

Máster Título Propio

Programación desde Cero



Máster Título Propio Programación desde Cero

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

Acceso web: www.techtitute.com/informatica/master/master-programacion-desde-cero

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 se ha convertido en una habilidad esencial en el siglo XXI, desempeñando un papel clave en la transformación digital de las sociedades y las economías. Según un informe de la OCDE, más del 60% de los trabajos actuales requieren competencias digitales avanzadas, dentro de las cuales la programación es una de las más demandadas. Teniendo en cuenta que es un campo esencial en la era digital, TECH ha ideado este programa que abordará los fundamentos de dicha disciplina y su aplicación en lenguajes de uso global. A través de una metodología 100% online, los especialistas descubrirán cómo estructurar algoritmos, gestionar datos y desarrollar proyectos funcionales, mientras construyen una base sólida para seguir avanzando en el mundo tecnológico.



“

Saber programar nunca fue tan accesible. Con el Máster Título Propio de TECH dominarás lenguajes como Python y JavaScript desde cualquier lugar, gracias a la metodología 100% online. ¡Inscríbete ya!”

La Programación es el núcleo de la transformación digital que experimentan las sociedades modernas. Desde el desarrollo de aplicaciones móviles, hasta la automatización de procesos empresariales, saber programar es una habilidad fundamental para abrirse camino en un entorno laboral en constante evolución. En este sentido, los profesionales que dominen este campo no solo encontrarán mejores oportunidades de crecimiento, sino que también potenciarán la capacidad de adaptarse y prosperar en un mundo donde la tecnología evoluciona constantemente.

Conscientes de esta necesidad, TECH ha diseñado el Máster Título Propio en Programación desde Cero, un programa completo que brindará los conocimientos más actualizados relacionados con este campo. El plan de estudios, diseñado por expertos pertenecientes al sector, abarcará desde los fundamentos básicos de algoritmos y estructuras de datos, hasta el uso práctico de lenguajes como *Python*, *JavaScript* y *HTML*. Asimismo, incluirá módulos especializados en diseño web, desarrollo de aplicaciones y gestión de bases de datos. De este modo, los especialistas adquirirán conocimientos técnicos y desarrollarán habilidades analíticas y creativas para resolver problemas mediante la programación. Todo ello, con un enfoque práctico que garantizará la implementación de los conceptos aprendidos.

Al especializarse en este ámbito, los egresados no solo encontrarán nuevas oportunidades en el sector tecnológico, sino que también destacarán en industrias como la financiera, sanitaria o logística, donde la Programación desempeña un papel clave. Además, se capacitarán para liderar proyectos de desarrollo, automatizar procesos y optimizar recursos, incrementando su perfil laboral y facilitando su acceso a salidas altamente competitivas.

Finalmente, la modalidad 100% online de este programa ofrecerá la flexibilidad necesaria para compaginar el aprendizaje con otras responsabilidades personales o profesionales. TECH pondrá a disposición del alumnado una innovadora plataforma accesible las 24 horas del día. Además, su metodología *Relearning* optimizará la capacitación mediante la reiteración de conceptos clave en diferentes contextos, lo que facilita una asimilación progresiva y eficiente del contenido.

Este **Máster Título Propio en Programación desde Cero** 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
- ♦ 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, 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



Desde algoritmos básicos, hasta el desarrollo de aplicaciones, este programa te llevará de cero a experto en poco tiempo. ¡Te convertirás en el profesional que la industria tecnológica necesita!”

“

Adéntrate en el mundo de la programación con un temario inigualable. Dominarás el Python, JavaScript y desarrollo web desde cero con el Máster Título Propio más completo del mercado”

Incluye en su cuadro docente a profesionales pertenecientes al ámbito de la programación, 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.

Con el respaldo de docentes expertos en tecnología, esta titulación te guiará paso a paso para que domines los lenguajes más demandados del mercado. ¡Empieza hoy y construye un mejor futuro!

Descubrirás un enfoque innovador que combina teoría y proyectos reales en este Máster Título Propio. ¡Inscríbete ya y haz que tus ideas cobren vida con 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».

Forbes
Mejor universidad
online del mundo

Plan
de estudios
más completo

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.

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.

Profesorado
TOP
Internacional



La metodología
más eficaz

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 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.

nº1
Mundial
Mayor universidad
online del mundo

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

A lo largo del programa, los profesionales abordarán los fundamentos de algoritmos, estructuras de datos y lógica de programación, para después adentrarse en lenguajes como *Python*, *JavaScript* y *HTML*. Además, profundizarán en áreas esenciales como el diseño web, el desarrollo de aplicaciones y la gestión de bases de datos, ofreciendo una visión integral sobre el mundo de la programación. Gracias a un enfoque progresivo y estructurado, este plan de estudios no solo garantizará la adquisición de conocimientos técnicos, sino también el desarrollo de competencias analíticas y creativas, indispensables en un entorno laboral competitivo.



“

Este posgrado se presenta como la opción ideal para iniciar tu trayectoria en el sector tecnológico y adquirir las herramientas necesarias para destacar en cualquier ámbito profesional relacionado con Programación”

Módulo 1. Programación y Desarrollo de Software desde Cero

- 1.1. Software y Hardware. Relación y Diferencias
 - 1.1.1. Software
 - 1.1.2. Diferencias entre Software y Hardware
 - 1.1.3. Relación entre Software, Hardware y Programación
- 1.2. Programación. Aspectos clave
 - 1.2.1. La Programación
 - 1.2.2. Objetivos y Aplicaciones
 - 1.2.3. Programas, Código Fuente, Compilación y Ejecución
 - 1.2.4. Errores: Errores de Sintaxis, de Ejecución, Lógicos
- 1.3. Programación desde Cero
 - 1.3.1. Estructura de un Programa
 - 1.3.2. Algoritmos: Estructura y Ejemplos
 - 1.3.3. Relación entre algoritmos y programas
 - 1.3.4. Resolución de problemas mediante algoritmos
- 1.4. Paradigmas y Tipos de Lenguajes de Programación
 - 1.4.1. Paradigmas de programación
 - 1.4.1.1. Paradigmas imperativo,
 - 1.4.1.2. Paradigma orientado a objetos
 - 1.4.1.3. Paradigma funcional
 - 1.4.1.4. Paradigma declarativo
 - 1.4.2. Lenguajes de bajo nivel vs alto nivel
 - 1.4.3. Lenguajes compilados vs interpretados
- 1.5. Traducción de Lenguajes de Programación
 - 1.5.1. El Compilador. Proceso de compilación
 - 1.5.2. El Intérprete: Proceso de interpretación
 - 1.5.3. Diferencias entre compilación e interpretación
- 1.6. Bits, Operaciones Binarias y Puertas Lógicas
 - 1.6.1. Bit. Representación binaria
 - 1.6.2. Operaciones básicas con bits: AND, OR, XOR, NOT
 - 1.6.3. Conversión entre binario y decimal
 - 1.6.4. Puertas lógicas: OR, AND, XOR, NOT, NOR y NAND





- 1.7. Diseño de Algoritmos desde Cero
 - 1.7.1. Diseño de un algoritmo desde Cero
 - 1.7.2. Algoritmos secuenciales, condicionales y repetitivos
 - 1.7.3. Algoritmos recursivos y su comparación con algoritmos iterativos
- 1.8. Elementos del Programa: Componentes y Estructura
 - 1.8.1. Entrada y salida de datos
 - 1.8.2. Variables y constantes: Uso
 - 1.8.3. Procesamiento de datos y su manipulación
 - 1.8.4. Funciones y procedimientos básicos con diagramas de flujo
- 1.9. Estructuras de Control con Diagramas de Flujo
 - 1.9.1. Estructuras de control. Función en la programación
 - 1.9.2. Estructuras condicionales: Ejemplos con diagramas de flujo
 - 1.9.3. Estructuras repetitivas: Ejemplos con diagramas de flujo
- 1.10. Ciclo de Vida y Modelos de Desarrollo de Software
 - 1.10.1. Ciclo de vida del Software. Fases
 - 1.10.2. Modelos de desarrollo: Cascada, iterativo y ágil
 - 1.10.3. Testing y mantenimiento en el desarrollo de software

Módulo 2. Desarrollo Backend I: Python desde Cero

- 2.1. Python desde Cero. Instalación
 - 2.1.1. Lenguaje Python. Características
 - 2.1.2. Instalación de Python en Windows, macOS y Linux
 - 2.1.3. Configuración del entorno de desarrollo: IDEs y editores de código
 - 2.1.4. Primer programa en Python: "Hola Mundo"
- 2.2. Sintaxis y Variables en Python
 - 2.2.1. Estructura del código en Python: Identación
 - 2.2.2. Comentarios en Python
 - 2.2.3. Variables y tipos de datos en Python
 - 2.2.4. Operaciones aritméticas y lógicas en Python
- 2.3. Control de Flujo: Condicionales
 - 2.3.1. Estructuras de control
 - 2.3.2. Condicionales: if, elif, else
 - 2.3.3. Condicional ternario

- 2.4. Bucles en Python
 - 2.4.1. Uso de Bucles en Programación
 - 2.4.2. Bucles “for” y “while”
 - 2.4.3. Control de flujo en bucles: break y continue
 - 2.4.4. Bucles anidados
- 2.5. Funciones en Python
 - 2.5.1. Función en Python. Utilidad
 - 2.5.2. Parámetros y argumentos de las Funciones
 - 2.5.3. Valores de retorno
 - 2.5.4. Funciones predefinidas vs. funciones creadas por el usuario
- 2.6. Listas y Tuplas en Python
 - 2.6.1. Creación y uso de listas en Python
 - 2.6.2. Operaciones comunes con listas: Añadir, eliminar, modificar
 - 2.6.3. Tuplas: Diferencias con listas
 - 2.6.4. Iteración sobre listas y tuplas
- 2.7. Diccionarios y Sets en Python
 - 2.7.1. Diccionarios: Clave-valor
 - 2.7.2. Métodos para manipular diccionarios
 - 2.7.3. Sets: Uso
 - 2.7.4. Comparación de diccionarios y sets.
- 2.8. Manejo de Archivos en Python desde Cero
 - 2.8.1. Apertura y cierre de archivos
 - 2.8.2. Modos de apertura: Lectura, escritura y añadido
 - 2.8.3. Lectura y escritura de archivos de texto
- 2.9. Manejo de Errores y Excepciones
 - 2.9.1. Tipos de Excepciones
 - 2.9.2. Uso de Try, Except para gestionar errores
 - 2.9.3. Creación de excepciones personalizadas
- 2.10. Buenas Prácticas y Depuración en Python
 - 2.10.1. Debugging: Propósito.
 - 2.10.2. Técnicas de depuración: Uso de print y breakpoints.
 - 2.10.3. Buenas Prácticas en la Escritura de Código

Módulo 3. Desarrollo Backend II - Algoritmos y Estructuras de Datos con Python desde Cero

- 3.1. Algoritmos de Búsqueda en Estructuras de Datos
 - 3.1.1. Finalidad de los Algoritmos de búsqueda en Estructuras de Datos
 - 3.1.2. Búsqueda lineal: Implementación y casos de uso
 - 3.1.3. Búsqueda binaria: Ejemplos
 - 3.1.4. Comparación de eficiencia: Búsqueda lineal vs binaria
- 3.2. Algoritmos de Ordenación en Estructuras de Datos (I). Técnicas básicas de Ordenación Bubble Sort e Insertion Sort
 - 3.2.1. Ordenación por burbuja (Bubble Sort): Implementación y análisis
 - 3.2.2. Ordenación por inserción (Insertion Sort): Implementación y casos de uso
 - 3.2.3. Comparación entre Bubble Sort e Insertion Sort
- 3.3. Algoritmos de Ordenación en estructuras de datos (II). Técnicas avanzadas de Ordenación Selection Sort, Merge Sort y Quick Sort
 - 3.3.1. Selection Sort. Implementación y análisis
 - 3.3.2. Merge Sort: Implementación
 - 3.3.3. Quick Sort: Implementación
 - 3.3.4. Comparación de eficiencia entre algoritmos de ordenación
- 3.4. Algoritmos Recursivos de búsqueda en estructuras de datos
 - 3.4.1. Recursión. Uso
 - 3.4.2. Recursión directa e indirecta
 - 3.4.3. Algoritmos recursivos: Factorial y Fibonacci. Ejemplos
- 3.5. Complejidad de Algoritmos de búsqueda en estructuras de datos
 - 3.5.1. Complejidad algorítmica. Medición de la eficiencia
 - 3.5.2. Notación Big-O
 - 3.5.3. Análisis de complejidad en algoritmos de búsqueda y ordenación
- 3.6. Estructuras de Datos Avanzadas
 - 3.6.1. Árboles: Terminología
 - 3.6.2. Árboles binarios: Operaciones
 - 3.6.3. Árboles de búsqueda binaria (BST): Búsqueda, inserción y eliminación

- 3.7. Algoritmos de Grafos
 - 3.7.1. Grafos. Representación
 - 3.7.2. Algoritmos de grafos: DFS y BFS
 - 3.7.3. Comparación entre DFS y BFS
 - 3.8. Programación Dinámica
 - 3.8.1. Programación Dinámica. Aplicación
 - 3.8.2. Diferencias entre programación dinámica y recursión
 - 3.8.3. Optimización mediante programación dinámica
 - 3.9. Técnicas de Optimización de Algoritmos de búsqueda de datos
 - 3.9.1. Importancia de la Optimización algorítmica de búsqueda de datos
 - 3.9.2. Técnicas de optimización: Memorización (Memoization)
 - 3.9.3. *Divide and Conquer*: Divide y vencerás
 - 3.10. Otros Algoritmos en Python
 - 3.10.1. Algoritmos de Permutación y Combinación
 - 3.10.2. Algoritmos de Hashing Básico
 - 3.10.3. Algoritmos de Conteo y Generación de Subconjuntos
- Módulo 4. Desarrollo Backend III - Flask, Creación de APIs y Arquitectura Básica desde cero**
- 4.1. Flask como Framework Backend
 - 4.1.1. Framework Backend. Finalidad
 - 4.1.2. Flask. Características
 - 4.1.3. Preparación del entorno de desarrollo e instalación de Flask
 - 4.1.4. Primer proyecto con Flask: "Hola Mundo"
 - 4.2. Rutas y Peticiones HTTP en Flask
 - 4.2.1. Ruta: Funcionamiento en una aplicación web
 - 4.2.2. Métodos HTTP en Flask: GET, POST, PUT y DELETE
 - 4.2.3. Rutas en Flask con parámetros y datos
 - 4.2.4. Organización de rutas en un proyecto
 - 4.3. Controladores y Manejo de Respuestas en Flask
 - 4.3.1. Controlador: Función y responsabilidades
 - 4.3.2. Tipos de respuestas en Flask: Texto, JSON y HTML
 - 4.3.3. Creación de controladores para APIs en Flask
 - 4.3.4. Operaciones CRUD en controladores
 - 4.4. APIs RESTful
 - 4.4.1. API RESTful. Principios
 - 4.4.2. Métodos HTTP y convenciones en APIs RESTful
 - 4.4.3. Creación de una API RESTful en Flask
 - 4.4.4. Diseño de una API de tareas con operaciones CRUD
 - 4.5. Bases de Datos y Flask con SQLite
 - 4.5.1. Bases de datos en aplicaciones web
 - 4.5.2. Conexión a SQLite en proyectos Flask
 - 4.5.3. Creación de tablas y modelos usando SQLAlchemy
 - 4.5.4. Consultas CRUD en SQLite para manejo de datos
 - 4.6. Autenticación y Seguridad Básica en APIs
 - 4.6.1. Autenticación y autorización en APIs
 - 4.6.2. Creación de un sistema de Autenticación de Usuario
 - 4.6.3. Uso de tokens para autenticación en Flask
 - 4.6.4. Protección de rutas y datos de usuario en APIs
 - 4.7. Validación de Datos y Manejo de Errores
 - 4.7.1. Manejo de errores y excepciones en Flask
 - 4.7.2. Validación de datos en peticiones a la API
 - 4.7.3. Creación de mensajes de error personalizados
 - 4.7.4. Estrategias de validación y manejo de errores en CRUD
 - 4.8. Estructuración de APIs Escalables
 - 4.8.1. Organización y estructura de un proyecto Flask escalable
 - 4.8.2. Modularización y separación de responsabilidades en APIs
 - 4.8.3. Optimización básica de APIs para rendimiento y escalabilidad
 - 4.8.4. Estrategia de organización para proyectos grandes

- 4.9. Comunicación en Tiempo Real con WebSockets
 - 4.9.1. WebSockets. Aplicaciones
 - 4.9.2. Implementación de WebSockets en Flask con Flask-SocketIO
 - 4.9.3. Comunicación en tiempo real en aplicaciones Flask
- 4.10. Deploy y Mantenimiento de Aplicaciones
 - 4.10.1. Preparación de aplicaciones Flask para producción
 - 4.10.2. Deploy en plataformas populares como Heroku y Render
 - 4.10.3. Uso de Docker para despliegue en contenedores
 - 4.10.4. Monitoreo y mantenimiento de aplicaciones backend

Módulo 5. Programación Orientada a Objetos y Patrones de Diseño desde Cero

- 5.1. Programación Orientada a Objetos (POO) desde Cero
 - 5.1.1. Programación Orientada a Objetos
 - 5.1.2. Diferencias entre POO y programación estructurada
 - 5.1.3. Elementos POO: Clases, objetos, métodos y atributos
- 5.2. Clases y Objetos en Python
 - 5.2.1. Creación de clases y objetos en Python
 - 5.2.2. Atributos de instancia y de clase
 - 5.2.3. Métodos especiales (init, str, repr, etc.)
 - 5.2.4. Métodos estáticos y de clase: Usos
- 5.3. Encapsulamiento y Abstracción en Clases
 - 5.3.1. Encapsulamiento: Utilidad
 - 5.3.2. Modificadores de acceso en Python
 - 5.3.2.1. Públicos, protegidos y privados
 - 5.3.3. Abstracción: Ocultación de detalles y mejora de la simplicidad
 - 5.3.4. Uso de propiedades (@property) para control de acceso
- 5.4. Herencia en Python. Utilidad en POO
 - 5.4.1. Herencia: Utilidad en POO
 - 5.4.2. Creación de clases derivadas y herencia múltiple en Python
 - 5.4.3. Métodos y atributos heredados y sobrecarga en la herencia
 - 5.4.4. Jerarquías de clases y gestión de clases base
- 5.5. Polimorfismo y Sobrecarga en Python
 - 5.5.1. Polimorfismo: Duck Typing
 - 5.5.2. Polimorfismo con clases y métodos en Python
 - 5.5.3. Sobrecarga y Sobrescritura de métodos en Python
 - 5.5.4. El polimorfismo en diseño de software. Aplicaciones y Ventajas
- 5.6. Relaciones entre Clases y Diseño de Estructuras Complejas
 - 5.6.1. Tipos de Relaciones: Asociación, agregación y composición
 - 5.6.2. Diferencias entre agregación y composición: Ejemplos
 - 5.6.3. Diseño de estructuras complejas utilizando relaciones entre clases
- 5.7. Patrones de Diseño y Principios SOLID
 - 5.7.1. Relevancia de los Patrones de diseño
 - 5.7.2. Aplicación de patrones de diseño en proyectos POO. Ventajas
 - 5.7.3. Clasificación de patrones de diseño.
 - 5.7.4. Principios SOLID y su importancia en el diseño orientado a objetos
- 5.8. Patrones de Diseño Creacionales
 - 5.8.1. Finalidad de los Patrones de Diseño creacionales
 - 5.8.2. Patrón Singleton
 - 5.8.3. Patrón Factory y Factory Method
 - 5.8.4. Patrón Builder
- 5.9. Patrones de Diseño Estructurales
 - 5.9.1. Finalidad de los Patrones de Diseño Estructurales
 - 5.9.2. Patrón Adapter
 - 5.9.3. Patrón Decorator
 - 5.9.4. Patrón Facade
- 5.10. Patrones de Diseño de Comportamiento
 - 5.10.1. Patrones de comportamiento.Aplicaciones
 - 5.10.2. Patrón Observer
 - 5.10.3. Patrón Strategy

Módulo 6. Frontend I - HTML y CSS desde Cero

- 6.1. HTML desde Cero
 - 6.1.1. HTML. Propósito en Desarrollo web
 - 6.1.2. Estructura de un documento HTML: DOCTYPE, <html>, <head>, <body>
 - 6.1.3. Etiquetas semánticas y de contenido: <header>, <nav>, <section>, <footer>
 - 6.1.4. Elementos esenciales: Párrafos (<p>), listas (,), enlaces (<a>), imágenes ()
 - 6.1.5. Buenas prácticas en HTML
- 6.2. Elementos de Texto y Multimedia en HTML
 - 6.2.1. Etiquetas de texto esenciales: <p>, encabezados, listas, negrita y cursiva
 - 6.2.2. Inserción de multimedia: Atributos de , <audio>, <video>
 - 6.2.3. Atributos esenciales para accesibilidad (alt, aria-label)
- 6.3. Formularios en HTML
 - 6.3.1. Estructura y componentes de formularios: <form>, <input>, <label>, <button>
 - 6.3.2. Tipos de entradas: Texto, email, contraseña, botones de envío
 - 6.3.3. Validación HTML5: Validación de campos en el lado del cliente
 - 6.3.4. Formularios con validación básica. Ejemplos
- 6.4. CSS desde Cero
 - 6.4.1. Lenguaje CSS desde Cero: Uso y relación con HTML
 - 6.4.2. Sintaxis de CSS: Selectores, propiedades y valores
 - 6.4.3. Aplicación de estilos en línea, internos y externos
 - 6.4.4. Selectores avanzados: De tipo, clase, ID, pseudoclasas (:hover, :focus)
- 6.5. Modelo de Caja (Box Model) en CSS
 - 6.5.1. Modelo de caja: Importancia en CSS
 - 6.5.2. Propiedades clave: margin, padding, border, width, height
 - 6.5.3. Uso de box-sizing para control preciso del box model
 - 6.5.4. El Diseño aplicado el modelo de caja. Ejemplos
- 6.6. Tipografía y Estilos de Texto en CSS
 - 6.6.1. Propiedades de colores y fuentes: color, font-family, font-size
 - 6.6.2. Estilos de texto avanzados: Negrita, cursiva, alineación (text-align)
 - 6.6.3. Espaciado y separación de texto: line-height, letter-spacing
 - 6.6.4. Unidades de medida en CSS (px, em, rem) y su uso en tipografía

- 6.7. Diseño de Layouts con CSS - Flexbox
 - 6.7.1. Flexbox: Propósito
 - 6.7.2. Propiedades de Flexbox: justify-content, align-items, flex-direction
 - 6.7.3. Distribución de elementos y alineación en Flexbox
 - 6.7.4. Ejemplos de layouts con Flexbox
- 6.8. CSS Grid y Diseño Responsivo con CSS
 - 6.8.1. CSS Grid: Filas, columnas y áreas
 - 6.8.2. Media Queries: Estructura y aplicación en distintos dispositivos
 - 6.8.3. Diseño responsivo para móvil, tablet y desktop
 - 6.8.4. Ajuste de tipografía y unidades fluidas en diseño responsivo
- 6.9. Animaciones y Transiciones en CSS
 - 6.9.1. Transiciones: Propiedad transition, efectos en :hover
 - 6.9.2. Animaciones con CSS: Uso de @keyframes, animaciones básicas
 - 6.9.3. Técnicas para suavizar transiciones y animaciones en la web
- 6.10. Accesibilidad Web en Diseño
 - 6.10.1. La accesibilidad web: Importancia
 - 6.10.2. Diseño de sitios accesibles. Buenas prácticas
 - 6.10.3. Etiquetas aria y herramientas de validación de accesibilidad

Módulo 7. Frontend II – JavaScript desde Cero

- 7.1. JavaScript desde Cero
 - 7.1.1. Lenguaje JavaScript
 - 7.1.2. Integración de JavaScript en HTML
 - 7.1.3. Primer programa en JavaScript: "Hola Mundo"
- 7.2. Variables y Tipos de Datos en JavaScript
 - 7.2.1. Declaración de variables con var, let y const
 - 7.2.2. Tipos de datos: Números, cadenas, booleanos
 - 7.2.3. Conversión entre tipos de datos
- 7.3. Estructuras de Control en JavaScript
 - 7.3.1. Condicionales: if, else if, else
 - 7.3.2. Bucles: for, while, do...while
 - 7.3.3. Switch-case: Alternativa a múltiples condicionales
 - 7.3.4. Break y continue en bucles

- 7.4. Funciones en JavaScript
 - 7.4.1. Declaración de funciones
 - 7.4.2. Parámetros, valores de retorno y scope
 - 7.4.3. Funciones flecha (=>) y funciones anónimas
 - 7.4.4. Callbacks y recursividad en funciones
- 7.5. Manipulación del DOM (Document Object Model) con JavaScript
 - 7.5.1. DOM: Estructura del documento HTML
 - 7.5.2. Selección de elementos del DOM (getElementById, querySelector)
 - 7.5.3. Manipulación de elementos: Cambiar texto, estilos y atributos
 - 7.5.4. Eventos: click, input, submit y más
- 7.6. Arreglos y Objetos en JavaScript
 - 7.6.1. Declaración y uso de arreglos
 - 7.6.2. Métodos comunes de arreglos: push, pop, map, filter
 - 7.6.3. Creación y uso de objetos
 - 7.6.4. Iteración sobre arreglos y objetos
- 7.7. Promesas y Asincronía en JavaScript
 - 7.7.1. Asincronía y uso de callbacks en JavaScript
 - 7.7.2. Promise en JavaScript: Creación y manejo
 - 7.7.3. Uso de async y await en JavaScript
- 7.8. APIs y Fetch en JavaScript
 - 7.8.1. API en JavaScript: Propósito
 - 7.8.2. Consumo de APIs REST con fetch
 - 7.8.3. Manejo de errores y estados de las solicitudes
- 7.9. Almacenamiento Local en navegadores web
 - 7.9.1. Local Storage y Session Storage en servicios y aplicaciones web
 - 7.9.2. Almacenamiento y recuperación de datos en Local Storage
 - 7.9.2. IndexedDB como base de datos para el navegador
 - 7.9.3. Manejo de cookies en JavaScript
 - 7.9.4. Almacenamiento en el navegador: Ejemplos
- 7.10. Buenas Prácticas en JavaScript y Herramientas para Desarrolladores
 - 7.10.1. Código en JavaScript: Buenas prácticas
 - 7.10.2. Uso de herramientas de desarrollo del navegador en JavaScript
 - 7.10.3. Depuración y manejo de errores en JavaScript

Módulo 8. Frontend III - React.js desde cero

- 8.1. React.js desde Cero
 - 8.1.1. React JS como librería para el desarrollo de aplicaciones web
 - 8.1.2. Componentes y Virtual DOM en React JS: Arquitectura y funcionamiento
 - 8.1.3. Instalación y configuración con el frameworks NextJS
 - 8.1.4. Primer componente en React: "Hola Mundo"
- 8.2. JavaScript XML o JSX, y componentes en React
 - 8.2.1. JSX: Sintaxis y características
 - 8.2.2. Creación de componentes funcionales en React.js
 - 8.2.3. Uso de props para el paso de datos entre componentes
 - 8.2.4. Componentes funcionales frente a componentes de clase para el desarrollo en React.js
- 8.3. Estado y Eventos en React,Js
 - 8.3.1. Estado de los componentes en React
 - 8.3.2. Uso de useState para la gestión del estado
 - 8.3.3. Manejo de eventos en React.js: onClick, onChange, entre otros
 - 8.3.4. Ejemplos de Gestión de estado y eventos en React.js
- 8.4. Ciclo de Vida de Componentes y Efectos en React
 - 8.4.1. Ciclo de vida en componentes en React
 - 8.4.2. Uso de useEffect para manejar efectos en React
 - 8.4.3. Componentes con montaje, actualización y desmontaje en React
- 8.5. Enrutamiento con React Router
 - 8.5.1. SPA (Single Page Applications) y enrutamiento en aplicaciones web
 - 8.5.2. Instalación y configuración de React Router
 - 8.5.3. Creación de rutas y navegación entre páginas con React Router
- 8.6. Formularios y Validación en React
 - 8.6.1. Creación de formularios interactivos en React
 - 8.6.2. Manejo de entradas del usuario y envío de datos en React
 - 8.6.3. Validación de formularios en tiempo real en React
- 8.7. Consumo de APIs en React
 - 8.7.1. Consumo de APIs con fetch y axios en React
 - 8.7.2. Manejo de estados de carga, éxito y error en React
 - 8.7.3. Actualización de componentes según datos de API en React

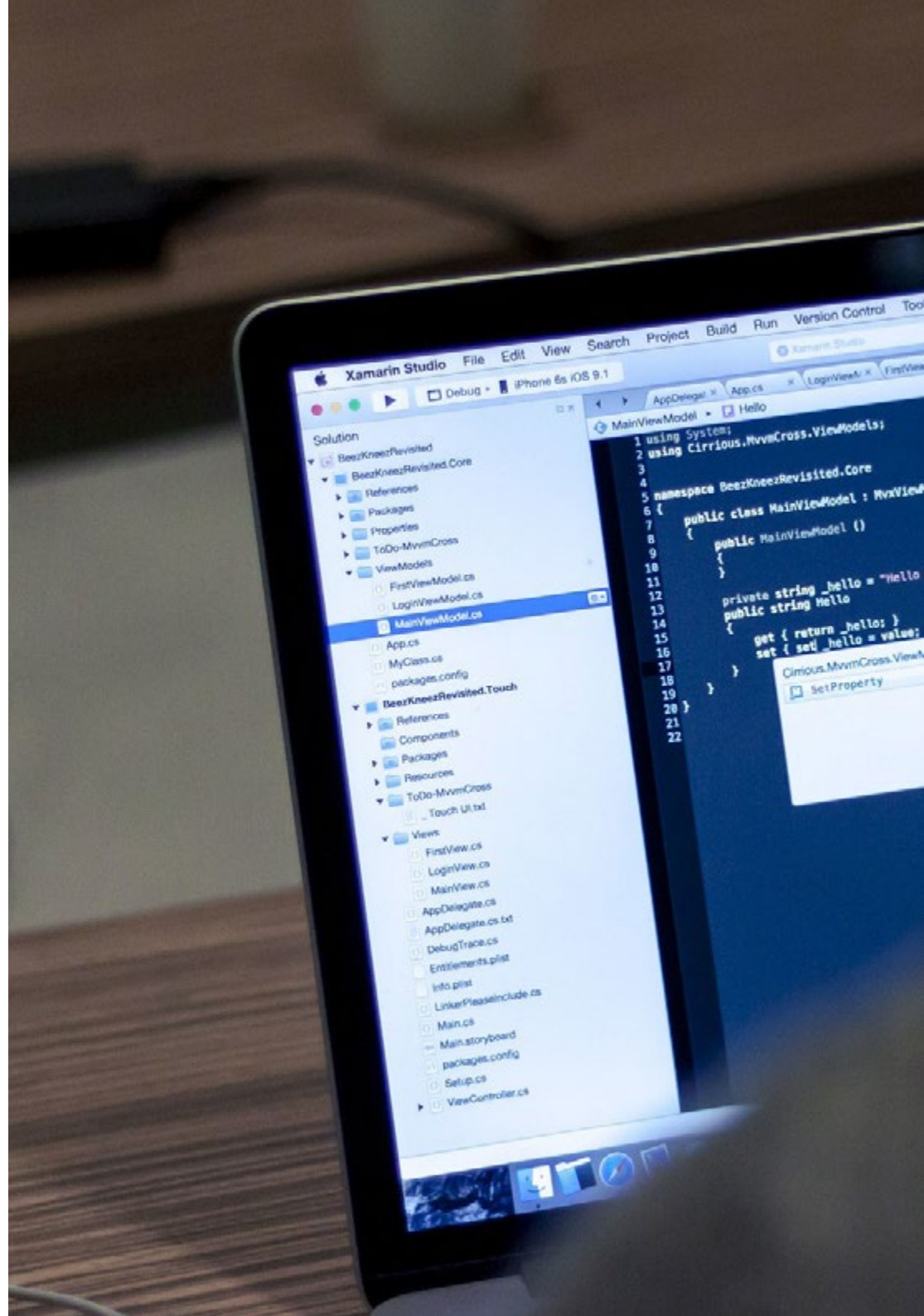
- 8.8. Componentes Reutilizables y Librerías Externas en React
 - 8.8.1. Componentes reutilizables en React
 - 8.8.2. Creación de componentes reutilizables en React
 - 8.8.3. Uso de librerías externas como Material UI y Bootstrap en React
- 8.9. Gestión Global del Estado en React
 - 8.9.1. Gestión global del estado con opciones nativas: Context API y Hooks personalizados
 - 8.9.2. Librerías externas para la gestión de datos
 - 8.9.3. Comparativa de enfoques para gestión global del estado. Ejemplos
- 8.10. Deploy y Optimización de Aplicaciones React
 - 8.10.1. Preparación de una aplicación React para producción
 - 8.10.2. Deploy en plataformas como Netlify y Vercel
 - 8.10.3. Optimización de rendimiento: Lazy Loading, memoización, server components y code splitting.
 - 8.10.4. Monitoreo y mantenimiento de aplicaciones React en producción. Herramientas y análisis de rendimiento

Módulo 9. Gestión y Optimización de Bases de Datos desde Cero

- 9.1. Base de Datos desde Cero
 - 9.1.1. Bases de datos: Tipos
 - 9.1.2. Bases de datos relacionales vs no relacionales
 - 9.1.3. Lenguajes de Programación SQL y NoSQL
- 9.2. Modelado de Datos Relacionales
 - 9.2.1. Modelo relacional de Base de Datos
 - 9.2.2. Tablas, filas y columnas en una base de datos relacional
 - 9.2.3. Claves primarias y foráneas: Relaciones entre tablas
 - 9.2.4. Normalización: 1NF, 2NF, 3NF
- 9.3. Lenguaje SQL: DML y DDL
 - 9.3.1. SQL: Lenguaje de consulta estructurada
 - 9.3.2. Consultas de creación y eliminación: CREATE, DROP
 - 9.3.3. Consultas SELECT, INSERT, UPDATE y DELETE
 - 9.3.4. Filtrado y ordenación de datos con SQL
- 9.4. Consultas Avanzadas en SQL
 - 9.4.1. Joins: INNER JOIN y OUTER JOIN
 - 9.4.2. Subconsultas y consultas anidadas
 - 9.4.3. Funciones agregadas en SQL: SUM, AVG, COUNT
- 9.5. Bases de Datos NoSQL y MongoDB
 - 9.5.1. Base de Datos NoSQL
 - 9.5.2. Comparación entre SQL y NoSQL
 - 9.5.3. MongoDB: Base de datos de documentos
 - 9.5.4. Esquemas flexibles en NoSQL
- 9.6. Optimización de Bases de Datos
 - 9.6.1. Importancia de la optimización de consultas
 - 9.6.2. Uso de índices en bases de datos relacionales
 - 9.6.3. Optimización de bases de datos NoSQL
- 9.7. Seguridad en Bases de Datos
 - 9.7.1. Seguridad en bases de datos
 - 9.7.2. Encriptación de datos sensibles
 - 9.7.3. Gestión de usuarios y permisos en bases de datos
 - 9.7.4. Estrategias de protección de bases de datos contra ataques
- 9.8. Escalabilidad de Bases de Datos
 - 9.8.1. Escalabilidad en bases de datos
 - 9.8.2. Particionamiento horizontal y vertical
 - 9.8.3. Replicación y clustering en bases de datos
- 9.9. Backup y Recuperación de Datos
 - 9.9.1. Importancia del backup en bases de datos
 - 9.9.2. Técnicas de backup automáticas y manuales
 - 9.9.3. Recuperación de datos en bases de datos relacionales y NoSQL
- 9.10. Implementación de Bases de Datos en Proyectos
 - 9.10.1. Diseño de bases de datos para un proyecto real
 - 9.10.2. Integración de bases de datos con aplicaciones backend

Módulo 10. Herramientas de Desarrollo desde cero: Linux, Control de Versiones, CI/CD, Docker y Metodologías Ágiles

- 10.1. Linux desde Cero
 - 10.1.1. Linux
 - 10.1.2. Diferencias entre Linux y otros sistemas operativos
 - 10.1.3. Distribuciones populares de Linux para Desarrolladores
 - 10.1.4. Configuración y personalización del entorno de desarrollo
 - 10.1.5. Editores de Texto en Linux
- 10.2. Uso de la Terminal en Linux desde Cero
 - 10.2.1. La Terminal. Usos y Funcionalidades
 - 10.2.2. Comandos de navegación y gestión de archivos en la terminal
 - 10.2.3. Permisos de archivos y directorios en Linux
 - 10.2.4. Redirección de comandos y uso de tuberías (pipes) para optimizar tareas
- 10.3. Control de Versiones con Git desde Cero
 - 10.3.1. Git: Proveedores CLOUD
 - 10.3.2. Creación y Gestión de Repositorios
 - 10.3.3. Flujo de trabajo: git init, git add, git commit y git status
 - 10.3.4. Trabajo con ramas: Creación, Fusión y Resolución de conflictos
- 10.4. Colaboración en Equipos con GitHub desde cero
 - 10.4.1. GitHub: Repositorios remoto
 - 10.4.2. Conexión de un repositorio local a GitHub: git remote. Configuración inicial
 - 10.4.3. Sincronización con repositorios remotos
 - 10.4.4. Pull requests y revisión de código colaborativa
- 10.5. CI/CD (I) - Integración Continua (CI) con GitHub Actions desde cero
 - 10.5.1. La Integración continua (CI)
 - 10.5.2. Configuración de workflows en GitHub Actions
 - 10.5.3. Automatización de pruebas y despliegues
- 10.6. Docker desde Cero
 - 10.6.1. Docker y Contenedores
 - 10.6.2. Instalación y configuración de Docker
 - 10.6.3. Creación y gestión de contenedores Docker
 - 10.6.4. Dockerfiles: Creación de imágenes personalizadas



- 10.7. CI/CD (II) - Entrega Continua (CD) con Docker y GitHub Actions desde cero
 - 10.7.1. Entrega continua (CD)
 - 10.7.2. Configuración de pipelines de CD con Docker y GitHub Actions
 - 10.7.3. Despliegue automatizado con Docker Compose
- 10.8. Metodologías Ágiles desde Cero (I). Principios y Valores
 - 10.8.1. Metodologías ágiles: Principios
 - 10.8.2. El Manifiesto Ágil: Valores y principios fundamentales
 - 10.8.3. Comparación con metodologías tradicionales: Cascada vs. Ágil
- 10.9. Metodologías Ágiles (II): Scrum desde Cero
 - 10.9.1. Scrum y su Aplicabilidad
 - 10.9.2. Roles clave en Scrum: Product Owner, Scrum Master y Development Team
 - 10.9.3. Artefactos en Scrum: Product Backlog, Sprint Backlog e Incremento
 - 10.9.4. Eventos Scrum: Sprint Planning, Daily Scrum, Sprint Review y Retrospective
- 10.10. Metodologías Ágiles (III): Kanban y Métricas desde Cero
 - 10.10.1. Kanban y su enfoque visual
 - 10.10.2. Elementos clave en Kanban: Columnas, tarjetas y límites WIP
 - 10.10.3. Métricas ágiles: Burnup, burndown charts, velocity y lead tim

“

Te capacitarás desde la comodidad de tu hogar y a tu ritmo, con la innovadora metodología online y la flexibilidad que necesitas. ¡No esperes más para impulsar tu carrera en tecnología! Matricúlate hoy mismo”

04

Objetivos docentes

Este Máster Título Propio tiene como meta principal dotar a los profesionales con las competencias necesarias para iniciar su carrera en el mundo de la programación de manera sólida y efectiva. A lo largo del programa, desarrollarán una comprensión profunda sobre los conceptos fundamentales de la programación, tales como algoritmos, estructuras de datos y lógica de programación, que constituyen la base sobre la cual construirán habilidades más avanzadas. Además, adquirirán un dominio de lenguajes de programación ampliamente utilizados en la industria (*Python*, *JavaScript* y *HTML*), lo que les permitirá enfrentarse con confianza a diversos proyectos tecnológicos.

“

Adquirirás las habilidades necesarias para liderar proyectos de desarrollo, gestionar bases de datos y optimizar procesos, incrementando tu competitividad profesional en el sector tecnológico”



Objetivos generales

- ♦ Manejar los fundamentos de la programación y el desarrollo de software mediante la identificación de los elementos esenciales de un programa, las estructuras de control y la relación entre software y hardware
- ♦ Comprender los fundamentos de Python y su sintaxis básica, desarrollando un entorno adecuado para el desarrollo backend
- ♦ Adquirir una sólida comprensión sobre los algoritmos y estructuras de datos fundamentales
- ♦ Dominar los fundamentos del desarrollo backend y su rol dentro de una arquitectura de software
- ♦ Gestionar los principios de la programación orientada a objetos y aplicarlos en la construcción de software flexible, reutilizable y mantenible, utilizando Python como lenguaje de programación
- ♦ Desarrollar los conceptos fundamentales de HTML y CSS para estructurar y dar estilo a páginas web
- ♦ Analizar el lenguaje JavaScript, desde los fundamentos, hasta las técnicas avanzadas
- ♦ Introducir los conceptos fundamentales de React.js para construir aplicaciones web interactivas
- ♦ Administrar los tipos de bases de datos y sus lenguajes de consulta, tanto en entornos relacionales como NoSQL
- ♦ Detallar la automatización de pruebas, integración y despliegue de códigos





Objetivos específicos

Módulo 1. Programación y Desarrollo de Software desde Cero

- ♦ Definir y diferenciar entre software y hardware
- ♦ Fundamentar los conceptos esenciales de la programación
- ♦ Conocer la estructura básica de un programa
- ♦ Explorar y analizar los diferentes paradigmas de programación

Módulo 2. Desarrollo Backend I: Python desde Cero

- ♦ Dominar las características de Python
- ♦ Comprender la estructura y sintaxis básica de Python
- ♦ Desarrollar habilidades en el control de flujo mediante condicionales
- ♦ Aplicar bucles para crear ciclos de repetición en Python

Módulo 3. Desarrollo Backend II - Algoritmos y Estructuras de Datos con Python desde Cero

- ♦ Implementar y comparar tipos de algoritmos de búsqueda en estructuras de datos
- ♦ Analizar algoritmos de ordenación como burbuja, inserción, selección, merge sort y quick sort
- ♦ Examinar la complejidad algorítmica y medición de eficiencia utilizando la notación Big O
- ♦ Representar grafos y realizar recorridos en profundidad (DFS) y en anchura (BFS)

Módulo 4. Desarrollo Backend III - Flask, Creación de APIs y Arquitectura Básica desde cero

- ♦ Desarrollar APIs RESTful utilizando Flask
- ♦ Integrar bases de datos en aplicaciones Flask
- ♦ Implementar autenticación y seguridad en APIs
- ♦ Diseñar la arquitectura básica de una aplicación backend con Flask

Módulo 5. Programación Orientada a Objetos y Patrones de Diseño desde Cero

- ♦ Definir los conceptos clave de la Programación Orientada a Objetos tales como clases, objetos, atributos, métodos, encapsulación, abstracción, herencia y polimorfismo
- ♦ Comprender el uso de encapsulamiento y abstracción en clases mediante el uso del lenguaje de programación de Python
- ♦ Examinar el concepto de polimorfismo y sobrecarga dentro del lenguaje Python comprendiendo sus aplicaciones y ventajas
- ♦ Determinar los tipos de relaciones entre clases como asociación, agregación y composición

Módulo 6. Frontend I - HTML y CSS desde Cero

- ♦ Identificar la estructura básica de un documento HTML y su importancia en el desarrollo web
- ♦ Utilizar HTML para organizar y presentar contenido en la web de forma semántica y accesible: Arquitectura web
- ♦ Aplicar estilos con CSS para mejorar la presentación visual de los elementos
- ♦ Emplear el modelo de caja de CSS para estructurar y distribuir los elementos en la interfaz



Módulo 7. Frontend II – JavaScript desde Cero

- ♦ Comprender la sintaxis y los tipos de datos en JavaScript
- ♦ Aprender a estructurar el código utilizando funciones y estructuras de control
- ♦ Manipular el DOM para interactuar con páginas web dinámicas
- ♦ Trabajar con APIs y manejar asincronía mediante Promesas y async/await

Módulo 8. Frontend III - React.js desde cero

- ♦ Comprender el uso de JSX para crear interfaces declarativas
- ♦ Aprender a trabajar con componentes funcionales, props y ciclos de vida
- ♦ Gestionar estados locales y globales utilizando herramientas modernas como Context API y Redux Toolkit
- ♦ Implementar enrutamiento para construir aplicaciones de una sola página (SPA)

Módulo 9. Gestión y Optimización de Bases de Datos desde Cero

- ♦ Reconocer los tipos de bases de datos y sus características
- ♦ Comprender y aplicar el modelo relacional de datos
- ♦ Desarrollar habilidades en SQL para la gestión de bases de datos
- ♦ Utilizar consultas avanzadas en SQL

Módulo 10. Herramientas de Desarrollo desde cero: Linux, Control de Versiones, CI/CD, Docker y Metodologías Ágiles

- ♦ Manejar el sistema operativo Linux a nivel de línea de comandos
- ♦ Dominar el uso de Git para el control de versiones
- ♦ Implementar pipelines de Integración y Despliegue Continuo (CI/CD)
- ♦ Crear y gestionar contenedores Docker



¿Quieres dominar el lenguaje del futuro? Este Máster Título Propio representa el primer paso hacia el futuro tecnológico que siempre has deseado. ¡Inscríbete ahora!

05

Salidas profesionales

Esta titulación abre una amplia gama de salidas profesionales en un sector tecnológico en constante expansión. A medida que las empresas de todos los sectores se digitalizan, la demanda de profesionales con habilidades en programación no solo aumenta, sino que se diversifica, abriendo oportunidades en áreas tan variadas como el desarrollo de software, la gestión de bases de datos, el diseño web y la automatización de procesos. Al finalizar, los egresados estarán perfectamente capacitados para asumir roles en empresas de tecnología, *startups* innovadoras y organizaciones de cualquier sector que busquen integrar soluciones digitales en sus operaciones.





“

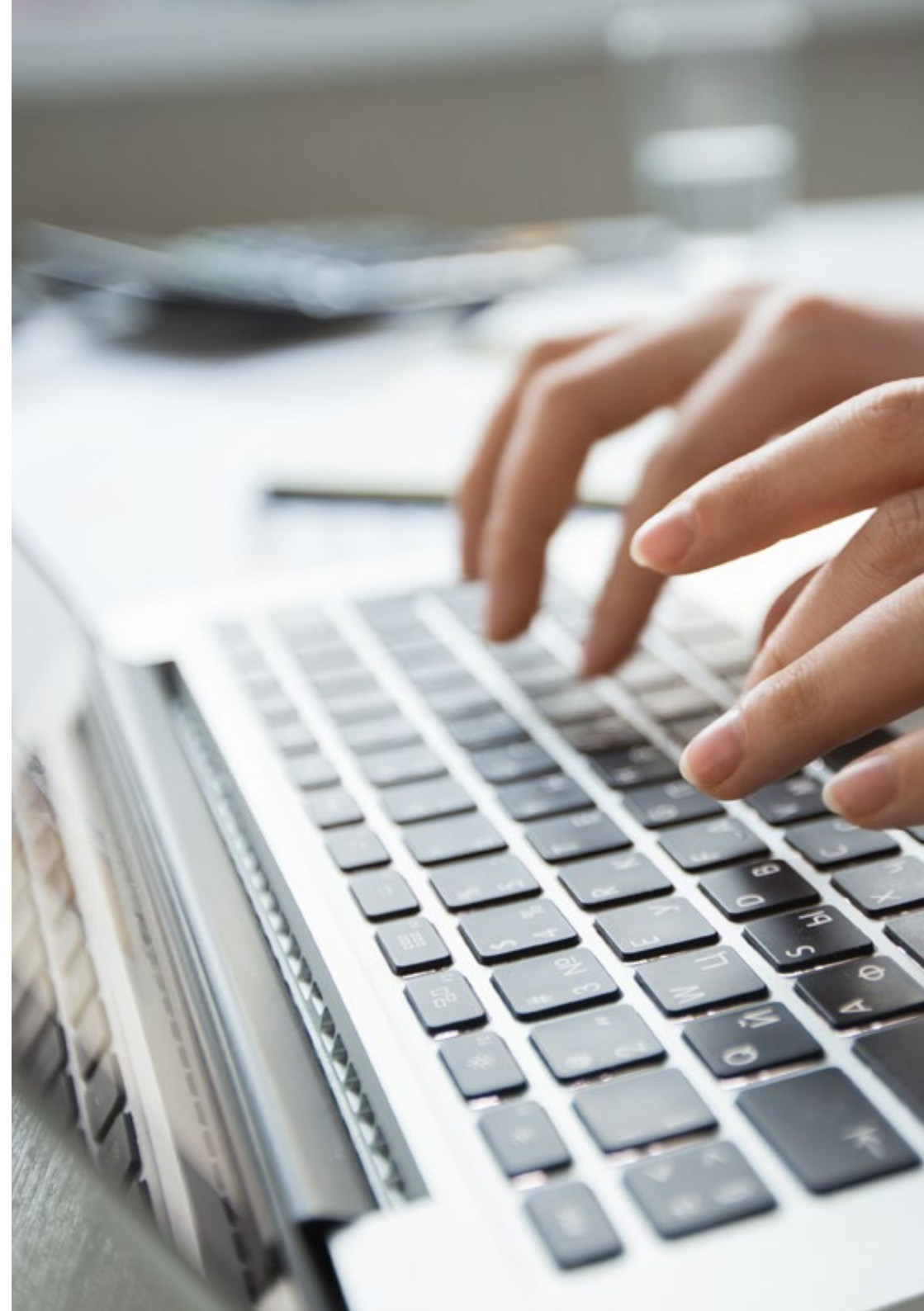
No solo te convertirás en un experto de alto nivel, sino que también abordarás una amplia variedad de oportunidades profesionales, asegurando tu éxito en un mercado laboral en continuo crecimiento y transformación”

Perfil del egresado

El egresado será un profesional altamente capacitado para afrontar los retos tecnológicos del futuro. En este sentido, no solo dominará los conceptos fundamentales de algoritmos, estructuras de datos y lenguajes de programación, sino que también habrá desarrollado una profunda comprensión sobre las herramientas y tecnologías más demandadas en la industria. Con una sólida base en programación y desarrollo de aplicaciones, este experto será capaz de abordar proyectos tecnológicos de forma autónoma y eficaz. Además, se destacará por su capacidad para resolver problemas complejos, aplicar soluciones innovadoras y gestionar sistemas y bases de datos.

Te convertirás en un profesional versátil, con una preparación técnica sólida y una visión estratégica, capaz de integrar la programación en soluciones tecnológicas que respondan a las empresas y la sociedad.

- ♦ **Pensamiento crítico y resolución de problemas:** Analizar situaciones complejas y encontrar soluciones innovadoras y eficientes mediante la programación
- ♦ **Trabajo en equipo y colaboración multidisciplinaria:** Trabajar en equipos multidisciplinares, colaborando eficazmente con profesionales de distintas áreas, como diseñadores, ingenieros y analistas de datos, para lograr objetivos comunes en proyectos tecnológicos
- ♦ **Gestión del tiempo y autonomía en el aprendizaje:** Manejar los tiempos de manera eficaz para enfrentarse a plazos ajustados y desarrollar proyectos de manera independiente y responsable
- ♦ **Adaptabilidad a nuevas tecnologías:** Gestionar nuevas herramientas y lenguajes de programación, asegurando su constante actualización en un entorno tecnológico que evoluciona a un ritmo acelerado





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

- 1. Desarrollador de Software:** Responsable de diseñar, crear y mantener aplicaciones y programas informáticos según las necesidades del cliente o empresa.
- 2. Desarrollador Web:** Gestor de la creación, diseño y mantenimiento de sitios web, asegurando su funcionalidad y optimización en diferentes dispositivos.
- 3. Ingeniero de Bases de Datos:** Encargado de diseñar, implementar y gestionar bases de datos, garantizando su eficiencia, seguridad y disponibilidad.
- 4. Analista de Sistemas:** Desarrollador de los sistemas informáticos existentes, proponiendo mejoras para optimizar el rendimiento y la eficiencia de los procesos.
- 5. Programador Front-End:** Responsable del desarrollo de la parte visual de aplicaciones y sitios web, mejorando la experiencia del usuario.
- 6. Programador Back-End:** Encargado de desarrollar la lógica de servidor y bases de datos de aplicaciones y sitios web, garantizando su correcto funcionamiento.
- 7. Administrador de Redes:** Encargado de gestionar y mantener las infraestructuras de red dentro de una empresa, asegurando su estabilidad y seguridad.
- 8. Consultor Tecnológico:** Orientador y asesor en las empresas, brindando información sobre la implementación de soluciones tecnológicas adecuadas a sus necesidades, mejorando su rendimiento y competitividad.
- 9. Desarrollador de Aplicaciones Móviles:** Gestor del diseño, desarrollo y mantenimiento de aplicaciones para dispositivos móviles, asegurando su usabilidad y rendimiento.
- 10. Ingeniero de Software de Automatización:** Responsable de diseñar y desarrollar soluciones automatizadas para mejorar la eficiencia de procesos empresariales, utilizando programación y tecnología avanzada.

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 intenso 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 en la plataforma de reseñas Trustpilot, 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.





Case Studies

Completarás una selección de los mejores *case studies* de la materia. Casos presentados, analizados y tutorizados por los mejores especialistas del panorama internacional.



Testing & Retesting

Evaluamos y reevaluamos periódicamente tu conocimiento a lo largo del programa. Lo hacemos sobre 3 de los 4 niveles de la Pirámide de Miller.



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 nuestras futuras decisiones difíciles.



Guías rápidas de actuación

TECH ofrece los contenidos más relevantes del curso en forma de fichas o guías rápidas de actuación. Una manera sintética, práctica y eficaz de ayudar al estudiante a progresar en su aprendizaje.



07

Cuadro docente

El cuadro docente está compuesto por profesionales altamente cualificados y con amplia experiencia en el ámbito tecnológico. Cada uno ha sido cuidadosamente seleccionado para garantizar que el alumnado reciba una capacitación de calidad, basada en conocimientos actualizados y en la práctica real del sector. En este sentido, su experiencia laboral en empresas de renombre y su preparación académica permiten ofrecer una perspectiva integral y aplicada sobre la Programación. En definitiva, los mentores son expertos en diversas áreas de la tecnología, lo que proporciona una visión multidisciplinaria y enriquecedora.



“

El cuadro docente es un pilar fundamental en el éxito del programa. La calidad y experiencia de los mentores asegurará una capacitación integral que te preparará para afrontar el sector tecnológico con confianza y capacidad”

Dirección



Dr. Lucas Cuesta, Juan Manuel

- ♦ Ingeniero Senior de Software y Analista en Indizen – Believe in Talent
- ♦ Ingeniero Senior de Software y Analista en Krell Consulting e IMAGiNA Artificial Intelligence
- ♦ Ingeniero de Software en Intel Corporation
- ♦ Ingeniero de Software en Intelligent Dialogue Systems
- ♦ Doctor en Ingeniería Electrónica de Sistemas para Entornos Inteligentes por la Universidad Politécnica de Madrid
- ♦ Graduado en Ingeniería de Telecomunicaciones en la Universidad Politécnica de Madrid
- ♦ Máster en Ingeniería Electrónica de Sistemas para Entornos Inteligentes en la Universidad Politécnica de Madrid



D. Márquez Ruiz de Lacanal, Juan Antonio

- ♦ Desarrollador de software en GTD Defense & Security Solutions
- ♦ Desarrollador de software en Solera Inc
- ♦ Ingeniero de Desarrollo e Investigación en GRVC Sevilla
- ♦ Cofundador de Unmute
- ♦ Cofundador de VR Educa
- ♦ Intercambio académico en Ingeniería y Emprendimiento en la Universidad Berkeley de California
- ♦ Grado en Ingeniería Industrial por la Universidad de Sevilla

Profesores

Dr. Luna Perejón, Francisco

- ◆ Especialista en Arquitectura y Tecnología de Computadores
- ◆ Doctor en Ingeniería Informática por la Universidad de Sevilla
- ◆ Máster Universitario en Ingeniería Informática por la Universidad de Sevilla
- ◆ Grado en Ingeniería de la Salud por la Universidad de Sevilla
- ◆ Grado en Ingeniería Informática y Tecnologías informáticas por la Universidad de Sevilla
- ◆ Miembro de: Grupo de investigación de Robótica y Tecnología Informática (TEP108)

D. Péris Millán, Eduardo

- ◆ Director del Área Tecnológica de Consultoría
- ◆ Especialista en Ingeniería Informática
- ◆ Máster en Gestión estratégica de la información y el conocimiento en las Organizaciones
- ◆ Máster en Liderazgo y Gestión Pública
- ◆ Experto en Gestión Pública
- ◆ Experto en Sistemas de Computación para Smart CITIES

D. Pi Morell, Oriol

- ◆ Analista Funcional en Fihoca
- ◆ Product Owner de Hosting y correo en CDmon
- ◆ Analista Funcional y Software Engineer en Atmira y Capgemini
- ◆ Docente en Capgemini, Forms Capgemini y en Atmira
- ◆ Licenciado en Ingeniería Técnica de Informática de Gestión por la Universidad Autónoma de Barcelona
- ◆ Máster en Inteligencia Artificial por la Universidad Católica de Ávila
- ◆ MBA en Dirección y Administración de Empresas por la IMF Smart Education
- ◆ Máster en Dirección de Sistemas de Información por la IMF Smart Education
- ◆ Postgrado en Patrones de Diseño por la Universitat Oberta de Catalunya

D. Grillo Hernández, José Enrique

- ◆ Desarrollador de aplicaciones y analista tecnológico
- ◆ Desarrollador Senior de Aplicaciones Móviles en Globant
- ◆ Desarrollador Android en Plexus Tech
- ◆ Desarrollador Senior Android en RoadStr
- ◆ Desarrollador Senior Mobile en Avantgarde IT-Information Technology Services
- ◆ Líder de Proyecto en Smartdss
- ◆ Desarrollado en Educatablet
- ◆ Analista de Tecnología en Corporate Mobile Solutions
- ◆ Máster en Ingeniería de Sistema por la Universidad Simón Bolívar

Dña. Domínguez Valderrama, Desirée

- ◆ *Lead Product & Growth Strategist*
- ◆ Máster en Diseño Gráfico y Creatividad por la Escuela de Negocios de la Cámara de Comercio de Sevilla
- ◆ Experta en UX/UI Designer por CoderHouse
- ◆ Experta en Tecnología y Emprendimiento



Aprovecha la oportunidad para conocer los últimos avances en esta materia para aplicarla a tu práctica diaria”

08

Titulación

El Máster Título Propio en Programación desde Cero 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 desde Cero** 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.

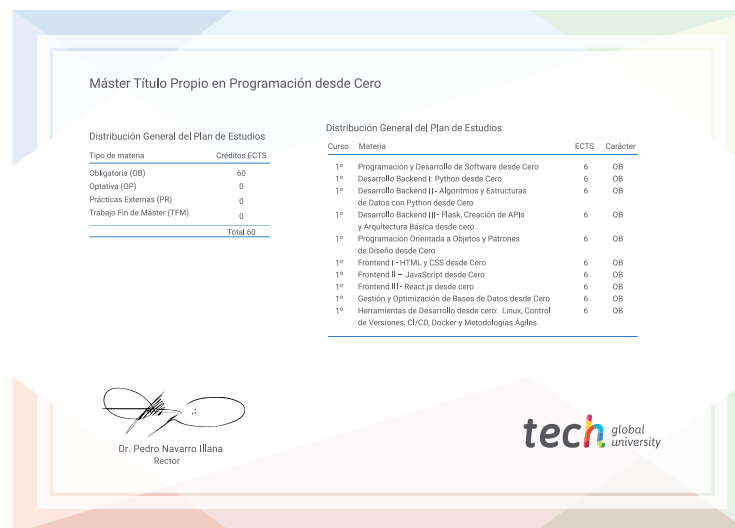
Este doble reconocimiento, de dos destacadas instituciones universitarias, suponen una doble recompensa a una formación integral y de calidad, asegurando que el estudiante obtenga una certificación reconocida tanto a nivel nacional como internacional. Este mérito académico le posicionará como un profesional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

Título: **Máster Título Propio en Programación desde Cero**

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 desde Cero

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

Máster Título Propio

Programación desde Cero

