

Máster Título Propio

Desarrollo en Python

Aval/Membresía



tech
universidad



Máster Título Propio Desarrollo en Python

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 12 meses
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

Acceso web: www.techtitute.com/informatica/master/master-desarrollo-python

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. 22

05

Salidas profesionales

pág. 28

06

Licencias de software incluidas

pág. 32

07

Metodología de estudio

pág. 36

08

Cuadro docente

pág. 46

09

Titulación

pág. 52

01

Presentación del programa

El Desarrollo en Python se ha consolidado como una de las áreas más demandadas en el campo de la programación, siendo utilizado en diversas industrias, desde la inteligencia artificial hasta la automatización de procesos. Según un informe del Departamento de Estadísticas de la Unión Europea, más del 30% de las empresas en la región emplean tecnologías basadas en Python para sus operaciones tecnológicas. Este fenómeno ha impulsado a TECH para la creación de un programa universitario que surge como respuesta a la necesidad de brindar una capacitación moderna y accesible. De hecho, la metodología empleada, a través de material didáctico 100% online, permitirá que los egresados se sumerjan en una capacitación flexible y de calidad.



“

*Un programa exhaustivo y 100% online,
exclusivo de TECH y con una perspectiva
internacional respaldada por nuestra
afiliación con Python Software Foundation”*

El lenguaje de programación Python ha emergido como una herramienta esencial en diversas industrias debido a su versatilidad y eficiencia. Actualmente, es ampliamente utilizado en áreas como inteligencia artificial, análisis de datos, desarrollo web y automatización de procesos. Gracias a su sintaxis sencilla y clara, Python ha ganado popularidad tanto entre desarrolladores novatos como entre profesionales experimentados.

En este contexto, TECH profundizará en el Desarrollo en Python mediante un novedoso plan de estudios el cual abordará la ejecución de programas para optimizar el rendimiento de las aplicaciones. Posteriormente, se trabajará en la programación orientada a objetos, facilitando la creación de software modular y reutilizable. Finalmente, se explorará el tratamiento de Big Data, lo que permitirá a los profesionales gestionar y analizar grandes volúmenes de información, mejorando la toma de decisiones. Este enfoque integral proporcionará herramientas clave para desenvolverse en entornos de desarrollo ágiles y dinámicos.

Además, al conocer herramientas como Python y Big Data, los profesionales estarán mejor posicionados para asumir roles clave en empresas que requieren habilidades para optimizar procesos y generar valor a través de la tecnología. Por otro lado, la metodología de TECH Universidad está diseñada para adaptarse a las necesidades del mundo moderno, utilizando un enfoque 100% online, accesible las 24 horas del día, los 7 días de la semana, y desde cualquier dispositivo con conexión a internet. Este modelo, basado en *Relearning*, permite una capacitación flexible y personalizada, lo que asegurará la asimilación de conceptos a través de la repetición y la práctica continua. Además, un prestigioso Director Invitado Internacional impartirá 10 exclusivas *Masterclasses*.

Asimismo, gracias a que TECH Universidad es miembro de **Python Software Foundation (PSF)**, el profesional contará con materiales especializados, guías y ejercicios avanzados para la práctica en este sector. Además, podrá asistir a eventos académicos, recibir descuentos en publicaciones y conectarse con una amplia red internacional de destacados investigadores, reforzando el conocimiento en este campo.

Este **Máster Título Propio en Desarrollo en Python** contiene el programa universitario más completo y actualizado del mercado. Sus características más destacadas son:

- ♦ El desarrollo de casos prácticos presentados por expertos en Desarrollo en Python
- ♦ 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
- ♦ 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 reconocido Director Invitado Internacional ofrecerá 10 rigurosas Masterclasses acerca de las últimas tendencias en el Desarrollo en Python”

“

Obtendrás competencias avanzadas para trabajar con metodologías ágiles, control de versiones y herramientas colaborativas propias de entornos profesionales”

Incluye en su cuadro docente a profesionales pertenecientes al ámbito del Desarrollo en Python, 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.

Dominarás las principales bibliotecas, frameworks y entornos de desarrollo asociados a Python en áreas como el desarrollo web.

Manejarás las técnicas de big data más modernas para optimizar el análisis y la toma de decisiones estratégicas.



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 Wistuba, 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 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 diez idiomas distintos, que nos convierten en la mayor institución educativa del mundo.

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.

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

Este exclusivo programa universitario ahondará en conceptos clave del Desarrollo en Python, comenzando con la ejecución eficiente de programas para mejorar el rendimiento de las aplicaciones. Además, se profundizará en la programación orientada a objetos, lo que permitirá la creación de software modular, reutilizable y fácil de mantener. A lo largo del itinerario, se abordará también el uso de *strings*, brindando herramientas para gestionar y manipular datos de manera más eficiente. Estas competencias, combinadas, capacitarán a los profesionales para desarrollar aplicaciones robustas y optimizadas, enfrentando con éxito los retos del entorno digital actual.





“

Profundizarás en los últimos avances sobre el uso de strings, mejorando la gestión de datos en aplicaciones de desarrollo”

Módulo 1. Programación en Python

- 1.1. Creación y ejecución de programas en Python
 - 1.1.1. Configuración del entorno de desarrollo
 - 1.1.2. Ejecución de *scripts* en Python
 - 1.1.3. Herramientas de Desarrollo Integrado (IDEs)
- 1.2. Datos en Python
 - 1.2.1. Tipos primitivos (*int*, *float*, *str*)
 - 1.2.2. Conversión y *casting* de tipos de datos en Python
 - 1.2.3. Inmutabilidad y almacenamiento de datos en Python
- 1.3. Referencias a objetos en Python
 - 1.3.1. Referencias en memoria
 - 1.3.2. Identidad vs. igualdad
 - 1.3.3. Gestión de referencias y recolección de basura
- 1.4. Datos de colección en Python
 - 1.4.1. Listas y operaciones comunes
 - 1.4.2. Tuplas y su inmutabilidad
 - 1.4.3. Diccionarios y acceso a datos
- 1.5. Operaciones lógicas en Python
 - 1.5.1. Operadores booleanos
 - 1.5.2. Expresiones condicionales
 - 1.5.3. *Short-Circuit Evaluation*
- 1.6. Operadores aritméticos en Python
 - 1.6.1. Operaciones aritméticas en Python
 - 1.6.2. Operadores de división
 - 1.6.3. Precedencia y asociatividad
- 1.7. Entrada/salida en Python
 - 1.7.1. Lectura de datos desde la entrada estándar
 - 1.7.2. Escritura de datos a la salida estándar
 - 1.7.3. Manejo de archivos
- 1.8. Creación y llamada de funciones en Python
 - 1.8.1. Sintaxis de funciones
 - 1.8.2. Parámetros y argumentos
 - 1.8.3. Valores de retorno y funciones anónimas



- 1.9. Uso de *strings* en Python
 - 1.9.1. Manipulación y formateo de *strings*
 - 1.9.2. Métodos comunes de *strings*
 - 1.9.3. Interpolación y F - *strings*
- 1.10. Gestión de errores y excepciones en Python
 - 1.10.1. Tipos comunes de excepciones
 - 1.10.2. Bloques *try - except*
 - 1.10.3. Creación de excepciones personalizadas

Módulo 2. Datos Avanzados y Control del Flujo con Python

- 2.1. Identificadores y palabras clave en Python
 - 2.1.1. Reglas para nombres de variables
 - 2.1.2. Palabras reservadas en Python
 - 2.1.3. Convenciones de nomenclatura
- 2.2. Tipos integrales y booleanos en Python
 - 2.2.1. Tipos integrales
 - 2.2.2. Operaciones específicas de booleanos
 - 2.2.3. Conversiones y representaciones
- 2.3. Tipos de punto flotante y números complejos en Python
 - 2.3.1. Precisión y representación
 - 2.3.2. Operaciones con punto flotante
 - 2.3.3. Uso de números complejos en cálculos
- 2.4. Formateo de *strings* y codificaciones en Python
 - 2.4.1. Métodos avanzados de formateo
 - 2.4.2. Codificaciones *Unicode* y UTF - 8
 - 2.4.3. Trabajo con caracteres especiales
- 2.5. Colecciones: tuplas, listas y diccionarios en Python
 - 2.5.1. Comparación y contraste entre tipos
 - 2.5.2. Métodos específicos de cada tipo
 - 2.5.3. Eficiencia y selección de tipo adecuado

- 2.6. Sets y *Frozen Sets* en Python
 - 2.6.1. Creación y operaciones en Sets
 - 2.6.2. *Frozen Sets*
 - 2.6.3. Aplicaciones prácticas y rendimiento
- 2.7. Iteración y copiado de colecciones en Python
 - 2.7.1. Bucles *for* y comprensiones de lista
 - 2.7.2. Copia superficial vs. profunda
 - 2.7.3. Iteradores y generadores
- 2.8. Uso de funciones *Lambda* en Python
 - 2.8.1. Sintaxis y creación de funciones *Lambda*
 - 2.8.2. Aplicaciones en filtros y mapas
 - 2.8.3. Limitaciones y buenas prácticas
- 2.9. Estructuras de control: Condicionales y bucles en Python
 - 2.9.1. Estructuras *if - else* y *elif*
 - 2.9.2. Bucles *while* y *for*
 - 2.9.3. Control de flujo con *break*, *continue* y *else*
- 2.10. Funciones y métodos avanzados en Python
 - 2.10.1. Funciones recursivas
 - 2.10.2. Funciones de orden superior
 - 2.10.3. Decoradores de funciones

Módulo 3. Programación Orientada a Objetos (POO) en Python

- 3.1. Programación orientada a objetos (POO) en Python
 - 3.1.1. Clases y objetos
 - 3.1.2. Encapsulación y abstracción
 - 3.1.3. Programación orientada a objetos (POO) en Python
- 3.2. Creación de clases y objetos en Python
 - 3.2.1. Clases en POO en Python
 - 3.2.2. Instanciación y métodos de inicialización
 - 3.2.3. Atributos y métodos

- 3.3. Atributos y métodos en Python
 - 3.3.1. Atributos de instancia vs. Clase
 - 3.3.2. Métodos de instancia, clase y estáticos
 - 3.3.3. Encapsulación y ocultamiento de información
- 3.4. Herencia y polimorfismo en Python
 - 3.4.1. Herencia simple y múltiple
 - 3.4.2. Sobreescritura y extensión de métodos
 - 3.4.3. Polimorfismo y *Duck Typing*
- 3.5. Propiedades y acceso a atributos en Python
 - 3.5.1. *Getters* y *Setters*
 - 3.5.2. Decorador *@property*
 - 3.5.3. Control de acceso y validación
- 3.6. Clases y colecciones personalizadas en Python
 - 3.6.1. Creación de tipos de colección
 - 3.6.2. Métodos especiales (*__len__*, *__getitem__*)
 - 3.6.3. Iteradores personalizados
- 3.7. Agregación y composición en clases en Python
 - 3.7.1. Relaciones entre clases
 - 3.7.2. Agregación vs composición
 - 3.7.3. Gestión del ciclo de vida de objetos
- 3.8. Uso de decoradores en clases en Python
 - 3.8.1. Decoradores para métodos
 - 3.8.2. Decoradores de clases
 - 3.8.3. Aplicaciones y casos de uso
- 3.9. Clases abstractas y métodos en Python
 - 3.9.1. Clases abstractas
 - 3.9.2. Métodos abstractos e implementación
 - 3.9.3. Uso de ABC (*Abstract Base Class*)
- 3.10. Excepciones y manejo de errores en POO en Python
 - 3.10.1. Excepciones personalizadas en clases
 - 3.10.2. Manejo de excepciones en métodos
 - 3.10.3. Buenas prácticas en excepciones y POO

Módulo 4. Desarrollo de Aplicaciones en Python

- 4.1. Arquitectura de aplicaciones en Python
 - 4.1.1. Diseño de Software
 - 4.1.2. Patrones arquitectónicos comunes
 - 4.1.3. Evaluación de requerimientos y necesidades
- 4.2. Diseño y modelado de aplicaciones en Python
 - 4.2.1. Uso de UML y diagramas
 - 4.2.2. Modelado de datos y flujo de información
 - 4.2.3. Principios SOLID y diseño modular
- 4.3. Gestión de dependencias y librerías en Python
 - 4.3.1. Manejo de paquetes con Pip
 - 4.3.2. Uso de entornos virtuales
 - 4.3.3. Resolución de conflictos de dependencias
- 4.4. Patrones de diseño en desarrollo en Python
 - 4.4.1. Patrones creacionales, estructurales y de comportamiento
 - 4.4.2. Aplicación práctica de patrones
 - 4.4.3. Refactorización y patrones
- 4.5. Pruebas y *Debugging* en aplicaciones en Python
 - 4.5.1. Estrategias de *Testing* (Unitario, Integración)
 - 4.5.2. Uso de *Frameworks* de pruebas
 - 4.5.3. Técnicas de *Debugging* y herramientas
- 4.6. Seguridad y autenticación en Python
 - 4.6.1. Seguridad en aplicaciones
 - 4.6.2. Implementación de autenticación y autorización
 - 4.6.3. Prevención de vulnerabilidades
- 4.7. Optimización y rendimiento de aplicaciones en Python
 - 4.7.1. Análisis de rendimiento
 - 4.7.2. Técnicas de optimización de código
 - 4.7.3. Manejo eficiente de recursos y datos
- 4.8. Despliegue y distribución de aplicaciones en Python
 - 4.8.1. Estrategias de despliegue
 - 4.8.2. Uso de contenedores y orquestadores
 - 4.8.3. Distribución y actualizaciones continuas

- 4.9. Mantenimiento y actualización en Python
 - 4.9.1. Gestión del ciclo de vida del Software
 - 4.9.2. Estrategias de mantenimiento y refactorización
 - 4.9.3. Actualización y migración de sistemas
- 4.10. Documentación y soporte técnico en Python
 - 4.10.1. Creación de documentación efectiva
 - 4.10.2. Herramientas para la documentación
 - 4.10.3. Estrategias de soporte y comunicación con usuarios

Módulo 5. Desarrollo Web y Móvil con Python

- 5.1. Desarrollo web con Python
 - 5.1.1. Estructura y componentes de una web
 - 5.1.2. Tecnologías en desarrollo web
 - 5.1.3. Tendencias en desarrollo web
- 5.2. *Frameworks* web populares con Python
 - 5.2.1. *Django*, *Flask* y otras opciones
 - 5.2.2. Comparativa y selección de *Frameworks*
 - 5.2.3. Integración con *Frontend*
- 5.3. Desarrollo *Frontend*: HTML, CSS y JavaScript con Python
 - 5.3.1. HTML y CSS
 - 5.3.2. JavaScript y manipulación del DOM
 - 5.3.3. *Frameworks* y librerías *Frontend*
- 5.4. *Backend* y bases de datos con Python
 - 5.4.1. Desarrollo *Backend* con Python
 - 5.4.2. Gestión de bases de datos relacionales y no relacionales
 - 5.4.3. Integración *Backend-Frontend*
- 5.5. APIs y servicios web con Python
 - 5.5.1. Diseño de APIs *RESTful*
 - 5.5.2. Implementación y documentación de APIs
 - 5.5.3. Consumo y seguridad en APIs
- 5.6. Desarrollo móvil con Python
 - 5.6.1. Plataformas de desarrollo móvil (Nativas, Híbridas)
 - 5.6.2. Herramientas y entornos de desarrollo
 - 5.6.3. Adaptación de aplicaciones para dispositivos móviles

- 5.7. Plataformas de desarrollo móvil con Python
 - 5.7.1. Android y IOS
 - 5.7.2. *Frameworks* para desarrollo cruzado
 - 5.7.3. Pruebas y *Deployment* en dispositivos móviles
- 5.8. Diseño y UX en aplicaciones móviles con Python
 - 5.8.1. Diseño de interfaces móviles
 - 5.8.2. Usabilidad y experiencia de usuario con Python
 - 5.8.3. Herramientas de prototipado y diseño
- 5.9. Pruebas y depuración en móviles con Python
 - 5.9.1. Estrategias de *Testing* en dispositivos móviles
 - 5.9.2. Herramientas de depuración y monitoreo
 - 5.9.3. Automatización de pruebas
- 5.10. Publicación en tiendas de aplicaciones con Python
 - 5.10.1. Proceso de publicación en App Store y Google Play
 - 5.10.2. Cumplimiento y políticas de las aplicaciones
 - 5.10.3. Estrategias de Marketing y promoción

Módulo 6. Interfaz y Experiencia de Usuario con Python

- 6.1. Diseño de interfaz de usuario con Python
 - 6.1.1. Diseño UI con Python
 - 6.1.2. Interacción usuario - computadora con Python
 - 6.1.3. Diseño centrado en el usuario con Python
- 6.2. Herramientas de diseño UI/UX con Python
 - 6.2.1. Software de diseño y prototipado
 - 6.2.2. Herramientas de colaboración y *Feedback*
 - 6.2.3. Integración de diseño en el proceso de desarrollo
- 6.3. Diseño responsivo y adaptativo con Python
 - 6.3.1. Técnicas de diseño responsivo
 - 6.3.2. Adaptación a diferentes dispositivos y pantallas
 - 6.3.3. *Testing* y aseguramiento de la calidad
- 6.4. Animaciones y transiciones con Python
 - 6.4.1. Creación de animaciones efectivas con Python
 - 6.4.2. Herramientas y librerías para animaciones
 - 6.4.3. Impacto en la UX y rendimiento

- 6.5. Accesibilidad y usabilidad con Python
 - 6.5.1. Accesibilidad web
 - 6.5.2. Herramientas y técnicas de evaluación
 - 6.5.3. Implementación de mejores prácticas
- 6.6. Prototipado y *Wireframes* con Python
 - 6.6.1. Creación de *Wireframes* y Mockups
 - 6.6.2. Herramientas de prototipado rápido
 - 6.6.3. *Tests* de usabilidad y *Feedback*
- 6.7. Pruebas de usabilidad con Python
 - 6.7.1. Métodos y técnicas de testeo de usabilidad
 - 6.7.2. Análisis y mejoras basados en resultados
 - 6.7.3. Herramientas para pruebas de usabilidad
- 6.8. Análisis de comportamiento del usuario con Python
 - 6.8.1. Técnicas de análisis y *Tracking*
 - 6.8.2. Interpretación de datos y métricas
 - 6.8.3. Mejora continua basada en datos
- 6.9. Mejoras basadas en *Feedback* con Python
 - 6.9.1. Gestión y análisis de *Feedback*
 - 6.9.2. Ciclos de *Feedback* y mejora continua
 - 6.9.3. Estrategias para implementar cambios efectivos
- 6.10. Tendencias futuras en UI/UX con Python
 - 6.10.1. Innovaciones y tendencias emergentes
 - 6.10.2. Impacto de nuevas tecnologías en UI/UX
 - 6.10.3. Preparación para el futuro del diseño

Módulo 7. Procesamiento de Datos y *Big Data* con Python

- 7.1. Uso de Python en datos
 - 7.1.1. Python en ciencia de datos y análisis
 - 7.1.2. Bibliotecas esenciales para datos
 - 7.1.3. Aplicaciones y ejemplos
- 7.2. Configuración del entorno de desarrollo con Python
 - 7.2.1. Instalación de Python y herramientas
 - 7.2.2. Configuración de entornos virtuales
 - 7.2.3. Herramientas de Desarrollo Integrado (IDE)

- 7.3. Variables, tipos de datos y operadores en Python
 - 7.3.1. Variables y tipos de datos primitivos
 - 7.3.2. Estructuras de datos
 - 7.3.3. Operadores aritméticos y lógicos
- 7.4. Control de Flujo: Condicionales y bucles
 - 7.4.1. Estructuras de control condicionales (*if, else, elif*)
 - 7.4.2. Bucles (*for, while*) y control de flujo
 - 7.4.3. Comprensiones de lista y expresiones generadoras
- 7.5. Funciones y modularidad con Python
 - 7.5.1. Uso de funciones
 - 7.5.2. Parámetros, argumentos y valores de retorno
 - 7.5.3. Modularidad y reutilización de código
- 7.6. Manejo de errores y excepciones con Python
 - 7.6.1. errores y excepciones
 - 7.6.2. Manejo de excepciones con try-except
 - 7.6.3. Creación de excepciones personalizadas
- 7.7. Herramienta *IPython*
 - 7.7.1. Herramienta *IPython*
 - 7.7.2. Uso de *IPython* para análisis de datos
 - 7.7.3. Diferencias con el intérprete estándar de Python
- 7.8. *Jupyter* Notebooks
 - 7.8.1. *Jupyter* Notebooks
 - 7.8.2. Uso de cuadernos para análisis de datos
 - 7.8.3. Publicación de cuadernos *Jupyter*
- 7.9. Mejores prácticas de codificación en Python
 - 7.9.1. Estilo y convenciones (PEP 8)
 - 7.9.2. Documentación y comentarios
 - 7.9.3. Estrategias de pruebas y depuración
- 7.10. Recursos y comunidades de Python
 - 7.10.1. Recursos en línea y documentación
 - 7.10.2. Comunidades y foros
 - 7.10.3. Aprendizaje y actualización en Python

Módulo 8. Estructuras de Datos y Funciones en Python

- 8.1. Conjuntos en Python
 - 8.1.1. Operaciones y métodos
 - 8.1.2. Diferencias y aplicación práctica
 - 8.1.3. Iteración y comprensiones
- 8.2. Diccionarios y su uso en Python
 - 8.2.1. Creación y manipulación de diccionarios
 - 8.2.2. Acceso y gestión de datos
 - 8.2.3. Patrones y técnicas avanzadas
- 8.3. Comprensiones de listas y diccionarios en Python
 - 8.3.1. Sintaxis y ejemplos
 - 8.3.2. Eficiencia y legibilidad
 - 8.3.3. Aplicaciones prácticas
- 8.4. Funciones en datos en Python
 - 8.4.1. Creación de funciones
 - 8.4.2. Ámbito y espacio de nombres
 - 8.4.3. Funciones anónimas y *Lambda*
- 8.5. Argumentos de funciones y retorno de valores en Python
 - 8.5.1. Argumentos posicionales y nombrados
 - 8.5.2. Valores de retorno múltiples
 - 8.5.3. Argumentos variables y palabras clave
- 8.6. Funciones *Lambda* y funciones de orden superior en Python
 - 8.6.1. Uso de funciones *Lambda*
 - 8.6.2. Funciones *Map*, *Filter* y *Reduce*
 - 8.6.3. Aplicaciones en procesamiento de datos
- 8.7. Manejo de archivos en Python
 - 8.7.1. Lectura y escritura de archivos
 - 8.7.2. Manejo de archivos binarios y de texto
 - 8.7.3. Buenas prácticas y manejo de excepciones

- 8.8. Lectura y escritura de archivos de texto y binarios en Python
 - 8.8.1. Formatos de archivo y codificación
 - 8.8.2. Manejo de archivos grandes
 - 8.8.3. Serialización y deserialización (JSON, *pickle*)
- 8.9. Contextos y operaciones con archivos
 - 8.9.1. Uso del administrador de contexto (*with*)
 - 8.9.2. Técnicas de procesamiento de archivos
 - 8.9.3. Seguridad y manejo de errores
- 8.10. Bibliotecas de modelado en Python
 - 8.10.1. *Scikit - learn*
 - 8.10.2. *TensorFlow*
 - 8.10.3. *PyTorch*

Módulo 9. Manejo de Datos en Python con NumPy y Pandas

- 9.1. Creación y manipulación de *Arrays* en *NumPy*
 - 9.1.1. *NumPy*
 - 9.1.2. Operaciones básicas con *Arrays*
 - 9.1.3. Manipulación y transformación de *Arrays*
- 9.2. Operaciones vectorizadas con *Arrays*
 - 9.2.1. Vectorización
 - 9.2.2. Funciones universales (*ufunc*)
 - 9.2.3. Eficiencia y rendimiento
- 9.3. Indexación y segmentación en *NumPy*
 - 9.3.1. Acceso a elementos y *Slicing*
 - 9.3.2. Indexación avanzada y booleana
 - 9.3.3. Reordenamiento y selección
- 9.4. *Pandas* series y *DataFrames*
 - 9.4.1. *Pandas*
 - 9.4.2. Estructuras de datos en *Pandas*
 - 9.4.3. Manipulación de *DataFrames*
- 9.5. Indexación y selección en *Pandas*
 - 9.5.1. Acceso a datos en series y *DataFrames*
 - 9.5.2. Métodos de selección y filtrado
 - 9.5.3. Uso de *loc* e *iloc*

- 9.6. Operaciones con *Pandas*
 - 9.6.1. Operaciones aritméticas y alineación
 - 9.6.2. Funciones de agregación y estadísticas
 - 9.6.3. Transformaciones y aplicación de funciones
- 9.7. Manejo de datos incompletos en *Pandas*
 - 9.7.1. Detección y manejo de valores nulos
 - 9.7.2. Llenado y eliminación de datos incompletos
 - 9.7.3. Estrategias para el manejo de datos incompletos
- 9.8. Funciones y aplicaciones en *Pandas*
 - 9.8.1. Concatenación y fusión de datos
 - 9.8.2. Agrupación y agregación (*groupby*)
 - 9.8.3. *Pivot Tables* y *Crosstabs*
- 9.9. Visualización con *Matplotlib*
 - 9.9.1. *Matplotlib*
 - 9.9.2. Creación de gráficos y personalización
 - 9.9.3. Integración con *Pandas*
- 9.10. Personalización de gráficos en *Matplotlib*
 - 9.10.1. Estilos y configuraciones
 - 9.10.2. Gráficos avanzados (*scatter*, *bar*, etc.)
 - 9.10.3. Creación de visualizaciones complejas

Módulo 10. Técnicas Avanzadas y Aplicaciones Prácticas en NumPy y Pandas

- 10.1. Carga de datos desde diferentes fuentes
 - 10.1.1. Importación desde CSV, Excel y bases de datos
 - 10.1.2. Lectura de datos de APIs y web
 - 10.1.3. Estrategias de manejo de datos grandes
- 10.2. Almacenamiento de datos en Python
 - 10.2.1. Exportación a diferentes formatos
 - 10.2.2. Eficiencia en almacenamiento
 - 10.2.3. Seguridad y privacidad de datos
- 10.3. Estrategias de limpieza de datos en Python
 - 10.3.1. Identificación y corrección de inconsistencias
 - 10.3.2. Normalización y transformación de datos
 - 10.3.3. Automatización de procesos de limpieza

- 10.4. Transformación avanzada de datos en Pandas
 - 10.4.1. Técnicas de manipulación y transformación
 - 10.4.2. Combinación y reestructuración de *DataFrames*
 - 10.4.3. Uso de expresiones regulares en Pandas
- 10.5. Combinación de *DataFrames* en Pandas
 - 10.5.1. *Merge*, *Join* y concatenación
 - 10.5.2. Manejo de conflictos y claves
 - 10.5.3. Estrategias de combinación eficiente
- 10.6. Transformación avanzada y pivotado de datos en Pandas
 - 10.6.1. *Pivot* y *Melt*
 - 10.6.2. Técnicas de *Reshape* y transposición
 - 10.6.3. Aplicaciones en análisis de datos
- 10.7. Series temporales en Pandas
 - 10.7.1. Manejo de fechas y tiempos
 - 10.7.2. *Resampling* y *Window Functions*
 - 10.7.3. Análisis de tendencias y estacionalidad
- 10.8. Manejo avanzado de índices en Pandas
 - 10.8.1. Índices multinivel y jerárquicos
 - 10.8.2. Selección y manipulación avanzada
 - 10.8.3. Optimización de consultas
- 10.9. Estrategias de optimización de rendimiento
 - 10.9.1. Mejoras en velocidad y eficiencia
 - 10.9.2. Uso de Cython y Numba
 - 10.9.3. Paralelización y procesamiento distribuido
- 10.10. Proyectos prácticos de manipulación de datos
 - 10.10.1. Desarrollo de ejemplos de uso reales
 - 10.10.2. Integración de técnicas de Python
 - 10.10.3. Estrategias para la solución de problemas de datos complejos

04

Objetivos docentes

Este Máster Título Propio tiene como finalidad proporcionar las competencias necesarias para aplicar el desarrollo en Python de manera precisa. Por lo tanto, los profesionales se especializarán en configurar entornos de desarrollo adecuados, optimizando el inicio y la ejecución de proyectos. Además, profundizarán en el manejo de datos en Python, mejorando la manipulación y análisis de grandes volúmenes de información. Como resultado, la comprensión de las referencias en memoria les permitirá gestionar de manera eficiente los recursos, mejorando el rendimiento de las aplicaciones.



“

*Evaluarás el impacto de las referencias
en memoria, gestionando de manera
eficiente los recursos del sistema”*



Objetivos generales

- ♦ Desarrollar habilidades en la programación en Python, facilitando la creación de aplicaciones robustas y eficientes
- ♦ Adquirir competencias avanzadas en el manejo de datos y control del flujo de ejecución en Python
- ♦ Dominar la programación orientada a objetos (POO) para desarrollar *software* modular y reutilizable
- ♦ Aplicar técnicas de desarrollo en Python para la creación de aplicaciones funcionales y escalables
- ♦ Implementar soluciones de desarrollo web y móvil utilizando herramientas y bibliotecas de Python
- ♦ Mejorar la experiencia del usuario mediante el diseño de interfaces interactivas en Python
- ♦ Optimizar el procesamiento de grandes volúmenes de datos con Python, utilizando herramientas como NumPy y Pandas
- ♦ Aplicar técnicas avanzadas en NumPy y Pandas para resolver problemas complejos en el manejo y análisis de datos





Objetivos específicos

Módulo 1. Programación en Python

- ♦ Configurar entornos de desarrollo y ejecutar scripts en Python utilizando herramientas de desarrollo integrado (IDEs) para optimizar el flujo de trabajo
- ♦ Manipular y gestionar tipos de datos en Python, incluyendo la conversión de tipos, inmutabilidad y almacenamiento, para una programación más eficiente
- ♦ Comprender y aplicar el manejo de referencias en memoria, diferenciando entre identidad e igualdad y gestionando la recolección de basura para mejorar el rendimiento
- ♦ Desarrollar habilidades en la creación de funciones y el manejo de errores y excepciones, implementando estructuras adecuadas para garantizar la estabilidad del código

Módulo 2. Datos Avanzados y Control del Flujo con Python

- ♦ Aplicar las reglas de nomenclatura y los identificadores en Python para una programación eficiente
- ♦ Utilizar correctamente los tipos integrales y booleanos, gestionando su precisión y operaciones
- ♦ Manipular colecciones como listas, tuplas y diccionarios, eligiendo el tipo adecuado para cada caso
- ♦ Emplear funciones Lambda y estructuras de control avanzadas para optimizar el código en Python

Módulo 3. Programación Orientada a Objetos (POO) en Python

- ♦ Crear clases y objetos en Python, aplicando conceptos fundamentales de la programación orientada a objetos
- ♦ Implementar herencia y polimorfismo para reutilizar y extender funcionalidades en proyectos en Python
- ♦ Utilizar decoradores para métodos y clases, optimizando la funcionalidad y reutilización del código
- ♦ Gestionar excepciones personalizadas y aplicar buenas prácticas para el manejo de errores en la programación orientada a objetos

Módulo 4. Desarrollo de Aplicaciones en Python

- ♦ Diseñar y modelar aplicaciones en Python utilizando principios de UML, diagramas y metodologías ágiles como SOLID
- ♦ Implementar estrategias de pruebas y *debugging* en aplicaciones Python, mejorando la calidad y funcionalidad del software
- ♦ Optimizar el rendimiento de aplicaciones en Python, aplicando técnicas de análisis y manejo eficiente de recursos
- ♦ Gestionar el ciclo de vida del software, incluyendo el despliegue, mantenimiento y actualización de aplicaciones en Python

Módulo 5. Desarrollo Web y Móvil con Python

- ♦ Desarrollar aplicaciones web con Python utilizando *frameworks* como Django y Flask, integrando tanto el *backend* como el *frontend*
- ♦ Crear y consumir APIs RESTful, asegurando la correcta integración y seguridad en el intercambio de datos entre aplicaciones
- ♦ Desarrollar aplicaciones móviles con Python, adaptando las soluciones para plataformas como Android e iOS, utilizando herramientas y *frameworks* adecuados
- ♦ Implementar pruebas y estrategias de depuración en aplicaciones móviles con Python, garantizando la estabilidad y funcionalidad en dispositivos reales

Módulo 6. Interfaz y Experiencia de Usuario con Python

- ♦ Diseñar interfaces de usuario (UI) con Python, aplicando principios de diseño centrado en el usuario para optimizar la interacción usuario - computadora
- ♦ Implementar técnicas de diseño responsivo y adaptativo en Python, garantizando una experiencia de usuario coherente en diferentes dispositivos y pantallas
- ♦ Crear animaciones y transiciones eficaces en Python, evaluando su influencia en la experiencia de usuario y el rendimiento
- ♦ Realizar pruebas de usabilidad con Python, utilizando herramientas de análisis de comportamiento del usuario para generar mejoras a partir de los resultados obtenidos

Módulo 7. Procesamiento de Datos y Big Data con Python

- ♦ Aplicar Python en ciencia de datos y análisis con bibliotecas especializadas
- ♦ Configurar entornos de desarrollo y herramientas como IDE y entornos virtuales
- ♦ Optimizar el control de flujo con estructuras condicionales y bucles
- ♦ Mejorar las prácticas de codificación siguiendo PEP 8 y documentando el código

Módulo 8. Estructuras de Datos y Funciones en Python

- ♦ Manipular conjuntos, diccionarios y aplicar técnicas avanzadas de iteración y comprensión
- ♦ Crear y gestionar funciones, incluyendo el uso de funciones Lambda y de orden superior
- ♦ Manejar archivos de texto y binarios, optimizando el procesamiento con herramientas como *JSON* y *Pickle*
- ♦ Aplicar bibliotecas como *Scikit - learn*, *TensorFlow* y *PyTorch* para modelado y procesamiento de datos

Módulo 9. Manejo de Datos en Python con NumPy y Pandas

- ♦ Crear, manipular y transformar *arrays* con NumPy, optimizando operaciones básicas y avanzadas
- ♦ Realizar operaciones vectorizadas y optimizar el rendimiento utilizando funciones universales (*ufunc*)
- ♦ Utilizar Pandas para manejar series, *DataFrames*, y aplicar métodos de indexación, selección y filtrado
- ♦ Generar visualizaciones con Matplotlib, personalizando gráficos y creando visualizaciones complejas e integradas con Pandas

Módulo 10. Técnicas Avanzadas y Aplicaciones Prácticas en NumPy y Pandas

- ♦ Cargar y gestionar datos desde diversas fuentes como CSV, Excel, bases de datos y APIs, optimizando el manejo de grandes volúmenes
- ♦ Exportar y almacenar datos en diferentes formatos, asegurando eficiencia, seguridad y privacidad
- ♦ Limpiar y transformar datos mediante técnicas avanzadas, como la normalización, corrección de inconsistencias y automatización de procesos
- ♦ Manipular y transformar datos en Pandas mediante técnicas avanzadas como *Merge*, *Pivot* y *Melt*, mejorando la eficiencia del análisis



Tendrás una sólida comprensión sobre los principios avanzados para la creación de gráficos con Matplotlib, optimizando la visualización de datos”

05

Salidas profesionales

Esta titulación universitaria brinda la experiencia académica más completa del mercado actual. Gracias a ello, los egresados podrán acceder a una amplia gama de roles en sectores clave, como analistas de datos, consultores en inteligencia empresarial o desarrolladores de soluciones visuales. De hecho, los conceptos adquiridos les permitirá interpretar y presentar datos complejos con precisión, posicionándolos como profesionales altamente demandados en empresas tecnológicas, consultoras y organismos internacionales. De este modo, estarán preparados para abordar desafíos reales y aportar soluciones innovadoras en el ámbito del Desarrollo en Python.



“

*Fortalecerás tu perfil profesional
desempeñándote como desarrollador
de soluciones visuales, rol clave en la
transformación digital de las empresas”*

Perfil del egresado

El egresado será capaz de diseñar y desarrollar soluciones avanzadas en Python, integrando herramientas y bibliotecas especializadas para el manejo de grandes volúmenes de datos. Por lo tanto, su dominio de las técnicas de programación orientada a objetos y su capacidad para crear aplicaciones escalables le permitirán enfrentar desafíos complejos en diversos sectores. Además, contará con habilidades para implementar algoritmos eficientes, optimizar el rendimiento y automatizar procesos. Este perfil, orientado al análisis y la inteligencia de datos, le permitirá liderar proyectos innovadores y adaptarse a los avances tecnológicos en el campo del desarrollo.

Desarrollarás herramientas efectivas para la realización de pruebas automáticas, migración de datos y scraping web.

- ♦ **Resolución de problemas complejos:** abordar y resolver desafíos técnicos de manera eficiente
- ♦ **Pensamiento lógico y analítico:** descomponer problemas en partes más pequeñas y gestionar la información de forma estructurada
- ♦ **Trabajo en equipo multidisciplinario:** colaborar con otros desarrolladores, diseñadores y gestores de proyectos es fundamental
- ♦ **Gestión de proyectos y tiempo:** organizar tareas, establecer prioridades y gestionar plazos en el desarrollo de proyectos





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

1. **Desarrollador de Software:** encargado de diseñar, desarrollar y mantener aplicaciones de *software* escalables en diferentes plataformas.
2. **Analista de Datos:** responsable en recolectar, procesar y analizar datos utilizando herramientas como Python y bibliotecas para extraer información clave que facilite la toma de decisiones.
3. **Ingeniero de Datos:** dedicado a construir y mantener sistemas de almacenamiento y procesamiento de datos a gran escala.
4. **Desarrollador de Inteligencia Artificial:** encargado de crear algoritmos y sistemas basados en IA. Utiliza Python para desarrollar y entrenar modelos de aprendizaje automático, redes neuronales y otros sistemas inteligentes.
5. **Desarrollador de Backend:** responsable de crear y mantener la lógica de servidor, bases de datos y aplicaciones web.
6. **Ingeniero de Machine Learning:** gestor de modelos y sistemas de aprendizaje automático utilizando Python.
7. **Consultor de Big Data:** asesor de organizaciones para manejar grandes volúmenes de datos utilizando tecnologías como Python, optimizando la recolección, almacenamiento y análisis de datos a gran escala.
8. **Arquitecto de Soluciones:** responsable de diseñar y planificar la infraestructura tecnológica necesaria para soportar aplicaciones complejas.

06

Licencias de software incluidas

TECH es referencia en el mundo universitario por combinar la última tecnología con las metodologías docentes para potenciar el proceso de enseñanza-aprendizaje. Para ello, ha establecido una red de alianzas que le permite tener acceso a las herramientas de software más avanzadas del mundo profesional.



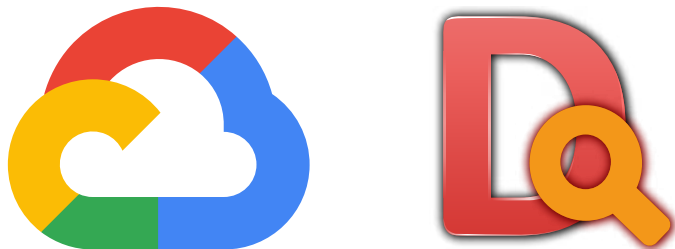
“

Al matricularte recibirás, de forma completamente gratuita, las credenciales de uso académico de las siguientes aplicaciones de software profesional”

TECH ha establecido una red de alianzas profesionales en la que se encuentran los principales proveedores de software aplicado a las diferentes áreas profesionales. Estas alianzas permiten a TECH tener acceso al uso de centenares de aplicaciones informáticas y licencias de software para acercarlas a sus estudiantes.

Las licencias de software para uso académico permitirán a los estudiantes utilizar las aplicaciones informáticas más avanzadas en su área profesional, de modo que podrán conocerlas y aprender su dominio sin tener que incurrir en costes. TECH se hará cargo del procedimiento de contratación para que los alumnos puedan utilizarlas de modo ilimitado durante el tiempo que estén estudiando el programa de Máster Título Propio en Desarrollo en Python, y además lo podrán hacer de forma completamente gratuita.

TECH te dará acceso gratuito al uso de las siguientes aplicaciones de software:



Anaconda

Anaconda es una distribución profesional de Python y R para ciencia de datos, con un precio comercial aproximado de **180 dólares** anuales. Durante el programa universitario en TECH se ofrece **gratis**, permitiendo a los egresados desarrollar, gestionar y escalar entornos de análisis de datos y aprendizaje automático en múltiples sectores.

Esta plataforma capacita al profesional para simplificar la instalación de librerías, gestionar dependencias de proyectos, crear entornos virtuales estables y ejecutar aplicaciones de *machine learning* y análisis predictivo. Además, integra herramientas visuales que optimizan flujos de trabajo y favorecen la reproducibilidad de resultados en investigación y práctica profesional.

Funciones principales:

- ♦ **Gestión de paquetes:** instalación y actualización de librerías de datos y ML
- ♦ **Entornos virtuales:** aislamiento y estabilidad para proyectos complejos
- ♦ **Compatibilidad amplia:** soporte para Python, R y múltiples *frameworks*
- ♦ **Herramientas visuales:** Anaconda Navigator y Jupyter Notebook
- ♦ **Escalabilidad:** integración con plataformas de big data y nube

En conclusión, **Anaconda** impulsa al egresado TECH a dominar la ciencia de datos con eficiencia, precisión y escalabilidad, consolidándose como referencia en entornos profesionales de análisis y desarrollo en Python y R.

Google Career Launchpad

Google Career Launchpad es una solución para desarrollar habilidades digitales en tecnología y análisis de datos. Con un valor estimado de **5.000 dólares**, se incluye de forma **gratuita** en el programa universitario de TECH, brindando acceso a laboratorios interactivos y certificaciones reconocidas en el sector.

Esta plataforma combina capacitación técnica con casos prácticos, usando tecnologías como BigQuery y Google AI. Ofrece entornos simulados para experimentar con datos reales, junto a una red de expertos para orientación personalizada.

Funcionalidades destacadas:

- ♦ **Cursos especializados:** contenido actualizado en cloud computing, machine learning y análisis de datos
- ♦ **Laboratorios en vivo:** prácticas con herramientas reales de Google Cloud sin configuración adicional
- ♦ **Certificaciones integradas:** preparación para exámenes oficiales con validez internacional
- ♦ **Mentorías profesionales:** sesiones con expertos de Google y partners tecnológicos
- ♦ **Proyectos colaborativos:** retos basados en problemas reales de empresas líderes

En conclusión, **Google Career Launchpad** conecta a los usuarios con las últimas tecnologías del mercado, facilitando su inserción en áreas como inteligencia artificial y ciencia de datos con credenciales respaldadas por la industria.

DPy - Designite

DPy – Designite es una herramienta de evaluación de calidad de código para proyectos en Python, con un precio comercial aproximado de **200 euros** anuales. Durante el programa universitario en TECH se ofrece **gratis**, permitiendo a los egresados identificar problemas estructurales y de diseño en entornos de desarrollo ágil y CI/CD.

Esta plataforma capacita al egresado TECH en Informática para detectar code smells arquitectónicos, de diseño e implementación, además de problemas específicos de *Machine Learning*. También ofrece métricas orientadas a objetos, visualizaciones claras y reportes automatizados que mejoran escalabilidad, fiabilidad y mantenibilidad del software en proyectos profesionales de programación en Python.

Funcionalidades destacadas:

- ♦ **Detección de Code Smells:** arquitectónicos, de diseño, de implementación y de *Machine Learning*
- ♦ **Métricas de calidad:** indicadores estructurales orientados a objetos
- ♦ **Visualización avanzada:** reportes gráficos claros e interpretables
- ♦ **Mejora estructural:** soporte para entornos ágiles y CI/CD

En conclusión, **DPy – Designite** impulsa al egresado TECH a mejorar la calidad del software en Python, garantizando soluciones sólidas, escalables y confiables.

07

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

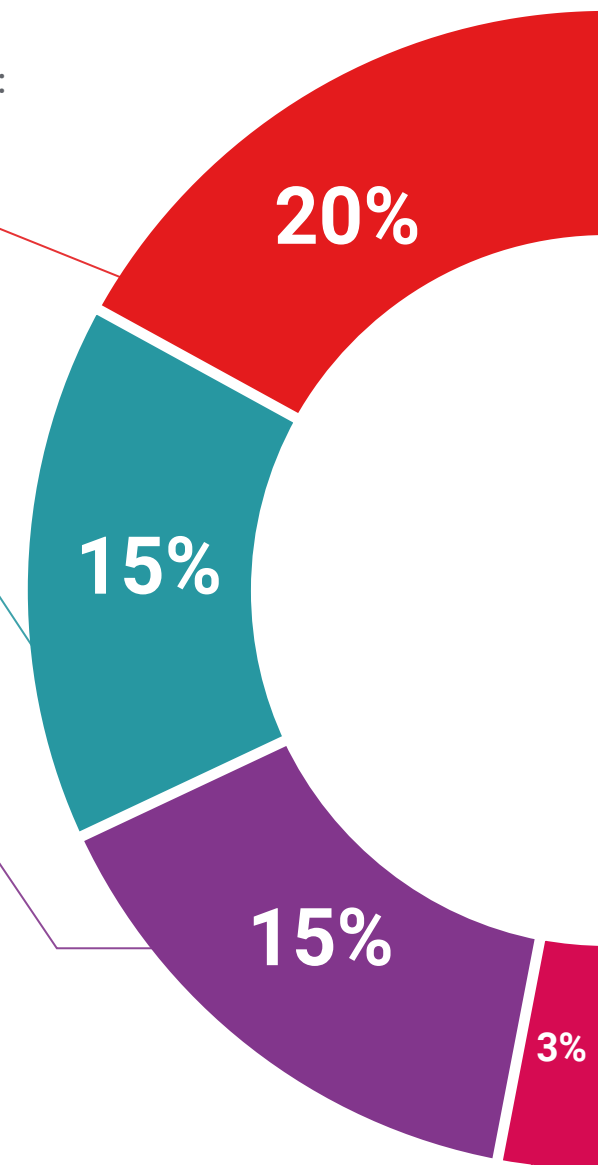
Presentamos los contenidos de manera atractiva y dinámica en píldoras multimedia que incluyen audio, vídeos, imágenes, esquemas y mapas conceptuales con el fin de afianzar el conocimiento.

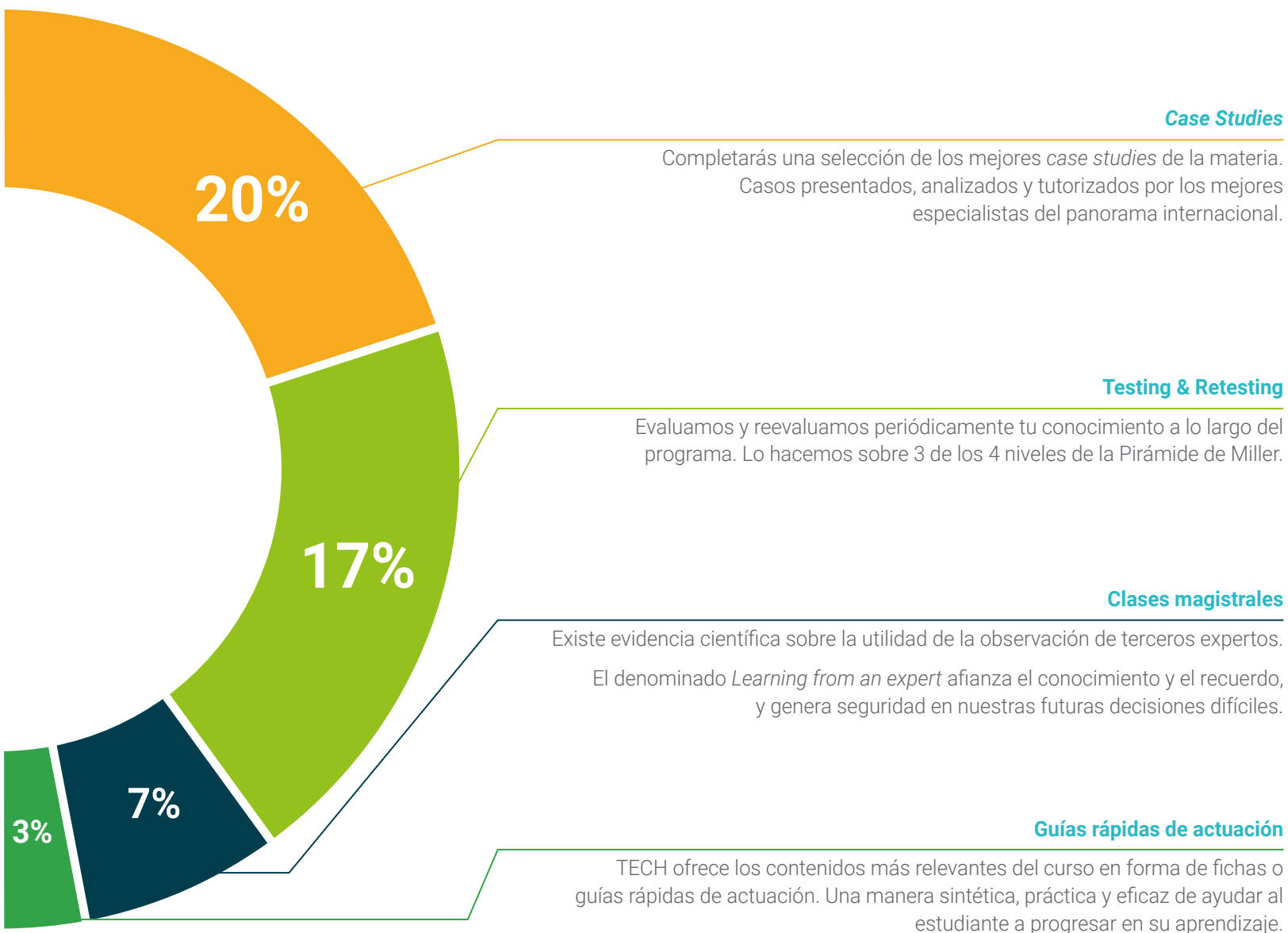
Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Lecturas complementarias

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





Case Studies



Testing & Retesting



Clases magistrales



Guías rápidas de actuación

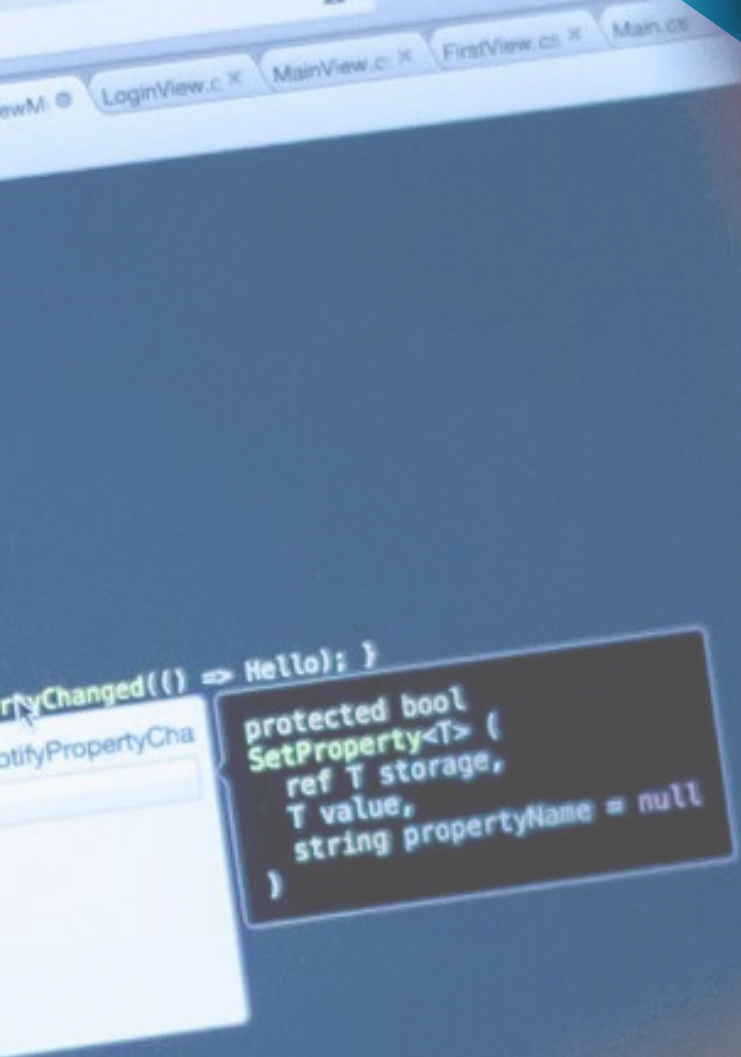


08

Cuadro docente

Los docentes de este Máster Título Propio son expertos en la industria, con una vasta experiencia práctica en el desarrollo de software y un historial comprobado de proyectos exitosos. De hecho, al unir la teoría con la aplicación práctica, estos mentores no solo transmitirán conocimientos avanzados en Python, sino que también compartirán su experiencia, proporcionando a los egresados una visión valiosa sobre las mejores prácticas y tendencias actuales en el sector. Además, su enfoque pedagógico destacará por su compromiso para proporcionar a los profesionales la capacitación técnica más actualizada, para destacar en el dinámico y altamente demandado.





“

*Accederás a un programa universitario
diseñado por auténticas referencias
del Desarrollo en Python”*

Director Invitado Internacional

Nayan Paul es un destacado Arquitecto de Tecnología Principal con una especialización en Modelos de Lenguaje Grande (LLM), Inteligencia Artificial Generativa, Ciencia de Datos y estrategias de habilitación de *Big Data*. Con una amplia trayectoria profesional, ha trabajado en la implementación de soluciones tecnológicas avanzadas para diversas plataformas de datos, enfocándose en la seguridad, la gobernanza y la gestión de datos, además del aprendizaje automático. Igualmente, ha sido ponente en eventos internacionales de renombre, como el *Hadoop Summit – Strata*, en Londres, y el *San Jose Data Summit*, donde ha compartido sus conocimientos sobre las últimas tendencias en tecnologías de datos.

De este modo, ha formado parte integral de *Accenture*, donde ha ocupado diversos roles de liderazgo, incluido el de Director Principal de Ingeniería de Aprendizaje Automático. Además de haber sido certificado en las plataformas de nube de *Amazon* y *Google*, también ha sido reconocido como Campeón de *Databricks*. De hecho, su experiencia en la implementación de estrategias de datos “sin servidor” y nativas de la nube le ha permitido posicionarse como un referente en la industria, contribuyendo a la transformación digital de grandes organizaciones a nivel global.

Asimismo, ha sido autor del libro *The 7 Pillars of Designing Well Architected Solution* y ha obtenido una patente por su trabajo en la creación de “Intelligent Data Foundation (IDF)” para *Accenture*. A su vez, ha desempeñado roles clave como Analista de TI y Consultor en empresas como *Capgemini* y *Tata Consultancy Services*, donde ha liderado equipos en diseño, planificación y desarrollo de aplicaciones innovadoras.

En definitiva, la combinación de su vasta experiencia técnica, su capacidad para liderar equipos internacionales y su enfoque en el avance de la Inteligencia Artificial y el *Big Data* lo ha consolidado como una de las figuras más influyentes en el ámbito de la tecnología a nivel mundial.



D. Paul, Nayan

- Director Principal de Ingeniería de Aprendizaje Automático en Accenture, Nueva York, Estados Unidos
- Director de Ingeniería de Aprendizaje Automático en Accenture
- Gerente en Accenture
- Consultor en Capgemini
- Analista de TI en Tata Consultancy Services
- Diploma en Gestión Empresarial, Administración y Gestión de Empresas por la Fundación ICFAI
- Licenciado en Tecnología e Ingeniería Informática por el Instituto Tecnológico Sikkim Manipal (SMU)

“

Gracias a TECH podrás aprender con los mejores profesionales del mundo”

Dirección



D. Matos Rodríguez, Dionis

- ♦ *Data Engineer* en Wide Agency Sadexo
- ♦ *Data Consultant* en Tokiota
- ♦ *Data Engineer* en Devoteam
- ♦ *BI Developer* en Ibermática
- ♦ *Applications Engineer* en Johnson Controls
- ♦ *Database Developer* en Suncapital España
- ♦ *Senior Web Developer* en Deadlock Solutions
- ♦ *QA Analyst* en Metaconzept
- ♦ Máster en *Big Data & Analytics* por la EAE Business School
- ♦ Máster en Análisis y Diseño de Sistemas
- ♦ Licenciatura en Ingeniería Informática por la Universidad APEC

Profesores

Dña. Gil Contreras, Milagros

- ♦ *Content Creator* en MPCTech LLC
- ♦ Gestora de proyectos
- ♦ *Freelance IT Writer*
- ♦ MBA por la Universidad Complutense de Madrid
- ♦ Licenciada/Graduada en Administración de Empresas por el Instituto Tecnológico de Santo Domingo

Dña. Delgado Feliz, Benedit

- ♦ Asistente Administrativo y Operador De Vigilancia Electrónica en la Dirección Nacional de Control de Drogas
- ♦ Servicio al Cliente en Cáceres y Equipos
- ♦ Reclamaciones y Servicio al Cliente en Express Parcel Services (EPS)
- ♦ Especialista en Microsoft Office por la Escuela Nacional de Informática
- ♦ Comunicadora Social por la Universidad Católica Santo Domingo

**D. Villar Valor, Javier**

- ♦ Director y Socio Fundador de Impulsa2
- ♦ *Chief Operations Officer* (COO) en Summa Insurance Brokers
- ♦ Director de Transformación y Excelencia Operacional en Johnson Controls
- ♦ Máster en *Coaching* Profesional
- ♦ Executive MBA por la Emlyon Business School, Francia
- ♦ Máster en Gestión de la Calidad por EOI
- ♦ Ingeniería Informática por la Universidad Acción Pro-Educación y Cultura (UNAPEC)

D. Gil Contreras, Armando

- ♦ *Lead Big Data Scientist* en Jhonson Controls
- ♦ *Data Scientist-Big Data* en Opensistemas S.A.
- ♦ Auditor de Fondos en Creatividad y Tecnología S.A. (CYTSA)
- ♦ Auditor del Sector Público en PricewaterhouseCoopers Auditores
- ♦ Máster en *Data Science* por el Centro Universitario de Tecnología y Arte
- ♦ Máster MBA en Relaciones y Negocios Internacionales por el Centro de Estudios Financieros (CEF)
- ♦ Licenciatura en Economía por el Instituto Tecnológico de Santo Domingo



*Una experiencia de capacitación
única, clave y decisiva para
impulsar tu desarrollo profesional”*

09

Titulación

El Máster Título Propio en Desarrollo en Python garantiza, además de la capacitación más rigurosa y actualizada, el acceso a un título de Máster Propio expedido por TECH Universidad.



“

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

Este **Máster Título Propio en Desarrollo en Python** contiene el programa universitario más completo y actualizado del mercado.

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

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

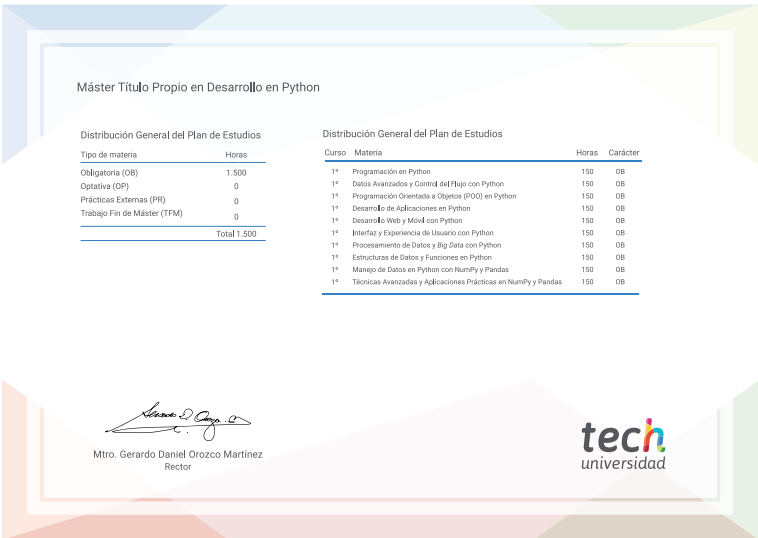
TECH es miembro de **Python Software Foundation (PSF)**, organización dedicada a promover la excelencia en la difusión del lenguaje de programación Python y la aplicación en Deep Learning. Esta afiliación refirma su compromiso con la calidad científica y práctica.



Título: **Máster Título Propio en Desarrollo en Python**

Modalidad: **No escolarizada (100% en línea)**

Duración: **12 meses**



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



Máster Título Propio Desarrollo en Python

- » Modalidad: No escolarizada (100% en línea)
- » Duración: 12 meses
- » Titulación: TECH Universidad
- » Horario: a tu ritmo
- » Exámenes: online

Máster Título Propio

Desarrollo en Python

Aval/Membresía



tech
universidad