

# Analítico: Desarrollo Web

Este curso tiene como finalidad profundizar en herramientas básicas para desarrollar aplicaciones web con bases de datos, y adquirir conocimientos que te permitan seguir construyendo tu perfil profesional para el sector Software y Servicios Informáticos.

## Objetivos:

### Dominio de Herramientas Básicas

- Adquirir habilidades avanzadas en el uso de herramientas esenciales para el desarrollo web, como editores de código, sistemas de control de versiones, y entornos de desarrollo integrado (IDEs).

### Dominio de Python:

- Adquirir habilidades avanzadas en programación con Python, incluyendo conceptos como estructuras de control, funciones y programación orientada a objetos.

### Introducción a Bases de Datos y SQL:

- Comprender los fundamentos de las bases de datos y su importancia en el desarrollo de aplicaciones.
- Aprender el lenguaje SQL para realizar consultas, modificar y gestionar bases de datos relacionales.

### Desarrollo de Aplicaciones Web:

- Aprender a desarrollar aplicaciones web completas, abordando tanto el frontend como el backend.

### Desarrollo de Aplicaciones con Django:

- Familiarizarse con el framework Django y su arquitectura MVC (Modelo-Vista-Controlador).
- Desarrollar habilidades en la creación de aplicaciones web dinámicas y escalables con Django.

## Contenidos del Curso:

### Introducción a la Programación:

- Conceptos básicos de lógica de programación y algoritmos.
- Instalación de Python y herramientas necesarias.

### Programación en Python:

- Sintaxis básica, estructuras de control, y estructuras de datos.
- Programación funcional y orientación a objetos (POO).
- Introducción a Tkinter.

### Bases de Datos:

- Fundamentos de bases de datos, diseño y normalización.
- Introducción y uso de SQL, incluyendo operaciones CRUD con SQLAlchemy.

### Desarrollo Web con Django:

- Instalación y configuración de Django.
- Desarrollo de modelos, vistas, y URLs.
- Implementación de formularios, autenticación y autorización.
- Uso de plantillas y desarrollo del frontend.

## Clases:

### Clase 1

- Presentación del curso y objetivos
- Introducción a la programación: conceptos básicos
  - ¿Qué es la lógica de programación?
  - Ejemplos de problemas lógicos y como abordarlos.
  - Algoritmos. Definición. Características de un buen algoritmo.
  - Ejemplos prácticos de algoritmos simples.
- Instalación de Python y herramientas necesarias (IDEs).

### Clase 2

- Sintaxis básica: variables, tipos de datos, operadores
- Pseudocódigo: Estructura y práctica de ejemplo mediante Pseint.

### Clase 3

- Pseudocódigo
  - Estructuras de control: condicionales (if, else, elif)
  - Estructuras de control: bucles (for, while)

### Clase 4

- Primeros pasos con Python: sintaxis básica, variables y tipos de datos
- PEP8.
- Integración de Conceptos.

### Clase 5

- Listas, tuplas, diccionarios y conjuntos.
- Métodos y operaciones comunes con estructuras de datos.

### Clase 6

- Integración con Python
  - Estructura de Datos.
  - Estructuras de control: condicionales (if, else, elif)
  - Estructuras de control: bucles (for, while)
- Practica.

### Clase 7

- Introducción a la programación funcional.
- Definición y uso de funciones.
- Parámetros y retorno de valores.
- Alcance de variables.

### Clase 8

- Módulos.
  - Importación de módulos estándar y módulos personalizados.
- Tkinter.
- Practica.

### Clase 9

- Conceptos básicos de POO: clases, objetos, instancia.
- Atributos y métodos.
- Principios SOLID.

### Clase 10

- Herencia y polimorfismo.
- Encapsulamiento.
- Métodos mágicos (dunder methods).
- Ejercicios prácticos de POO.

### Clase 11

- Introducción a bases de datos.
  - Explicación de BD y la importancia en el desarrollo de aplicaciones.
- Introducción a SQL.
  - Diferencias entre BD relacionales y NoSQL).
  - Breve explicación de Mysql, PostgreSQL y SQLite3.
- Modelos de datos.
  - Entidades.
  - Atributos.
  - Relaciones.
  - Modelo Entidad Relación.

### Clase 12

- Diseño de Bases de Datos.
  - Normalización y desnormalización.
  - Diagramas entidad-relación (ERD).

### Clase 13

- MER a Tablas.
- SQL Básico.
  - Sentencias SELECT.
  - Filtros, WHERE, ORDER BY.
  - Joins (INNER JOIN, LEFT JOIN, RIGHT JOIN).
- SQL Avanzado.
  - Subconsultas.
  - Funciones agregadas (SUM, COUNT, AVG, etc.).
  - Transacciones y control de concurrencia.

### Clase 14

- Integración de SQL con Python
  - Conexión a bases de datos usando sqlite3 y psycopg2.
  - Ejecución de consultas desde Python.
  - Práctica y ejercicios.

### Clase 15

- ORM con SQLAlchemy
  - Introducción a SQLAlchemy
  - Definición de modelos y relaciones
  - Operaciones CRUD

### Clase 16

- Proyectos con Bases de Datos
  - Desarrollo de un proyecto simple utilizando SQL y Python
  - Práctica y ejercicios

### Clase 17

- Introducción a Django
  - ¿Qué es Django?.
  - Instalación y configuración del entorno.
  - Estructura de un proyecto Django.
- Introducción al Proyecto Final.
  - Definición del proyecto.
  - Planificación y diseño.

### Clase 18

- Metodologías Ágiles.

### Clase 19

- Modelos y Bases de Datos en Django
  - Definición de modelos
  - Migraciones
  - Administración de Django

### Clase 20

- Implementación del Proyecto Final
  - Crear el modelo entidad-relación (ERD) de la base de datos.
  - Diseñar la arquitectura de la aplicación.
  - Desarrollo de la base de datos y modelos.
  - Definir modelos y realizar migraciones iniciales.

### Clase 21

- Vistas y URLs en Django
  - Definición de vistas.
  - Manejo de URLs y rutas.

### Clase 22

- Continuación del Proyecto Final.
  - Desarrollo de vistas y rutas.
  - Implementar vistas y URLs para las funcionalidades básicas (registro, inicio de sesión, creación de eventos).

### Clase 23

- Formularios y Validación
  - Creación y manejo de formularios
  - Validación de datos

### Clase 24

- Autenticación y Autorización
  - Sistema de usuarios de Django
  - Manejo de sesiones y autenticación
  - Permisos y grupos

### Clase 25

- Continuación del Proyecto Final
  - Implementación de formularios y validación.
  - Crear formularios y validaciones para la creación y gestión de eventos.
  - Implementación del Sistema de autenticación.

### Clase 26

- Plantillas en Django
  - Sistema de plantillas de Django
  - Herencia de plantillas
  - Uso de contextos

### Clase 27

- Desarrollo del Proyecto Final
  - Implementación de plantillas y diseño de interfaz.
- Desarrollo del Frontend
  - Crear plantillas HTML y aplicar CSS para las páginas principales (inicio, registro, creación de eventos)

### Clase 28

- Desarrollo del Frontend
  - Implementar interacciones básicas con JavaScript (opcional).
  - Integrar formularios y datos dinámicos en las plantillas.
  - Añadir funcionalidades adicionales en el backend.
  - Mejorar las plantillas y la interfaz de usuario.

### Clase 29

- Preparación para el despliegue.
- Configurar y desplegar la aplicación en <https://www.pythonanywhere.com>

### Clase 30

- Clase de consulta.

### Clase 31

- Muestra de Trabajos Finales

### Clase 32

- Repaso y consolidación de lo aprendido durante el proyecto.

## Criterios de Aprobación:

### Cuestionarios Semanales:

Presentado al final de cada módulo semanal, los estudiantes deben completar un cuestionario, dichos cuestionarios estarán habilitados a partir del día viernes a las 12:00 hs y cerrarán el viernes siguiente a la misma hora cada semana.

Durante las 16 semanas del curso, habrá un total de 16 cuestionarios semanales. Además de lo mencionado anteriormente, es importante destacar que los estudiantes deben aprobar al menos el 80% de los cuestionarios semanales para cumplir con los requisitos del curso. En caso de que un estudiante no apruebe el cuestionario de la semana en curso, habrá la oportunidad de realizar un recuperatorio la semana siguiente. Esto brinda a los estudiantes una segunda oportunidad para alcanzar el nivel de aprobación requerido.

### Asistencia y Participación en clases:

La asistencia regular y la participación activa en clase son fundamentales para comprender el material del curso y tener un buen desempeño en él. Si no puedes asistir a una clase, envía un correo electrónico al instructor con anticipación.

### Proyecto Final:

Es importante informarles que antes de finalizar el curso, habrá un proyecto final que deberán completar y aprobar para obtener la aprobación final. Este proyecto final servirá como una culminación de los conocimientos adquiridos a lo largo del programa y les brindará la oportunidad de aplicar de manera integral lo que han aprendido en todas las áreas temáticas. La aprobación del proyecto final es un requisito esencial para obtener la aprobación del curso en su totalidad.

Para aprobar el proyecto final se tendrán en cuenta los siguientes items:

- **Funcionalidad:** El trabajo final debe cumplir con los requisitos funcionales establecidos en el enunciado del proyecto. Es decir, el proyecto debe ser capaz de hacer lo que se espera que haga, según la especificación.
- **Diseño:** El proyecto debe tener un diseño coherente y atractivo. Esto incluye el diseño de la interfaz de usuario, la presentación de la información y la usabilidad del sistema.
- **Estructura del código:** El código debe estar organizado de manera clara y estructurada, de modo que sea fácil de entender y mantener. Esto incluye la utilización de buenas prácticas de programación, como la modularidad y la reutilización del código.
- **Calidad del código:** El código debe cumplir con estándares de calidad y buenas prácticas de programación, como la legibilidad, la eficiencia, la seguridad y la escalabilidad.