



PYTHON

Modalidad: semipresencial

Duración: 40 horas

Inversión: RD\$20,000.00

Descripción del programa

Este diplomado de 40 horas ofrece una formación integral en el desarrollo de software con Python, abarcando desde la programación orientada a objetos hasta el consumo de APIs. A lo largo de 13 semanas, los participantes explorarán diversas áreas clave, comenzando con los fundamentos de la Programación Orientada a Objetos y el manejo de librerías, para luego avanzar hacia el uso de excepciones, manejo de archivos, y programación funcional.

Se incluirá un enfoque práctico en el uso de expresiones regulares, además de introducir a los estudiantes en el manejo de bases de datos y el uso del ORM (Object Relational Mapper) SQLAlchemy. La creación de APIs con FastAPI será otro componente esencial del curso, culminando con la integración de estas en el frontend y el consumo de servicios externos.

Este diplomado está diseñado para desarrollar competencias técnicas sólidas y prácticas que permitan a los participantes enfrentar desafíos reales en el desarrollo de software moderno.

Objetivo general

Se persigue capacitar a los participantes en el desarrollo de aplicaciones avanzadas con Python y sus tecnologías asociadas, proporcionando las habilidades necesarias para implementar soluciones efectivas en programación orientada a objetos, manejo de archivos, y creación y consumo de APIs.

PROGRAMA ACADÉMICO

Competencias generales

1. Desarrollar aplicaciones avanzadas con Python, aplicando metodologías y lenguajes modernos para crear soluciones informáticas de alta calidad, que respondan de manera efectiva a las demandas del mercado.
2. Garantizar la calidad de software mediante el uso de técnicas de validación y pruebas adecuadas, asegurando que las soluciones desarrolladas cumplan con los requisitos establecidos y asegurando un funcionamiento confiable.
3. Dominar la Programación Orientada a Objetos (OOP) en Python, aprendiendo a crear clases, definir métodos, aplicar principios como la herencia, polimorfismo y encapsulación, y aprovechar técnicas avanzadas como los métodos mágicos y la sobrecarga de operadores.
4. Gestionar eficientemente archivos y datos en Python, utilizando técnicas avanzadas de lectura y escritura en formatos como CSV, JSON y XML, así como el manejo de excepciones y la serialización de objetos con librerías como pickle.
5. Crear y consumir APIs modernas con FastAPI, aprendiendo a desarrollar APIs RESTful, gestionar autenticación con JWT, y consumir servicios externos de forma eficiente, integrando todo con el frontend para ofrecer soluciones completas y escalables.

Módulo 1. Principios Avanzados de Programación y Modularización en Python (Semanas 1, 2, 3, 4 y 5)

Tema 1.1. Programación Orientada a Objetos (OOP) en Python

- 1.1.1 Tipos de objetos como clases en Python.
- 1.1.2 Proceso para la definición de clases en un enfoque OOP.
- 1.1.3 Métodos.
- 1.1.4 La función isinstance y la clase object.
- 1.1.5 Abstracción, encapsulación, polimorfismo y herencia en Python.
- 1.1.6 Métodos mágicos y sobrecarga de operadores (Ej. __init__, __str__, __repr__, __add__).
- 1.1.7 Decoradores de clases y métodos estáticos (@classmethod, @staticmethod).

Tema 1.2. Librerías (módulos) en Python

- 1.2.1 Definición de módulo de un programa.
- 1.2.2 Importación de módulos.
- 1.2.3 Módulos definidos por el programador.
- 1.2.4 Librerías comunes en Python (random, os, etc.).
- 1.2.5 Manejo de fechas con la librería datetime.
- 1.2.6 Manejo de entornos virtuales y gestión de paquetes con pip.
- 1.2.7 Creación de un paquete Python e instalación local (Distribuir tu propio paquete con setup.py).

PROGRAMA ACADÉMICO

Tema 1.3. Manejo de Excepciones en Python

- 1.3.1 Uso del bloque try-except-finally.
- 1.3.2 Excepciones predefinidas y comunes en Python.
- 1.3.3 Levantar excepciones con raise.
- 1.3.4 Excepciones personalizadas.
- 1.3.5 Uso de else y finally en excepciones.

Módulo 2. Manejo de Archivos, Programación Funcional y Expresiones Regulares (Semanas 6, 7, 8 y 9)

Tema 2.1. Manejo de archivos en Python

- 2.1.1 Técnicas de escritura en archivos.
- 2.1.2 Técnicas de lectura de archivos.
- 2.1.3 Modo de apertura de archivos.
- 2.1.4 Uso de with para manejar archivos.
- 2.1.5 Manejo de excepciones en operaciones con archivos.
- 2.1.6 Escritura y lectura de archivos CSV.
- 2.1.7 Manejo de archivos JSON y XML.
- 2.1.8 Serialización y deserialización de objetos con pickle y JSON.

Tema 2.2. Programación Funcional en Python

- 2.2.1 Funciones anónimas (lambda).
- 2.2.2 Funciones de orden superior (map(), filter(), reduce()).
- 2.2.3 Funciones generadoras y uso de yield.
- 2.2.4 List comprehensions, set comprehensions, y dict comprehensions.
- 2.2.5 Decoradores en funciones (creación y uso).

Tema 2.3. Expresiones Regulares

- 2.3.1 Introducción a las expresiones regulares con re.
- 2.3.2 Patrones comunes y su uso en Python.
- 2.3.3 Extracción y sustitución de texto.
- 2.3.4 Validación de datos usando expresiones regulares.

Módulo 3. Bases de Datos y Creación/Consumo de APIs con FastAPI (Semanas 10, 11, 12 y 13)

Tema 3.1. Manejo de Bases de Datos en Python

- 3.1.1 Conexión a bases de datos con sqlite3.
- 3.1.2 Consultas y operaciones básicas de bases de datos.
- 3.1.3 ORM en Python: Introducción a SQLAlchemy.
- 3.1.4 Manejo de transacciones y conexiones múltiples.
- 3.1.5 Consultas avanzadas y buenas prácticas.

Tema 3.2. Introducción a FastAPI para la creación de APIs

- 3.2.1 Instalación y configuración de FastAPI.
- 3.2.3 Creación de rutas básicas (GET, POST, PUT, DELETE).
- 3.2.4 Uso de dependencias en FastAPI.
- 3.2.4 Autenticación y autorización con OAuth2 y JWT.

Tema 3.3. Consumo de APIs con FastAPI

- 3.3.1 Consumir APIs desde Python con requests y httpx.
- 3.3.2 Uso de la librería openai para consumir el API de OpenAI (ChatGPT).
- 3.3.3 Integración de FastAPI con frontend (ej. Angular Flask).
- 3.3.4 CORS y manejo de solicitudes desde diferentes dominios.

Conocimiento deseable para tomar el curso

- Conocimiento deseable para tomar el curso
- Conceptos básicos de programación
- Conceptos básicos de Python
- Manejo de Texto en Python
- Condicionales en Python
- Iteraciones en Python
- Funciones en Python