



DESARROLLO WEB

**PYTHON Y FRAMEWORKS BACKEND**

## PYTHON Y FRAMEWORKS BACKEND: FLASK Y DJANGO

Python se ha establecido como uno de los lenguajes más versátiles y populares para el desarrollo web backend, ofreciendo dos frameworks principales que atienden diferentes necesidades y filosofías de desarrollo: Flask y Django. Flask adopta un enfoque minimalista y flexible, proporcionando solo los componentes esenciales necesarios para construir aplicaciones web, mientras que Django sigue la filosofía de "batteries included", ofreciendo una suite completa de herramientas y funcionalidades integradas. Esta diversidad de opciones permite a los desarrolladores elegir la herramienta más adecuada, según los requisitos específicos de cada proyecto.

Flask se caracteriza por su simplicidad y flexibilidad, permitiendo a los desarrolladores construir aplicaciones web desde cero con un control total sobre cada componente. Por otro lado, Django ofrece un framework más robusto y estructurado que incluye un ORM integrado, sistema de autenticación, panel de administración automático y muchas otras características avanzadas. Contreras Castañeda (2016), enfatiza que la selección de frameworks backend, debe considerar factores como la complejidad del proyecto, los requisitos de escalabilidad y la experiencia del equipo de desarrollo, aspectos donde tanto Flask como Django ofrecen ventajas distintivas, según el contexto de aplicación.

En el campo de la ingeniería de software, Flask se utiliza frecuentemente para desarrollar microservicios y APIs ligeras, mientras que Django es preferido para aplicaciones web complejas que requieren múltiples funcionalidades integradas. Por ejemplo, Instagram utiliza Django para manejar su backend, aprovechando su escalabilidad y las herramientas integradas para gestión de usuarios, mientras que Pinterest emplea Flask para ciertos microservicios específicos, que requieren mayor flexibilidad y personalización en su implementación.

**Ejercicio práctico.** Crear una aplicación web básica con Flask que maneje formularios.

**Paso 1.** Instalar Flask con `pip install flask`.

**Paso 2.** Crear archivo `app.py`: `from flask import Flask, render_template, request; app = Flask(__name__); @app.route('/') def home(): return '<form method="POST" action="/procesar"><input name="nombre" placeholder="Nombre"><button type="submit">Enviar</button></form>'; @app.route('/procesar', methods=['POST']) def procesar(): nombre = request.form['nombre']; return f'Hola {nombre}!'; if __name__ == '__main__': app.run(debug=True).`

**Paso 3.** Ejecutar con `python app.py` y acceder a `localhost:5000`.

**Resultado.** Se obtiene una aplicación web funcional que procesa formularios, demostrando la simplicidad de Flask para crear aplicaciones web interactivas.