

Máster Título Propio

Desarrollo en Python



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

pág. 4

02

Objetivos

pág. 8

03

Competencias

pág. 12

04

Dirección del curso

pág. 16

05

Estructura y contenido

pág. 20

06

Metodología

pág. 30

07

Titulación

pág. 38

01

Presentación

Python destaca como un lenguaje de programación versátil y eficiente. En primer lugar, su sintaxis clara y legible facilita la escritura y comprensión del código, lo que agiliza el proceso de desarrollo y reduce la probabilidad de errores. Además, la amplia variedad de bibliotecas y *Frameworks* disponibles brinda a los desarrolladores acceso a herramientas poderosas y preexistentes, permitiendo un desarrollo más rápido y eficiente. Su naturaleza de código abierto y la activa comunidad de desarrolladores también contribuyen a la constante mejora y actualización del lenguaje, garantizando un entorno de desarrollo dinámico y moderno. Por ello, TECH ha desarrollado un programa exhaustivo, destinado a capacitar a los profesionales con los últimos avances en Desarrollo en Python, con una metodología 100% online.



“

Python es reconocido por su capacidad de adaptación, siendo utilizado en aplicaciones, desde desarrollo web hasta Inteligencia Artificial. ¿A qué esperas para matricularte?”

Python es un lenguaje de alto nivel de Programación, muy utilizado entre los informáticos, pues cuenta con una amplia gama de bibliotecas y marcos de trabajo que simplifican tareas comunes, permitiendo a los desarrolladores enfocarse en la lógica de sus aplicaciones en lugar de perder tiempo en detalles técnicos. Además, su versatilidad es otra de sus ventajas significativas, ya que se puede utilizar en una variedad de entornos, desde desarrollo web hasta análisis de datos y aprendizaje automático.

Así nace este Máster Título Propio en Desarrollo en Python, un programa que abarcará el manejo avanzado de datos y tipos en Python, explorando identificadores, palabras clave, tipos integrales y booleanos, así como el formateo avanzado de *strings* y codificaciones. Adicionalmente, se examinarán colecciones como tuplas, listas y diccionarios, junto con técnicas de iteración y funciones lambda, proporcionando una base sólida en los fundamentos del lenguaje.

Asimismo, se profundizará en el desarrollo de aplicaciones en Python, haciendo énfasis en mejores prácticas y metodologías modernas. Desde la arquitectura de aplicaciones, hasta el despliegue y mantenimiento, se tocarán aspectos como el diseño y modelado avanzado, la gestión de dependencias, los patrones de diseño, *testing* y *debugging*, la optimización de rendimiento las y estrategias de despliegue y distribución.

De igual forma, el informático ahondará en el desarrollo web y móvil con Python, cubriendo *frameworks*, como Django y Flask, así como el desarrollo de APIs y servicios web. Además, se enfocará en el diseño de interfaz y experiencia de usuario (UI/UX), desde el uso de herramientas de diseño, hasta la mejora de la accesibilidad y usabilidad. Finalmente, se abordará el manejo y análisis de datos, utilizando Python y herramientas como NumPy, Pandas y Matplotlib.

Así, esta titulación se presenta como una oportunidad única, mediante una propuesta académica totalmente en línea y adaptable. Con este enfoque, los profesionales disfrutarán de una mayor libertad para gestionar su tiempo de acceso, permitiéndoles armonizar sus compromisos personales y laborales diarios.

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 está concebido recogen una información teórica 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



Una capacitación integral y especializada en el Desarrollo en Python, que te preparará para enfrentar desafíos reales en el mundo del desarrollo de Software”

“

A través de este Máster Título Propio, 100% online, profundizarás en el diseño Frontend y Backend, la gestión de bases de datos y las estrategias de publicación en tiendas de aplicaciones”

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

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

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

Cubrirás, desde los fundamentos como variables y tipos de datos, hasta las técnicas más avanzadas de visualización de datos y estrategias de optimización de rendimiento y almacenamiento. ¡Inscríbete ahora!

Indagarás en la Programación Orientada a Objetos (POO), además de en temas como clases, herencia, polimorfismo, creación de clases abstractas y excepciones personalizadas.



02

Objetivos

Este Máster Título Propio se ha diseñado con el firme propósito de cultivar profesionales altamente competentes y especializados en el vasto universo de la programación en este lenguaje de alto rendimiento. A través de módulos exhaustivos, los egresados obtendrán una inmersión profunda en la sintaxis y funcionalidades de Python, desde el manejo avanzado de datos, hasta la maestría en Programación Orientada a Objetos (POO) y el diseño eficiente de aplicaciones web y móviles. Con un enfoque pedagógico riguroso y actualizado, este programa garantizará una experiencia educativa única que posicionará a los informáticos como expertos codiciados en la industria del desarrollo de Software. Prepárese para superar sus límites y adentrarse en un viaje educativo que no solo eleva sus habilidades en Python, sino que también le brinda las herramientas para destacar y liderar en el dinámico mundo del desarrollo tecnológico.



“

Conviértete en un líder en el campo del desarrollo de Software, dotándolos de conocimientos avanzados y habilidades especializadas para abordar los desafíos contemporáneos de la tecnología”



Objetivos generales

- Proporcionar una comprensión integral de Python
- Capacitar el manejo avanzado de datos y tipos en Python
- Aplicar los principios de la Programación Orientada a Objetos (POO) en Python
- Fomentar el uso de mejores prácticas y metodologías modernas en desarrollo de Software
- Proporcionar una capacitación integral en desarrollo web y móvil con Python
- Integrar principios de UI/UX en el desarrollo de Software
- Capacitar en la configuración y uso de herramientas y entornos de desarrollo para datos
- Profundizar en el uso de estructuras de datos y funciones en Python
- Capacitar en técnicas avanzadas de visualización de datos con Matplotlib
- Capacitar en estrategias de optimización de rendimiento y almacenamiento de datos



Gracias a la amplia biblioteca que ofrece esta titulación, llena con los recursos multimedia más innovadores, obtendrás las herramientas para destacar y liderar en el dinámico mundo del desarrollo tecnológico”





Objetivos específicos

Módulo 1. Programación en Python

- ♦ Habilitar la configuración y uso efectivo del entorno de desarrollo de Python
- ♦ Comprender de conceptos avanzados de Programación

Módulo 2. Datos avanzados y control del flujo con Python

- ♦ Dominar las convenciones y prácticas para el manejo de identificadores y palabras clave
- ♦ Aplicar estructuras de datos complejas y sus operaciones

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

- ♦ Dominar la creación y uso de clases y objetos en Python
- ♦ Aplicar herencia y polimorfismo en Python

Módulo 4. Desarrollo de aplicaciones en Python

- ♦ Especializarse en el diseño y modelado avanzado de aplicaciones
- ♦ Instruirse en la optimización, despliegue y mantenimiento de aplicaciones

Módulo 5. Desarrollo web y móvil con Python

- ♦ Usar *Frameworks* web populares en Python
- ♦ Preparar para el desarrollo y publicación de aplicaciones móviles

Módulo 6. Interfaz y experiencia de usuario con Python

- ♦ Instruir en técnicas de diseño responsivo y adaptativo
- ♦ Preparar para realizar pruebas de usabilidad y análisis de comportamiento del usuario

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

- ♦ Manejar técnicas de control de flujo y funciones para el manejo de datos
- ♦ Promover las mejores prácticas de codificación y manejo de errores en Python

Módulo 8. Estructuras de datos y funciones en Python

- ♦ Crear y usar funciones de manera avanzada
- ♦ Leer y escribir archivos, así como su procesamiento

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

- ♦ Crear y manipular Arrays con NumPy
- ♦ Promover la competencia en visualización de datos con Matplotlib

Módulo 10. Técnicas avanzadas y aplicaciones prácticas en NumPy y Pandas

- ♦ Desarrollar conocimiento especializado en carga y almacenamiento de datos desde y hacia diversas fuentes
- ♦ Instruir en estrategias avanzadas de limpieza y transformación de datos

03

Competencias

El programa no solo impartirá conocimientos técnicos, sino que también se centrará en cultivar competencias que transformarán a los egresados en profesionales destacados, en el ámbito del desarrollo de Software. Así, se abarcará la capacidad de desarrollar soluciones innovadoras y eficientes, mediante un dominio experto de Python, desde el manejo avanzado de datos hasta la creación de aplicaciones web y móviles de alto rendimiento. Además, los informáticos se equiparán con habilidades especializadas en Programación Orientada a Objetos (POO), diseño de interfaces de usuario y experiencia de usuario (UI/UX), y el análisis avanzado de datos con bibliotecas como NumPy y Pandas.



“

Se fomentará el desarrollo de competencias blandas cruciales, como el pensamiento crítico, la resolución de problemas, la colaboración en equipo y la adaptabilidad”



Competencias generales

- ♦ Desarrollar habilidades prácticas en Programación
- ♦ Equiparse con destrezas avanzadas en el control del flujo de programas
- ♦ Manejar el diseño y la implementación de Software Orientado a Objetos
- ♦ Capacitarse en el desarrollo integral de aplicaciones en Python
- ♦ Dominar el diseño y la gestión de aplicaciones web y móviles
- ♦ Controlar el diseño de interfaz y experiencia de usuario usando Python
- ♦ Desarrollar competencias en el manejo y análisis de datos con Python
- ♦ Adquirir habilidades en el manejo avanzado de archivos y modelado en Python
- ♦ Desarrollar habilidades en el manejo avanzado de datos con NumPy y Pandas
- ♦ Profundizar en el manejo avanzado de datos con NumPy y Pandas

“

Enfrentarás desafíos del mundo real y destacarás en un mercado laboral competitivo y en constante evolución. ¡Apuesta por TECH!”





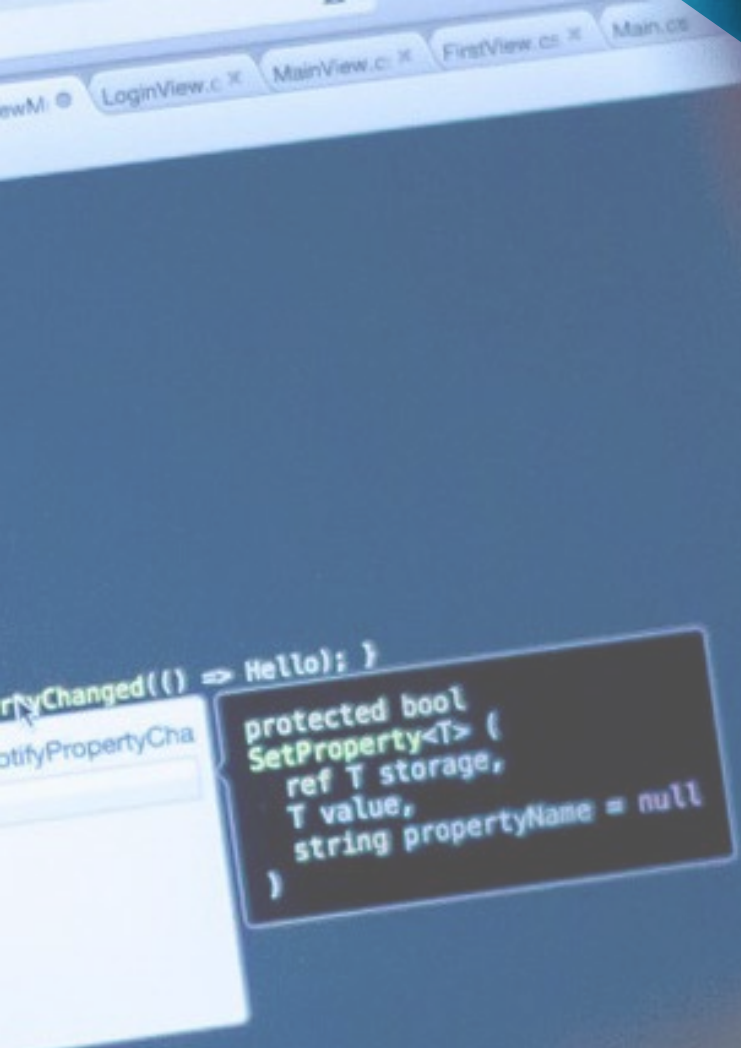
Competencias específicas

- ♦ Capacitarse en el manejo avanzado de datos en Python
- ♦ Manejar el uso avanzado de funciones en Python
- ♦ Implementar conceptos avanzados de Programación Orientada a Objetos (POO) como clases abstractas y excepciones personalizadas
- ♦ Controlar pruebas y *Debugging*
- ♦ Desarrollar APIs y servicios web
- ♦ Dominar el uso de herramientas de diseño UI/UX con Python
- ♦ Usar bibliotecas esenciales para datos en Python
- ♦ Aplicar diferentes estructuras de datos de forma práctica
- ♦ Usar Pandas para el manejo de datos estructurados
- ♦ Promover competencias en análisis y manipulación de series temporales y datos complejos

04 Dirección del curso

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. Al unir la teoría con la aplicación práctica, estos mentores no solo transmitirán conocimientos avanzados en Python, sino que también compartirán su experiencia del mundo real, proporcionando a los estudiantes 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 egresados la capacitación técnica más actualizada, para destacar en el dinámico campo del desarrollo de Software.





“

Triunfarás de la mano de los mejores y adquirirás los conocimientos y competencias que necesitas para embarcarte en el sector del Desarrollo en Python”

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

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

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



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

05

Estructura y contenido

El temario ha sido meticulosamente diseñado para ofrecer una experiencia integral para los egresados. Desde la inmersión profunda en la sintaxis y funcionalidades avanzadas de Python, hasta la especialización en Programación Orientada a Objetos (POO), diseño de aplicaciones web y móviles, y el manejo experto de datos con bibliotecas como NumPy y Pandas, cada módulo estará cuidadosamente estructurado para proporcionar conocimientos sólidos y habilidades prácticas. Además, se indagará temas cruciales, como el diseño de interfaces de usuario y experiencia de usuario (UI/UX), el manejo avanzado de datos y la optimización de rendimiento y almacenamiento.

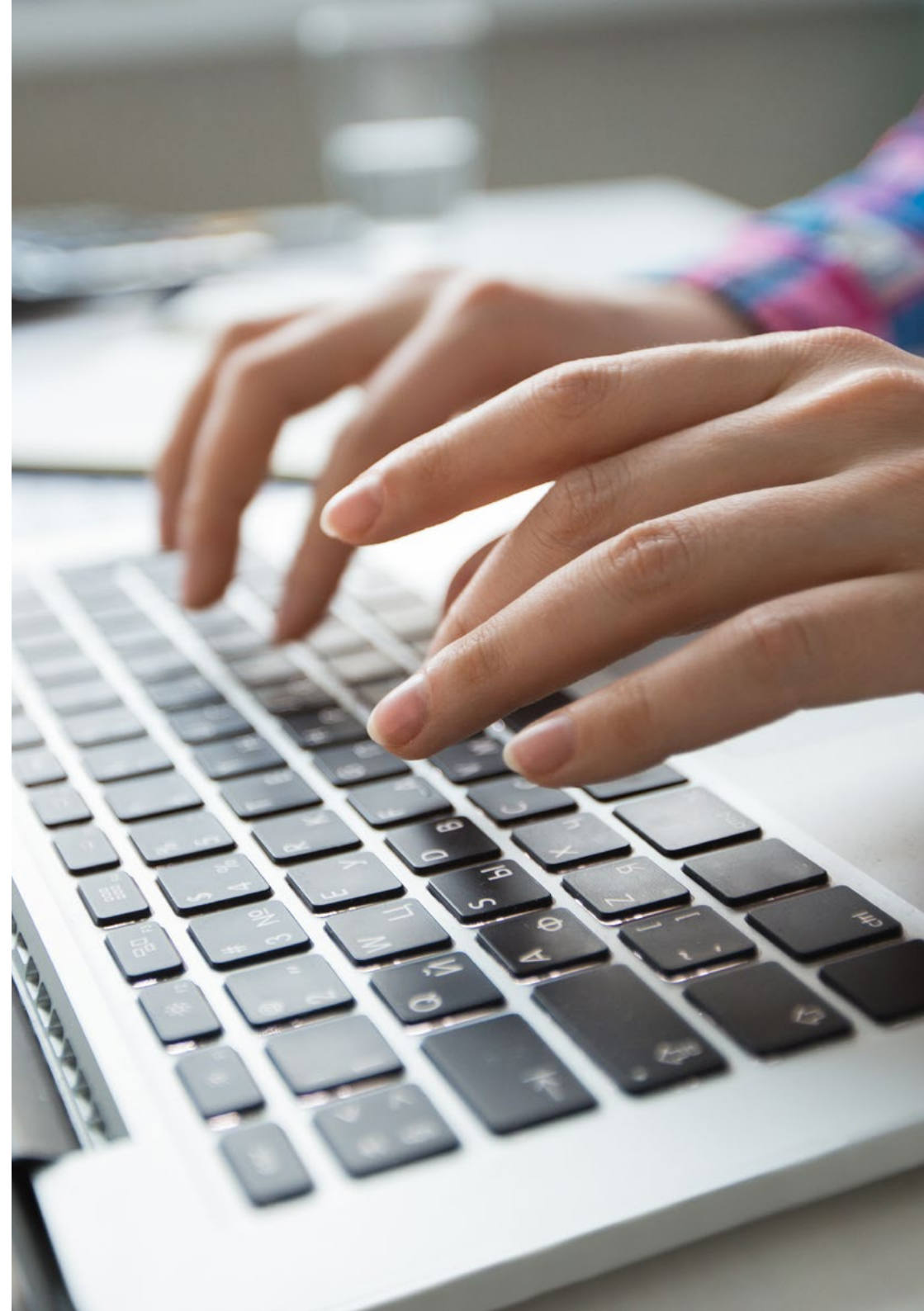


“

No solo adquirirás un dominio completo en Python, sino que también te prepararás para enfrentar con confianza los desafíos dinámicos y de la industria tecnológica”

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 *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

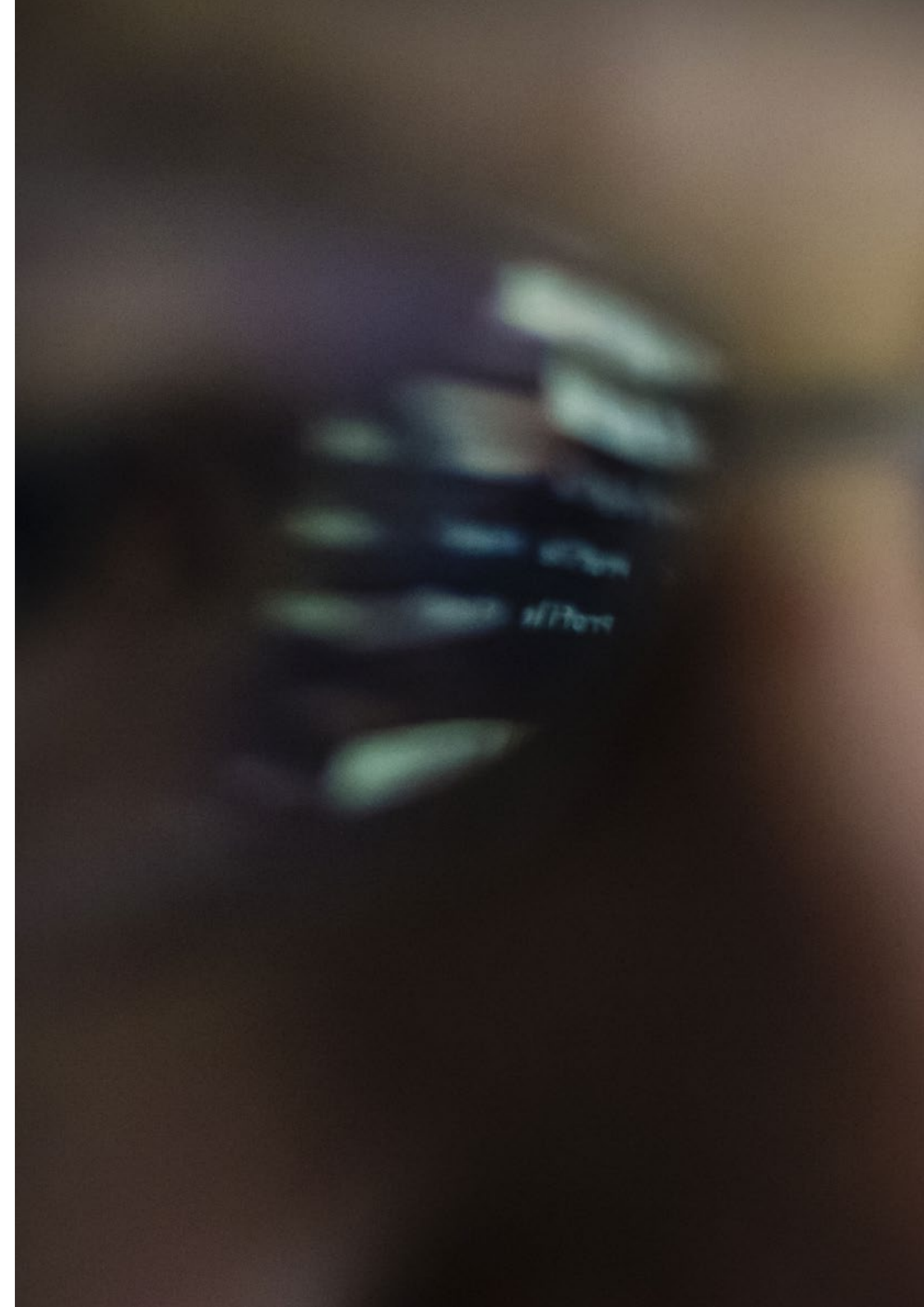
Módulo 6. Interfaz y experiencia de usuario con Python

- 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



Este innovador programa se adaptará a ti para que te mantengas al día con las últimas tendencias y tecnologías, garantizándote estar a la vanguardia de la innovación en el desarrollo de Software”

06

Metodología

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

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



“

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

06

Metodología de estudio

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

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



“

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

El alumno: la prioridad de todos los programas de TECH

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

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

“

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



Los planes de estudios más exhaustivos a nivel internacional

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

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

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

“

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

Case studies o Método del caso

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

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

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



Método Relearning

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

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

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

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



Un Campus Virtual 100% online con los mejores recursos didácticos

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

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

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

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



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

La eficacia del método se justifica con cuatro logros fundamentales:

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

La metodología universitaria mejor valorada por sus alumnos

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

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

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

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



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



Material de estudio

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

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



Prácticas de habilidades y competencias

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



Resúmenes interactivos

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

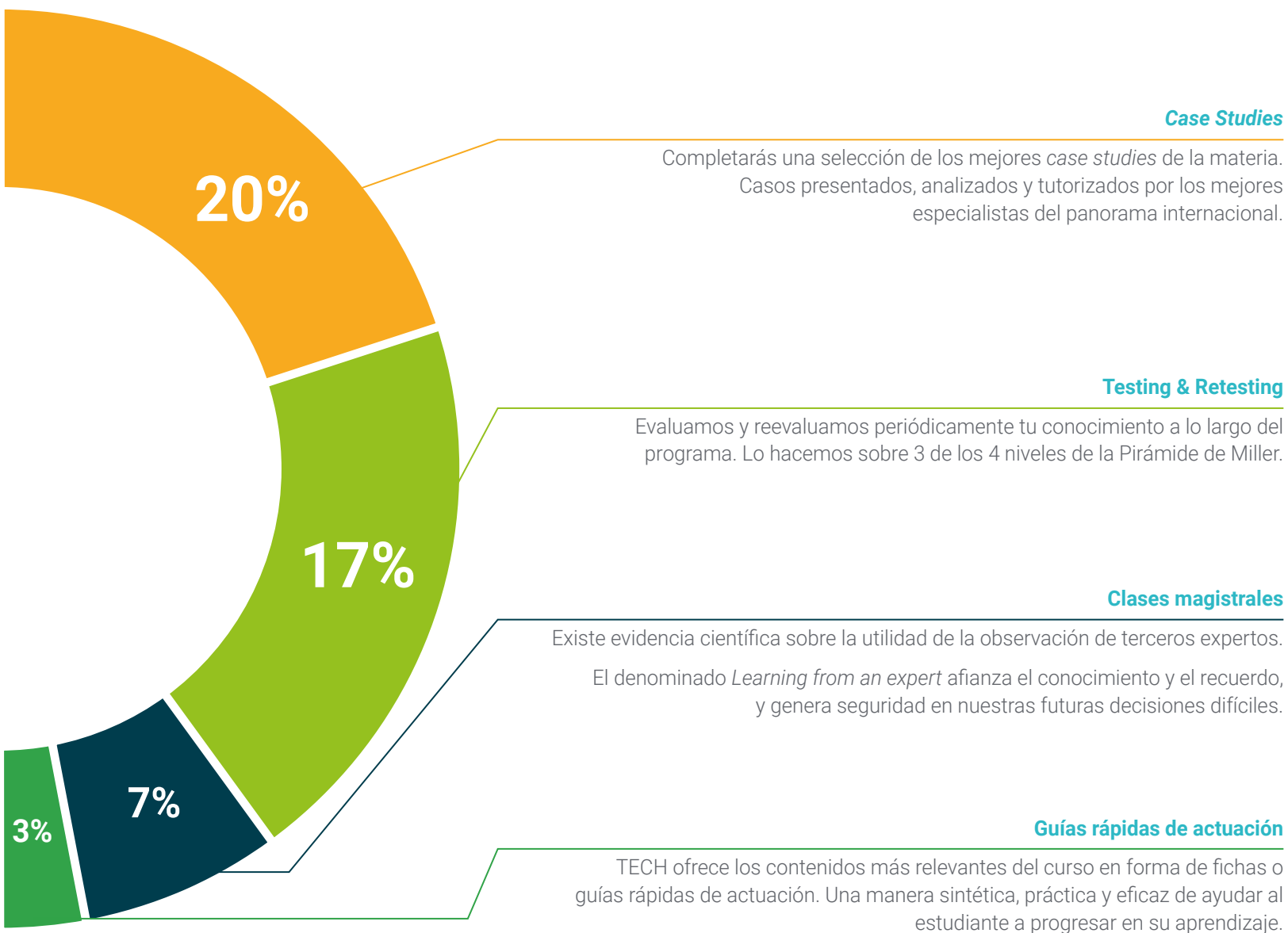
Este sistema exclusivo educativo para la presentación de contenidos multimedia fue premiado por Microsoft como "Caso de éxito en Europa".



Lecturas complementarias

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





07

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.

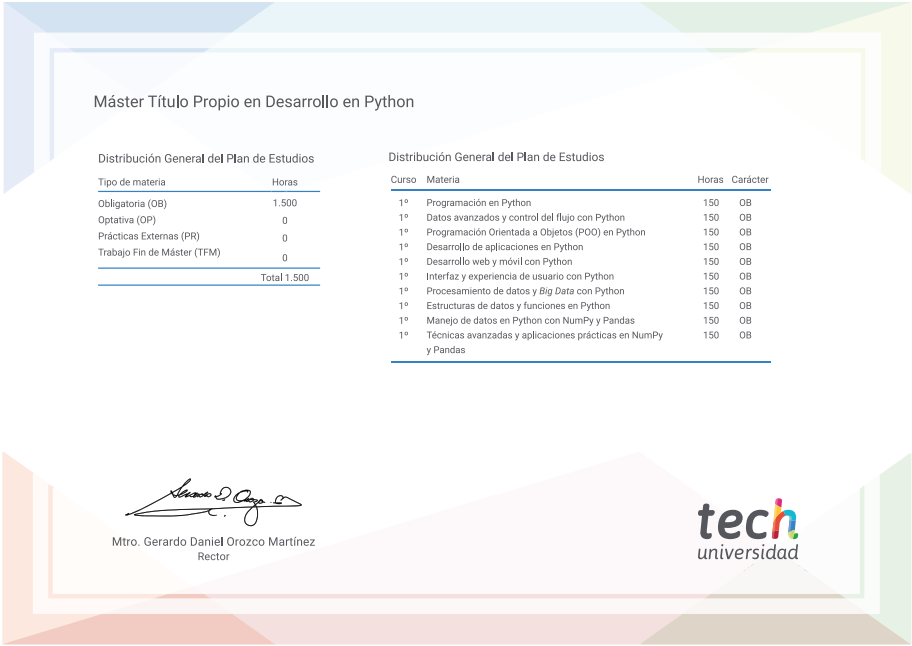
Tras la superación de la evaluación, el alumno recibirá por correo postal* con acuse de recibo su correspondiente título de **Máster Propio** emitido por **TECH Universidad**.

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.

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