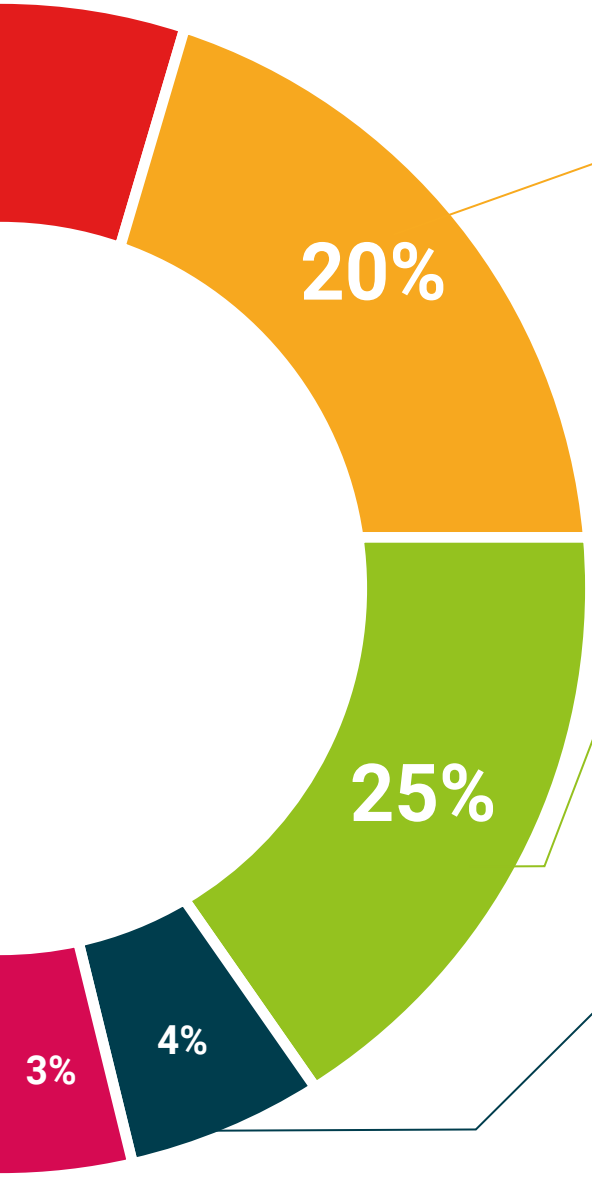
Prácticas de habilidades y competencias

Realizarán 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.

Lecturas complementarias

Artículos recientes, documentos de consenso y guías internacionales, entre otros. En la biblioteca virtual de TECH el estudiante tendrá acceso a todo lo que necesita para completar su capacitación.





Case studies

Completarán una selección de los mejores casos de estudio elegidos expresamente para esta titulación. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.

Resúmenes interactivos

El equipo de TECH presenta los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audios, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

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

Testing & Retesting

Se evalúan y reevalúan periódicamente los conocimientos del alumno a lo largo del programa, mediante actividades y ejercicios evaluativos y autoevaluativos para que, de esta manera, el estudiante compruebe cómo va consiguiendo sus metas.

07

Titulación

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





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

Este **Máster Título Propio en Programación Front End Full Stack Developer** contiene el programa más completo y actualizado del mercado.

Tras la superación de la evaluación, el alumno recibirá por correo postal* con acuse de recibo su correspondiente título de **Máster Propio** emitido por **TECH Universidad Tecnológica.**

El título expedido por **TECH Universidad Tecnológica** expresará la calificación que haya obtenido en el Máster Título Propio, y reunirá los requisitos comúnmente exigidos por las bolsas de trabajo, oposiciones y comités evaluadores de carreras profesionales.

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

N.º Horas Oficiales: **1.500 h.**


tech universidad tecnológica

Otorga la presente

CONSTANCIA

a

C. _____, con documento de identificación nº_____

Por haber superado con éxito y acreditado el programa de

MÁSTER TÍTULO PROPIO

en

Programación Front End Full Stack Developer

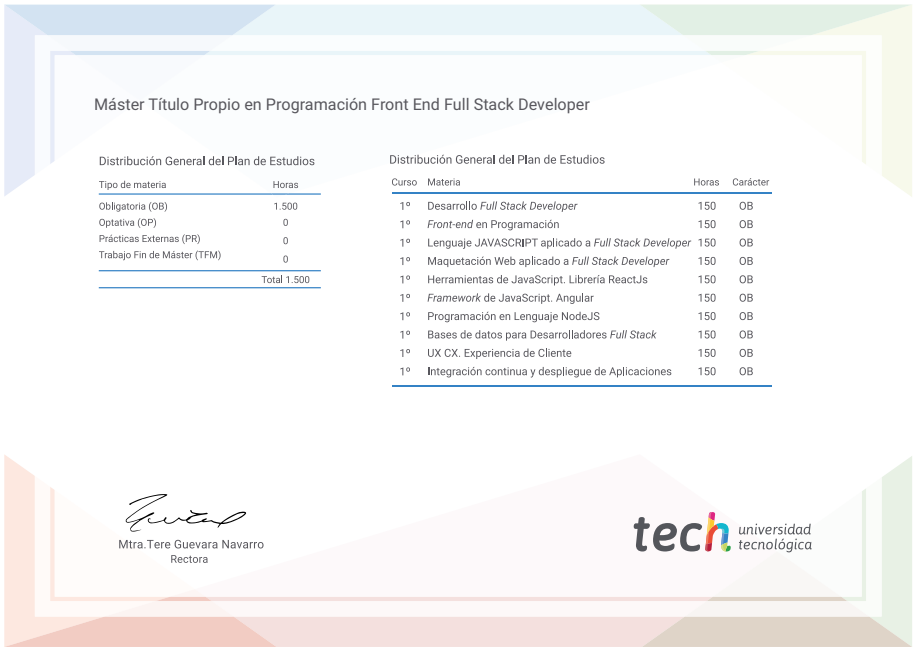
Se trata de un título propio de esta Universidad con una duración de 1.500 horas, con fecha de inicio dd/mm/aaaa y fecha de finalización dd/mm/aaaa.

TECH es una Institución Particular de Educación Superior reconocida por la Secretaría de Educación Pública a partir del 28 de junio de 2018.

A 17 de junio de 2020

Mtra.Tere Guevara Navarro
Rectora

código único TECH: AFWOR23S techtitute.com/titulos


Máster Título Propio en Programación Front End Full Stack Developer

Distribución General del Plan de Estudios

Tipo de materia	Horas
Obligatoria (OB)	1.500
Optativa (OP)	0
Prácticas Externas (PR)	0
Trabajo Fin de Máster (TFM)	0
	Total 1.500

Distribución General del Plan de Estudios

Curso	Materia	Horas	Carácter
1º	Desarrollo *Full Stack Developer*	150	OB
1º	*Front-end* en Programación	150	OB
1º	Lenguaje JAVASCRIPT aplicado a *Full Stack Developer*	150	OB
1º	Maquetación Web aplicado a *Full Stack Developer*	150	OB
1º	Herramientas de JavaScript. Librería ReactJs	150	OB
1º	*Framework* de JavaScript. Angular	150	OB
1º	Programación en Lenguaje NodeJS	150	OB
1º	Bases de datos para Desarrolladores *Full Stack*	150	OB
1º	UX CX. Experiencia de Cliente	150	OB
1º	Integración continua y despliegue de Aplicaciones	150	OB

Mtra.Tere Guevara Navarro
Rectora

tech universidad tecnológica

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

tech universidad tecnológica

Máster Título Propio

Programación Front End Full Stack Developer

Modalidad: **Online**

Duración: **12 meses**

Titulación: **TECH Universidad Tecnológica**

Horas lectivas: **1.500 h.**

Máster Título Propio

Programación Front End Full Stack Developer



tech universidad tecnológica