Backend con Python (Django)

Expo No. 2 - Grupo 2

Alumno: Aldo Yax

Fecha de entrega: 30 de agosto de 2025

# Introducción

El backend es la parte de una aplicación que se ejecuta en el servidor y se encarga de procesar la lógica de negocio, conectarse con la base de datos y enviar las respuestas adecuadas al cliente. En Python, los frameworks más usados para el desarrollo de aplicaciones web son Django y Flask. En este documento se abordará el uso de Django.

# Objetivos

## Objetivo General

Explicar los fundamentos del desarrollo Backend con Python utilizando Django.

## Objetivos Específicos

1. Comprender el funcionamiento del ORM en Django.

2. Analizar el papel de las vistas, URLs y templates en una aplicación web.

3. Mostrar un ejemplo práctico de implementación de un proyecto Django.

# Backend

## 1. Python y Django

Django es un framework de alto nivel que permite crear aplicaciones web rápidas, seguras y mantenibles. Incluye un ORM, un sistema de rutas, vistas, plantillas y herramientas de seguridad integradas.

## 2. Modelos ORM

El ORM (Object Relational Mapping) permite trabajar con la base de datos mediante clases y objetos en lugar de escribir directamente sentencias SQL. Ejemplo de modelo en Django:

From django.db import models  
class Estudiante(models.Model):  
 nombre = models.CharField(max\_length=100)  
 edad = models.IntegerField()  
 correo = models.EmailField(unique=True)  
  
 def \_\_str\_\_(self):  
 return self.nombre

## 3. Vistas

Las vistas son funciones o clases que reciben solicitudes HTTP y devuelven respuestas. Ejemplo en Django:

from django.http import HttpResponse  
  
def saludo(request):  
 return HttpResponse('Hola desde Django Backend')

## 4. URLs

Las rutas en Django se definen en el archivo urls.py para asociar URLs con vistas. Ejemplo:

from django.urls import path  
from . import views  
  
urlpatterns = [  
 path('saludo/', views.saludo),  
]

## 5. Templates

Django utiliza un sistema de plantillas para generar páginas dinámicas en HTML. Ejemplo:

<h1>Bienvenido {{ nombre }}</h1>

# Conclusiones

El desarrollo backend con Python y Django permite estructurar aplicaciones de forma ordenada y eficiente. Django ofrece una solución completa con ORM, vistas, rutas y plantillas, lo que facilita la creación de aplicaciones escalables. El uso de modelos ORM simplifica el acceso a la base de datos y el manejo de la lógica de negocio.

Recomendaciones

Bibliografía

# Cuestionario

1. ¿Qué diferencia hay entre Django y Flask?

2. ¿Qué es un ORM y para qué sirve?

3. ¿Cómo se definen las rutas en Django?

4. ¿Qué ventaja tienen las vistas en un backend?

5. ¿Cómo se integran los templates en un proyecto de Django?