

Máster Título Propio

Programación Front End Full Stack Developer



Máster Título Propio Programación Front End Full Stack Developer

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Universidad FUNDEPOS
- » Acreditación: 60 ECTS
- » Horario: a tu ritmo
- » Exámenes: online

Acceso web: www.techtute.com/informatica/master/master-programacion-front-end-full-stack-developer

Índice

01

Presentación del programa

pág. 4

02

¿Por qué estudiar en TECH?

pág. 8

03

Plan de estudios

pág. 12

04

Objetivos docentes

pág. 24

05

Salidas profesionales

pág. 30

06

Metodología de estudio

pág. 34

07

Cuadro docente

pág. 44

08

Titulación

pág. 48

01

Presentación del programa

La Programación *Front End Full Stack Developer* representa hoy una de las competencias más demandadas en el ámbito digital, en un entorno donde el diseño visual y la funcionalidad del servidor convergen para dar vida a plataformas dinámicas. Según datos de la Organización de las Naciones Unidas, los perfiles relacionados con el desarrollo web figuran entre los más requeridos en el sector tecnológico, con una tendencia de crecimiento sostenido en los últimos años. Ante este panorama, este programa universitario de TECH surge como una respuesta académica. Mediante una metodología 100% online y el uso de material didáctico actualizado, se facilitará una experiencia autónoma, interactiva y alineada con los desafíos actuales del mercado.



“

Gracias a este programa 100% online, dominarás las herramientas y lenguajes más innovadores en Programación Front End Full Stack Developer para desarrollar soluciones web funcionales”

En el contexto digital actual, dominar los fundamentos del desarrollo web se ha vuelto indispensable para construir soluciones dinámicas, intuitivas y adaptadas a las demandas del entorno tecnológico. Por este motivo, la especialización en *Front End Full Stack Developer* representa una vía estratégica para quienes buscan crear experiencias interactivas desde la interfaz hasta la lógica de servidor. De hecho, esta área no solo permite desarrollar productos digitales funcionales, sino también garantizar su rendimiento, accesibilidad y escalabilidad, aspectos clave en el mercado competitivo y orientado al usuario que prevalece hoy en día.

Por lo tanto, TECH ha diseñado un innovador plan de estudios que profundizará en aspectos esenciales como los roles que intervienen en los entornos de programación, el diseño e interpretación de algoritmos eficientes y el uso de repositorios de software para una gestión colaborativa del código. A través de un enfoque riguroso, se articularán estos contenidos con herramientas actuales del sector, favoreciendo una comprensión integral de los procesos que intervienen en el desarrollo completo de aplicaciones web.

Posteriormente, este programa universitario brindará a los profesionales una oportunidad única para perfeccionar sus competencias en entornos web, consolidando habilidades que permiten construir proyectos tecnológicos desde cero y liderar iniciativas digitales con autonomía. Además, fortalecerán la toma de decisiones técnicas, el pensamiento lógico y la capacidad de adaptar soluciones a distintos contextos empresariales. Gracias a una preparación alineada con las exigencias del mercado, se favorecerá una proyección profesional sólida en un campo con amplia demanda laboral a nivel global.

Finalmente, la metodología de TECH se caracteriza por su flexibilidad, permitiendo acceder a todos los contenidos en línea en cualquier momento del día, desde cualquier lugar y mediante múltiples dispositivos conectados a internet. Este modelo se apoya en el innovador sistema *Relearning*, que favorecerá una asimilación progresiva y duradera del conocimiento mediante la repetición inteligente de conceptos clave, reforzando la capacitación autónoma y dinámica.

Este **Máster Título Propio en Programación Front End Full Stack Developer** contiene el programa universitario 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 están concebidos 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 en sistemas informáticos
- ♦ Las lecciones teóricas, preguntas al experto, foros de discusión de temas controvertidos y trabajos de reflexión individual
- ♦ La disponibilidad de acceso a los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet



Un plan de estudios basado en el disruptivo sistema del Relearning de TECH, que te facilitará la asimilación de conceptos complejos de un modo rápido y flexible”



Ahondarás en los roles que intervienen en los entornos de programación, adaptando su comprensión a distintos contextos de desarrollo digital

Incluye en su cuadro docente a profesionales pertenecientes al ámbito de la Programación Front End Full Stack Developer, que vierten en este programa 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 un estudio inmersivo programado para entrenarse ante situaciones reales.

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

Serás capaz de automatizar procesos de desarrollo y despliegue con herramientas DevOps.

Manejarás el uso de repositorios de software, optimizando la gestión colaborativa de código.



02

¿Por qué estudiar en TECH?

TECH es la mayor Universidad digital del mundo. Con un impresionante catálogo de más de 14.000 programas universitarios, disponibles en 11 idiomas, se posiciona como líder en empleabilidad, con una tasa de inserción laboral del 99%. Además, cuenta con un enorme claustro de más de 6.000 profesores de máximo prestigio internacional.



“

Estudia en la mayor universidad digital del mundo y asegura tu éxito profesional. El futuro empieza en TECH”

La mejor universidad online del mundo según FORBES

La prestigiosa revista Forbes, especializada en negocios y finanzas, ha destacado a TECH como «la mejor universidad online del mundo». Así lo han hecho constar recientemente en un artículo de su edición digital en el que se hacen eco del caso de éxito de esta institución, «gracias a la oferta académica que ofrece, la selección de su personal docente, y un método de aprendizaje innovador orientado a formar a los profesionales del futuro».

El mejor claustro docente top internacional

El claustro docente de TECH está integrado por más de 6.000 profesores de máximo prestigio internacional. Catedráticos, investigadores y altos ejecutivos de multinacionales, entre los cuales se destacan Isaiah Covington, entrenador de rendimiento de los Boston Celtics; Magda Romanska, investigadora principal de MetaLAB de Harvard; Ignacio Wistumba, presidente del departamento de patología molecular traslacional del MD Anderson Cancer Center; o D.W Pine, director creativo de la revista TIME, entre otros.

La mayor universidad digital del mundo

TECH es la mayor universidad digital del mundo. Somos la mayor institución educativa, con el mejor y más amplio catálogo educativo digital, cien por cien online y abarcando la gran mayoría de áreas de conocimiento. Ofrecemos el mayor número de titulaciones propias, titulaciones oficiales de posgrado y de grado universitario del mundo. En total, más de 14.000 títulos universitarios, en once idiomas distintos, que nos convierten en la mayor institución educativa del mundo.



Forbes
Mejor universidad
online del mundo

Plan
de estudios
más completo

Profesorado
TOP
Internacional


La metodología
más eficaz

nº1
Mundial
Mayor universidad
online del mundo

Los planes de estudio más completos del panorama universitario

TECH ofrece los planes de estudio más completos del panorama universitario, con temarios que abarcan conceptos fundamentales y, al mismo tiempo, los principales avances científicos en sus áreas científicas específicas. Asimismo, estos programas son actualizados continuamente para garantizar al alumnado la vanguardia académica y las competencias profesionales más demandadas. De esta forma, los títulos de la universidad proporcionan a sus egresados una significativa ventaja para impulsar sus carreras hacia el éxito.

Un método de aprendizaje único

TECH es la primera universidad que emplea el *Relearning* en todas sus titulaciones. Se trata de la mejor metodología de aprendizaje online, acreditada con certificaciones internacionales de calidad docente, dispuestas por agencias educativas de prestigio. Además, este disruptivo modelo académico se complementa con el "Método del Caso", configurando así una estrategia de docencia online única. También en ella se implementan recursos didácticos innovadores entre los que destacan vídeos en detalle, infografías y resúmenes interactivos.

La universidad online oficial de la NBA

TECH es la universidad online oficial de la NBA. Gracias a un acuerdo con la mayor liga de baloncesto, ofrece a sus alumnos programas universitarios exclusivos, así como una gran variedad de recursos educativos centrados en el negocio de la liga y otras áreas de la industria del deporte. Cada programa tiene un currículo de diseño único y cuenta con oradores invitados de excepción: profesionales con una distinguida trayectoria deportiva que ofrecerán su experiencia en los temas más relevantes.

Líderes en empleabilidad

TECH ha conseguido convertirse en la universidad líder en empleabilidad. El 99% de sus alumnos obtienen trabajo en el campo académico que ha estudiado, antes de completar un año luego de finalizar cualquiera de los programas de la universidad. Una cifra similar consigue mejorar su carrera profesional de forma inmediata. Todo ello gracias a una metodología de estudio que basa su eficacia en la adquisición de competencias prácticas, totalmente necesarias para el desarrollo profesional.



Google Partner Premier

El gigante tecnológico norteamericano ha otorgado a TECH la insignia Google Partner Premier. Este galardón, solo al alcance del 3% de las empresas del mundo, pone en valor la experiencia eficaz, flexible y adaptada que esta universidad proporciona al alumno. El reconocimiento no solo acredita el máximo rigor, rendimiento e inversión en las infraestructuras digitales de TECH, sino que también sitúa a esta universidad como una de las compañías tecnológicas más punteras del mundo.



La universidad mejor valorada por sus alumnos

Los alumnos han posicionado a TECH como la universidad mejor valorada del mundo en los principales portales de opinión, destacando su calificación más alta de 4,9 sobre 5, obtenida a partir de más de 1.000 reseñas. Estos resultados consolidan a TECH como la institución universitaria de referencia a nivel internacional, reflejando la excelencia y el impacto positivo de su modelo educativo.



03

Plan de estudios

Este exclusivo itinerario académico ahondará en temas de gran relevancia, que complementan la Programación *Front End Full Stack Developer*. Por lo tanto, se ahondará progresivamente en herramientas como el lenguaje HTML, esencial para estructurar correctamente la información en entornos digitales. A su vez, se profundizará en el diseño de *layouts* modernos, con énfasis en la adaptabilidad y la experiencia del usuario, elementos cruciales en el desarrollo de interfaces eficientes. De manera complementaria, se incorporará el dominio del lenguaje JavaScript, indispensable para implementar funcionalidades dinámicas. Esta combinación de conocimientos impulsa un conocimiento técnico riguroso, alineado con las necesidades del sector tecnológico.



“

Profundizarás en los principios estructurales más recientes del lenguaje HTML, aplicados al desarrollo de interfaces funcionales”

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: Prototipo 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

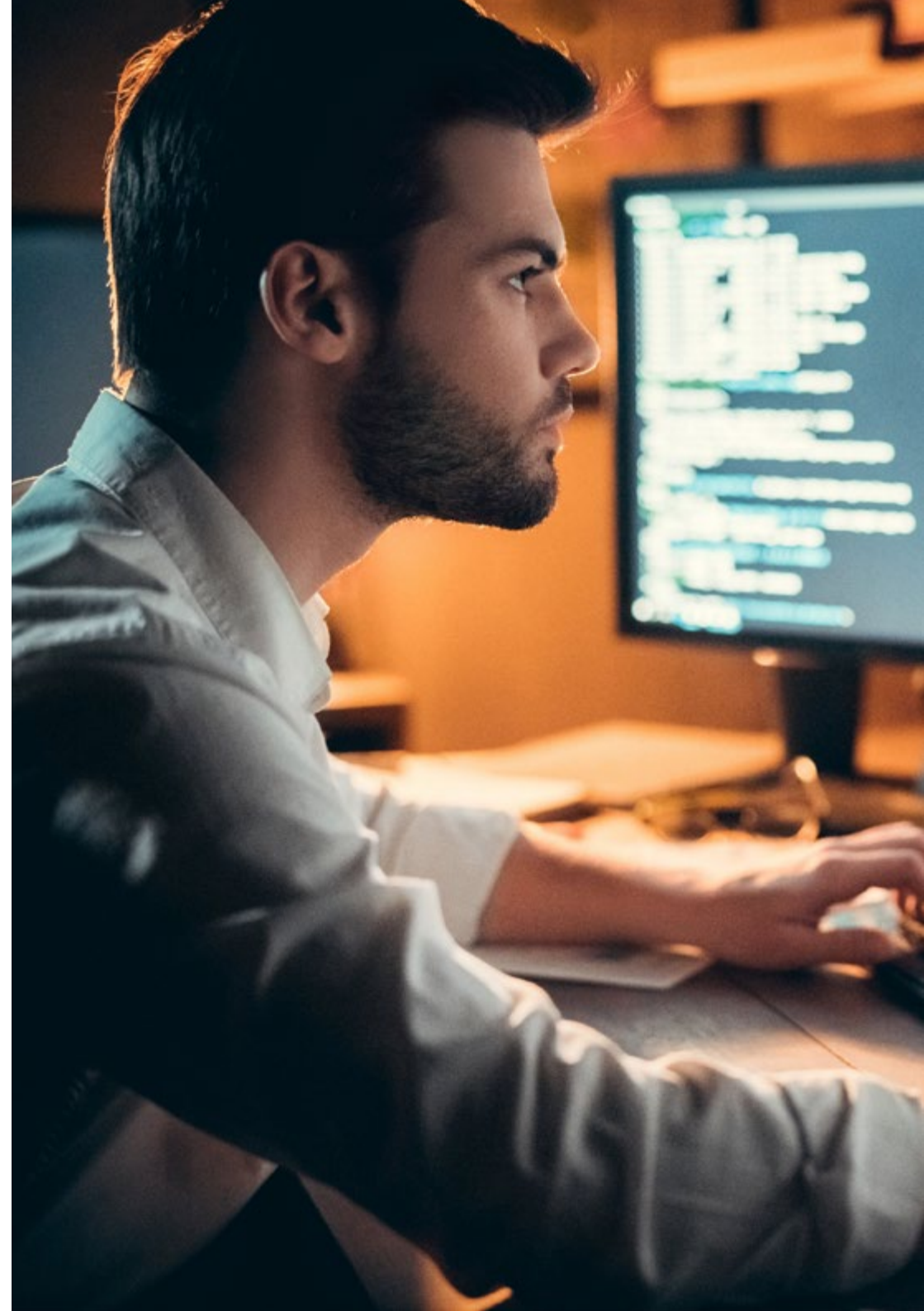
- 3.10. Programación asíncrona
 - 3.10.1. La programación asíncrona
 - 3.10.2. *Event loop*
 - 3.10.3. *Callbacks*
 - 3.10.4. *Promises*
 - 3.10.5. *Async/await*

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. *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. createContext*
 - 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. Suscripciones
 - 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 AI al proceso de CI/CD
 - 10.10.2. DevSecOps
 - 10.10.3. *Chaos engineering*



Ponte al día en el uso de bases de datos orientadas a desarrolladores Full Stack, clave para gestionar información de manera eficiente en entornos web

04

Objetivos docentes

Este programa universitario tiene como enfoque principal el fortalecimiento de competencias técnicas que permitan a los profesionales desenvolverse con solvencia en el desarrollo de soluciones digitales. Para ello, se priorizará el dominio de lenguajes de programación y *frameworks* actuales, junto con una comprensión profunda de algoritmos y estructuras de datos aplicadas a entornos reales. Gracias a este enfoque, se promoverá la capacidad de diseñar sistemas funcionales y alineados con las necesidades del sector tecnológico.



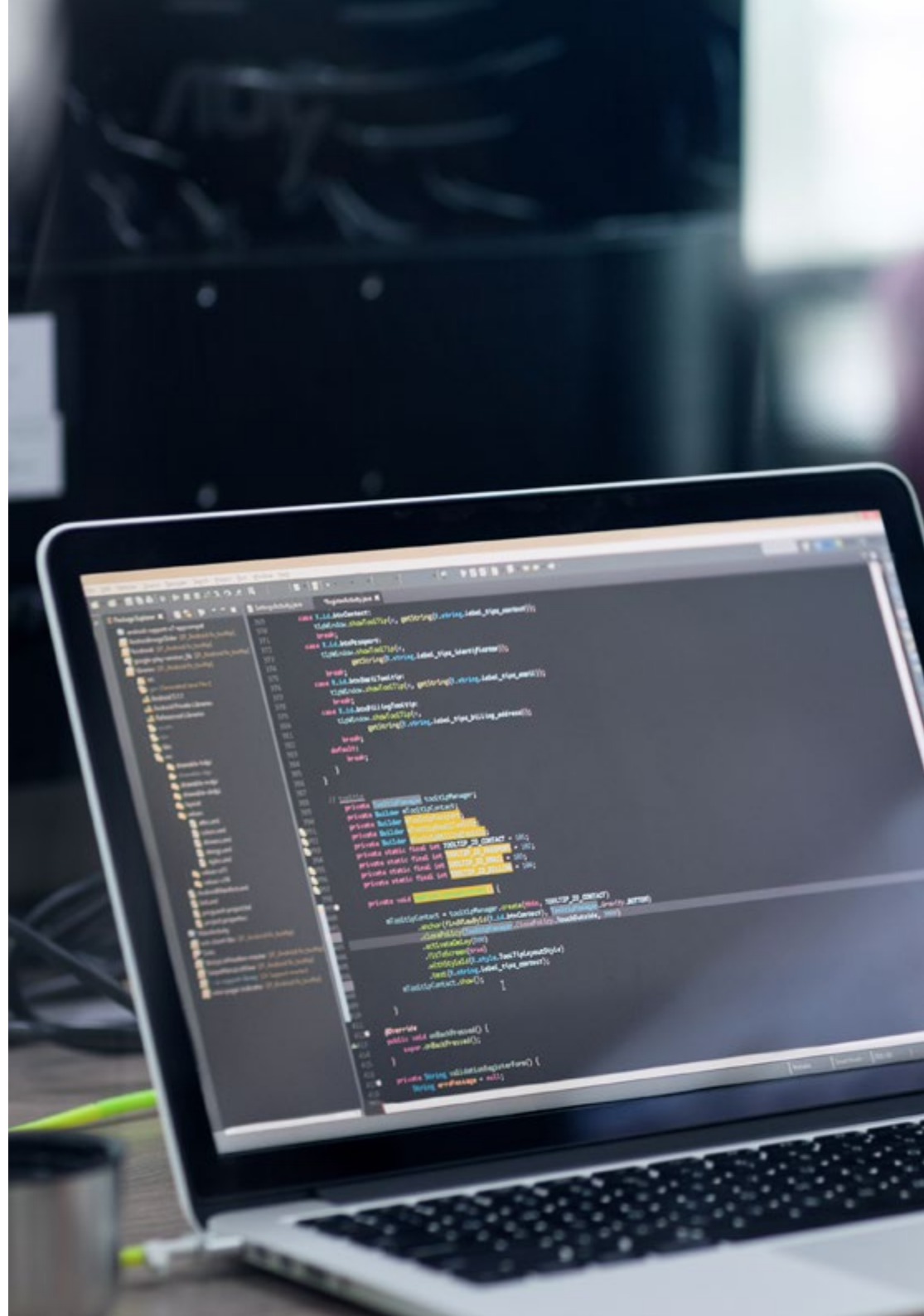
“

Evaluarás el impacto del uso de frameworks actuales en entornos de programación, identificando sus ventajas operativas”



Objetivos generales

- ♦ Desarrollar competencias integrales en el entorno *Full Stack*, abarcando tanto el diseño de interfaces como la lógica del servidor y la gestión de datos
- ♦ Consolidar el manejo de estructuras *front - end*, incorporando buenas prácticas de programación para garantizar interfaces funcionales y eficientes
- ♦ Aplicar el lenguaje JavaScript en la construcción de aplicaciones dinámicas, integrando lógica avanzada y componentes reutilizables
- ♦ Implementar técnicas de maquetación web orientadas a la adaptabilidad, la experiencia del usuario y la coherencia visual del proyecto
- ♦ Integrar librerías como ReactJS para optimizar el rendimiento y la escalabilidad de las aplicaciones desarrolladas
- ♦ Emplear *frameworks* como angular para estructurar aplicaciones complejas, manteniendo un flujo de trabajo organizado y sostenible
- ♦ Utilizar el entorno NodeJS en la programación del lado del servidor, garantizando una comunicación eficiente entre cliente y base de datos
- ♦ Gestionar bases de datos y procesos de integración continua, asegurando la automatización del despliegue y la calidad del desarrollo *Full Stack*





Objetivos específicos

Módulo 1. Desarrollo *Full Stack Developer*

- ♦ Emplear algoritmos y estructuras de datos para resolver problemas en entornos de desarrollo *Full Stack*
- ♦ Reconocer los distintos roles dentro de un equipo de programación y su impacto en el desarrollo de proyectos
- ♦ Gestionar versiones de código mediante Git y servicios de alojamiento colaborativo
- ♦ Implementar metodologías ágiles para optimizar la organización y ejecución de tareas en proyectos tecnológicos

Módulo 2. *Front-end* en Programación

- ♦ Estructurar contenidos web utilizando elementos fundamentales del lenguaje HTML y principios del diseño semántico
- ♦ Aplicar hojas de estilo en cascada (CSS) para definir la apariencia visual de sitios web, incluyendo animaciones y transiciones
- ♦ Implementar modelos de caja y técnicas de posicionamiento para construir interfaces ordenadas y funcionales
- ♦ Utilizar principios de diseño adaptable y Media Queries para garantizar la correcta visualización en distintos dispositivos
- ♦ Diferenciar y aplicar sistemas de maquetación modernos como Flexbox y Grid, optimizando la organización del contenido
- ♦ Integrar preprocesadores como Sass y *frameworks* como Bootstrap para agilizar el desarrollo y mantener coherencia visual

Módulo 3. Lenguaje JAVASCRIPT aplicado a *Full Stack Developer*

- ♦ Dominar los tipos primitivos de JavaScript y sus operadores para establecer una base sólida en el tratamiento de datos y estructuras lógicas.
- ♦ Diseñar funciones avanzadas que incluyan *closures*, recursividad y gestión de ámbitos, fortaleciendo el control sobre el flujo de ejecución
- ♦ Manipular estructuras complejas como objetos, *arrays*, *sets* y *maps*, aplicando métodos modernos de acceso, transformación e inmutabilidad
- ♦ Gestionar la programación asíncrona mediante *callbacks*, promesas y funciones *async/await*, optimizando la respuesta y eficiencia en entornos dinámicos

Módulo 4. Maquetación web aplicado a *Full Stack Developer*

- ♦ Implementar diseños responsivos mediante CSS Grid, Flexbox y Media Queries, adaptando interfaces web a múltiples dispositivos y resoluciones
- ♦ Construir estructuras visuales organizadas aplicando principios de maquetación, diseño fluido y reglas del modelo de caja
- ♦ Utilizar preprocesadores como Sass para optimizar el código CSS mediante variables, mixins, funciones y modularización de proyectos
- ♦ Integrar Bootstrap en proyectos web, personalizando su *theming* con Sass para desarrollar componentes reutilizables y sistemas visuales coherentes

Módulo 5. Herramientas de Javascript. Librería ReactJS

- ♦ Construir interfaces dinámicas mediante componentes reutilizables, gestionando su estado y ciclo de vida con *Hooks* integrados y personalizados
- ♦ Implementar navegación fluida en aplicaciones de una sola página utilizando React Router y sus funcionalidades para enrutado dinámico
- ♦ Gestionar flujos de datos compartidos entre componentes a través del uso eficaz del Context API y sus métodos asociados
- ♦ Preparar, probar y desplegar proyectos desarrollados con ReactJS utilizando herramientas como Jest y plataformas como Heroku

Módulo 6. *Framework* de JavaScript. Angular

- ♦ Diseñar aplicaciones modulares y escalables en angular, comprendiendo su arquitectura, CLI y principios de organización de proyectos
- ♦ Implementar interacción avanzada entre componentes, directivas y plantillas utilizando técnicas como *data binding*, proyección de contenido y pipes personalizados
- ♦ Gestionar servicios, dependencias e integración con formularios y peticiones HTTP mediante inyección de dependencias y uso de interceptores
- ♦ Incorporar navegación dinámica, seguridad, *Lazy Loading* y animaciones, optimizando la experiencia del usuario en aplicaciones de una sola página

Módulo 7. Programación en lenguaje NodeJS

- ♦ Comprender la arquitectura de NodeJS y su funcionamiento mediante módulos, NPM y el loop de eventos
- ♦ Crear servidores con Express y Socket.io, incorporando manejo de peticiones HTTP y comunicación en tiempo real
- ♦ Conectar aplicaciones a bases de datos con Mongoose y Sequelize, gestionando datos de forma eficiente
- ♦ Diseñar APIs RESTful y desplegarlas con NestJS y Docker, aplicando principios de arquitectura moderna

Módulo 8. Base de datos para desarrolladores *Full Stack*

- ♦ Distinguir los distintos tipos de bases de datos y seleccionar la opción más adecuada según las necesidades del desarrollo
- ♦ Implementar modelos relacionales y conectar aplicaciones a sistemas como MySQL, Oracle y SQL Server
- ♦ Desarrollar proyectos con bases de datos NoSQL, de grafos y clave - valor, integrando soluciones como MongoDB, Neo4j y Redis
- ♦ Incorporar bases de datos en entornos *cloud native*, microservicios y flujos de CI/CD con un enfoque avanzado

Módulo 9. UX CX. Experiencia de cliente

- ♦ Comprender la evolución del *Customer Experience* y su impacto en los nuevos hábitos de consumo
- ♦ Integrar tecnologías innovadoras para mejorar la interacción con el usuario y personalizar la experiencia
- ♦ Analizar el comportamiento del usuario a través de herramientas de analítica avanzada y pruebas de usabilidad
- ♦ Diseñar experiencias centradas en el usuario aplicando metodologías de UX, UI y comunicación visual efectiva

Módulo 10. Integración continua y despliegue de aplicaciones

- ♦ Distinguir las fases de CI/CD y su integración en los nuevos modelos de desarrollo orientados a DevOps
- ♦ Utilizar herramientas como Docker, Kubernetes y Jenkins para automatizar procesos de despliegue y orquestación
- ♦ Aplicar buenas prácticas de infraestructura como código mediante GitOps y validaciones automáticas
- ♦ Incorporar enfoques avanzados como DevSecOps, Chaos Engineering y AI en entornos de integración continua

05

Salidas profesionales

Esta titulación universitaria le brindará la experiencia académica necesaria para acceder a cargos como arquitecto de *software*, responsable de automatización o especialista en infraestructura *cloud*. A través de sus competencias, será posible integrarse en centros de desarrollo tecnológico, compañías de soluciones digitales o proyectos en empresas globales. Además, facilitará la adaptación a entornos donde la eficiencia operativa, la integración continua y el despliegue automatizado representan ventajas competitivas. Así, se abrirán oportunidades concretas para liderar iniciativas estratégicas, optimizar procesos técnicos y participar activamente en el diseño de sistemas escalables y seguros.



“

*¿Buscas desempeñarte como
Desarrollador de Aplicaciones Web
Reponsivas? Lógralo con esta titulación
universitaria en tan solo 12 meses”*

Perfil del egresado

El egresado de este programa se caracterizará por una alta capacidad para desarrollar aplicaciones web completas, tanto en el lado cliente como en el servidor. A lo largo del programa universitario, incorporará técnicas modernas de maquetación, gestión de APIs y manipulación de bases de datos. A su vez, podrá implementar soluciones interactivas, optimizadas y adaptables a múltiples dispositivos. Por otro lado, dominará herramientas de control de versiones, pruebas automatizadas y despliegue continuo. Finalmente, estará preparado para integrarse en proyectos reales, liderar desarrollos escalables y adaptarse con rapidez a nuevas tecnologías del entorno digital.

Asumirás roles clave en la manipulación de bases de datos, optimizando consultas y estructuras con precisión.

- ♦ **Pensamiento lógico:** Capacidad para estructurar soluciones eficientes y coherentes frente a problemas complejos, facilitando el desarrollo de código limpio, funcional y escalable
- ♦ **Comunicación efectiva:** Aptitud que favorece la colaboración con equipos multidisciplinarios, permitiendo transmitir ideas técnicas de forma clara, tanto en entornos presenciales como remotos
- ♦ **Adaptabilidad tecnológica:** Habilidad para facilitar la incorporación rápida de nuevas herramientas, *frameworks* y entornos digitales en constante evolución dentro del ecosistema del desarrollo web
- ♦ **Gestión del tiempo:** Competencia para organizar tareas y cumplir plazos en proyectos ágiles, optimizando recursos y mejorando la productividad en entornos exigentes





Después de realizar el programa universitario, podrás desempeñar tus conocimientos y habilidades en los siguientes cargos:

- 1. Desarrollador Full Stack:** Responsable de diseñar y construir aplicaciones completas, desde la interfaz visual hasta la lógica del servidor y la gestión de bases de datos.
- 2. Front End Developer:** Dedicado a crear interfaces interactivas, accesibles y adaptables, optimizando la experiencia del usuario en navegadores y dispositivos.
- 3. Back End Developer:** Encargado de la lógica del servidor, gestión de bases de datos y desarrollo de APIs, asegurando el funcionamiento correcto del sistema.
- 4. Ingeniero de Software Web:** Responsable de desarrollar y mantener aplicaciones web complejas con enfoque en escalabilidad, seguridad y rendimiento.
- 5. Desarrollador de Aplicaciones Web:** Dedicado a la creación de plataformas digitales funcionales a través de tecnologías modernas y *frameworks* actualizados como React, Angular o Vue.
- 6. Arquitecto de Software:** Encargado de definir la estructura técnica de los proyectos, seleccionando herramientas, lenguajes y patrones que aseguren calidad y sostenibilidad.
- 7. Especialista en DevSecOps:** Responsable de implementar prácticas de seguridad en todo el ciclo de desarrollo, integrando protección desde la escritura del código hasta el despliegue.
- 8. Administrador de Bases de Datos:** Gestiona la integridad de los datos, facilitando su disponibilidad y recuperación ante fallos.
- 9. Diseñador de Interfaces:** Responsable de diseñar componentes visuales intuitivos, alineados con la usabilidad y la identidad visual del producto digital.
- 10. Consultor en Transformación Digital:** Encargado de asesorar a empresas en la adopción de soluciones digitales, mejorando procesos y servicios mediante herramientas tecnológicas innovadoras.

06

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

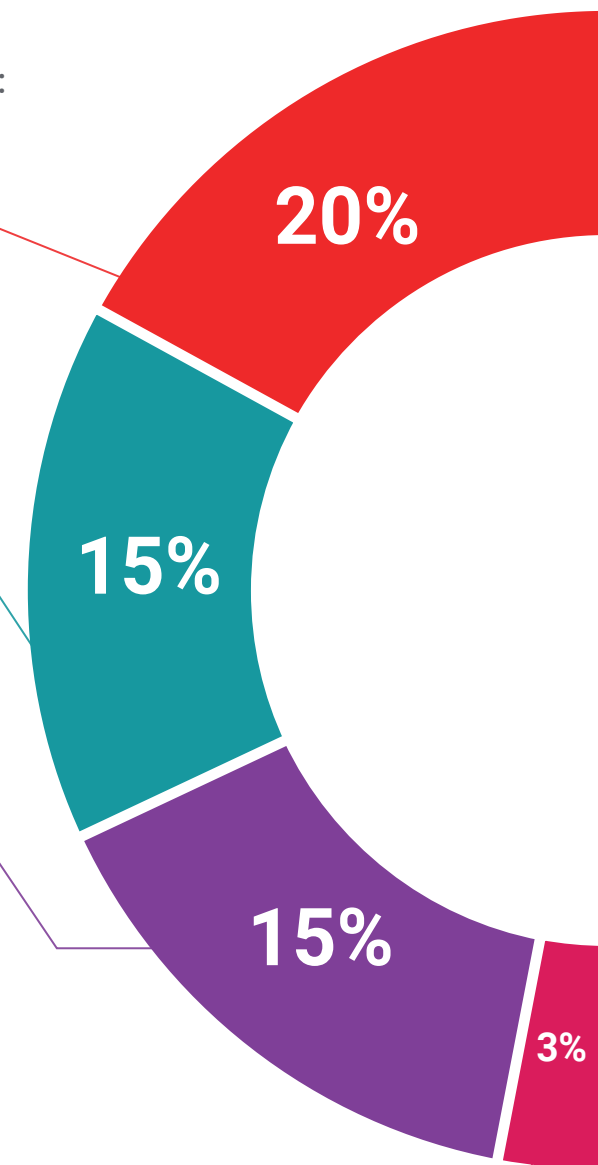
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

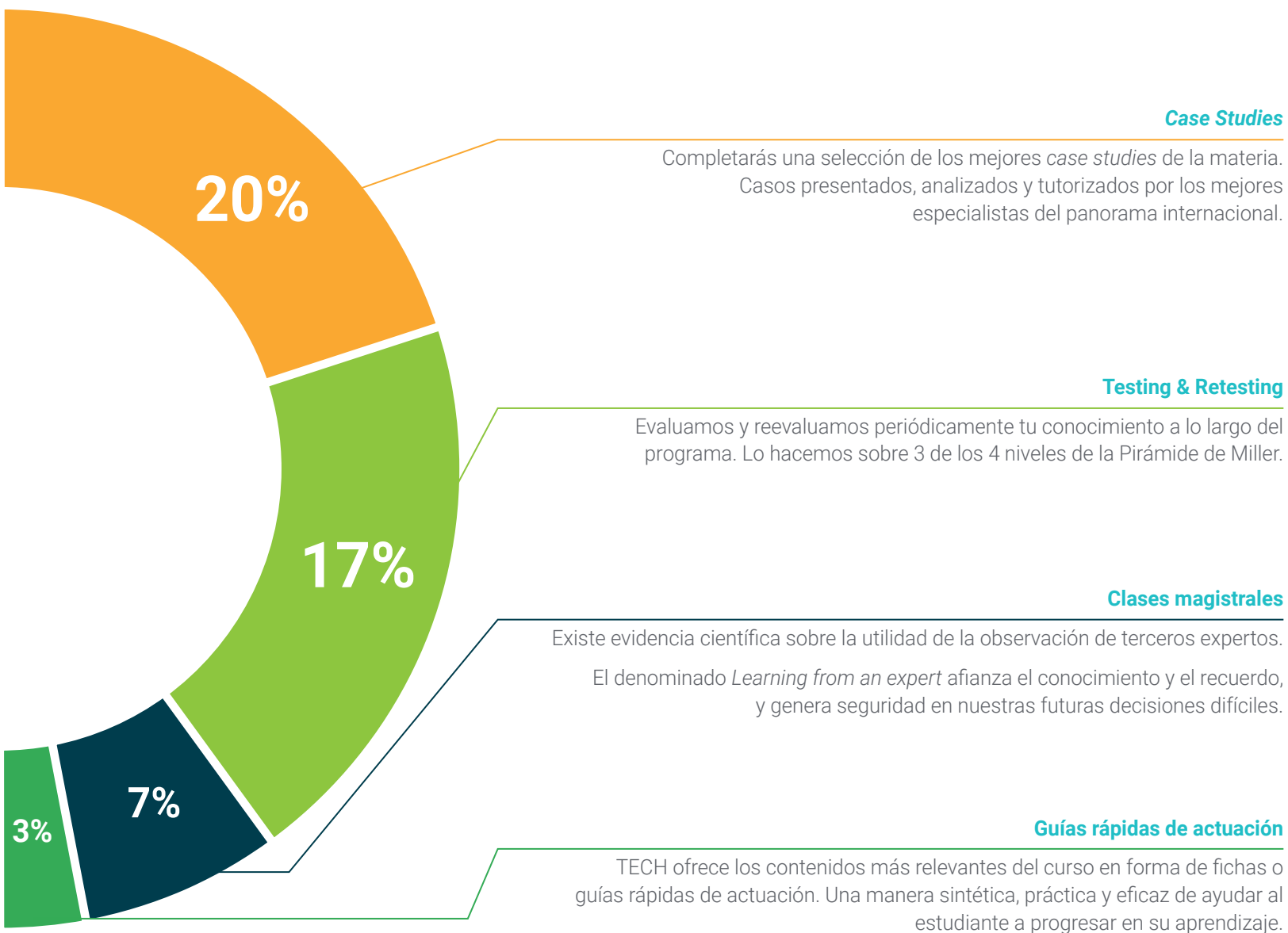
Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Lecturas complementarias

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





07

Cuadro docente

TECH aplica un riguroso proceso en la selección del cuerpo docente responsable de sus titulaciones. Gracias a este exigente proceso, se conforman equipos académicos de alto nivel, comprometidos con la excelencia y con una visión orientada a la evolución profesional del informático. Como resultado, se garantizará no solo la solidez y actualidad de los contenidos, sino también una experiencia académica respaldada por materiales didácticos de calidad, recursos complementarios y un enfoque personalizado. Así, el egresado encuentra en TECH Universidad FUNDEPOS el respaldo de la mayor institución online del mundo, comprometida con su crecimiento y proyección laboral.



“

El equipo docente, especializado en Programación Front End Full Stack Developer, ha diseñado horas de contenido adicional para que amplíes cada apartado del temario de manera personalizada”

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. 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*

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. 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. 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. 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. 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

08 Titulación

El Máster Título Propio en Programación Front End Full Stack Developer garantiza, además de la capacitación más rigurosa y actualizada, el acceso a dos diplomas de Máster Propio, uno expedido por TECH Global University y otro expedido por Universidad FUNDEPOS.



“

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

El programa del **Máster Título Propio en Programación Front End Full Stack Developer** es el más completo del panorama académico actual. A su egreso, el estudiante recibirá un diploma universitario emitido por TECH Global University, y otro por Universidad FUNDEPOS.

Estos títulos de formación permanente y actualización profesional de TECH Global University y Universidad FUNDEPOS garantizan la adquisición de competencias en el área de conocimiento, otorgando un alto valor curricular al estudiante que supere las evaluaciones y acredite el programa tras cursarlo en su totalidad.

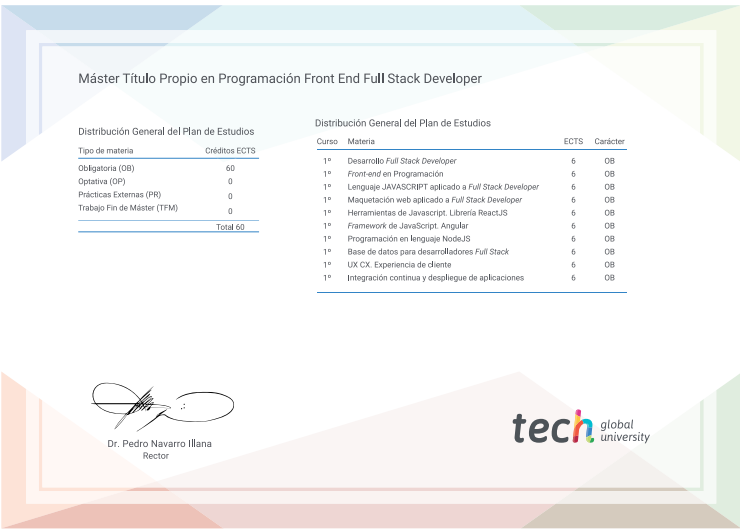
Este doble reconocimiento, de dos destacadas instituciones universitarias, suponen una doble recompensa a una formación integral y de calidad, asegurando que el estudiante obtenga una certificación reconocida tanto a nivel nacional como internacional. Este mérito académico le posicionará como un profesional altamente capacitado y preparado para enfrentar los retos y demandas en su área profesional.

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

Modalidad: **online**

Duración: **12 meses**

Acreditación: **60 ECTS**



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



Máster Título Propio Programación Front End Full Stack Developer

- » Modalidad: online
- » Duración: 12 meses
- » Titulación: TECH Universidad FUNDEPOS
- » Acreditación: 60 ECTS
- » Horario: a tu ritmo
- » Exámenes: online

Máster Título Propio

Programación Front End Full Stack Developer