🐍 Mini–Tutorial de Flask

Introducción, rutas, GET/POST y Jinja2

---

1. ¿Qué es Flask?

Flask es un *microframework* de Python para crear aplicaciones web de forma muy sencilla.
Su filosofía es: *"lo mínimo para empezar"*. Flask te da:

- Un **servidor web de pruebas**
- Un sistema de **rutas**
- Manejo de **métodos HTTP** (GET, POST…)
- Integración con **plantillas Jinja2**

No incluye cosas pesadas (autenticación, ORM, etc.) pero se puede ampliar.

2. Instalación y primer "Hola mundo"

pip install flask

Estructura mínima:

```
mi_app/
 ├─ app.py
 └─ templates/
    └─ index.html
```

app.py:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def inicio():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(debug=True)
```

templates/index.html:

```
<!DOCTYPE html>
<html>
<head><title>Hola Flask</title></head>
<body>
```

```
    <h1>Hola Flask</h1>
</body>
</html>
```

Ejecuta:

python app.py

3. ¿Qué es una ruta en Flask?

Una **ruta** es la dirección a la que un usuario puede acceder en tu aplicación.

```
@app.route("/saludo")
def saludo():
    return "Hola desde /saludo"
```

Cada ruta se asocia a una función que devuelve:

- HTML
- Texto
- JSON
- Una plantilla

4. Métodos HTTP: GET y POST

GET

Se usa para pedir información. Ejemplo básico:

```
@app.route("/info")
def info():
    return "Esto es una petición GET"
```

POST

Se usa para **enviar datos** (formularios, etc.).

Ejemplo:

```
from flask import request

@app.route("/procesar", methods=["POST"])
def procesar():
    dato = request.form["nombre"]
    return f"Has enviado: {dato}"
```

5. Ejemplo central del tutorial: Mini aplicación de notas

Objetivo:

- Mostrar una lista de notas
- Añadir una nota mediante formulario
- Usar rutas GET/POST
- Usar Jinja2 para renderizar la página

6. Estructura del proyecto

```
notas/
├── app.py
├── templates/
│   ├── base.html
│   └── notas.html
└── static/
    └── estilos.css
```

7. Plantilla base con Jinja2

templates/base.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Mini Notas</title>
   <link rel="stylesheet" href="/static/estilos.css">
</head>
<body>
   <header><h1>Mini App de Notas</h1></header>

   <main>
      {% block contenido %}{% endblock %}
   </main>
</body>
</html>
```

8. Página principal con formulario (Jinja2)

templates/notas.html

```
{% extends "base.html" %}

{% block contenido %}

<h2>Lista de notas</h2>

<ul>
   {% for n in notas %}
      <li>{{ n }}</li>
   {% endfor %}
</ul>

<form action="/agregar" method="POST">
   <input type="text" name="nota" required placeholder="Nueva nota">
   <button type="submit">Añadir</button>
</form>

{% endblock %}
```

9. Código Flask completo (GET + POST)

app.py

```
from flask import Flask, render_template, request, redirect

app = Flask(__name__)

# Base de datos en memoria (solo mientras el servidor está encendido)
notas = []

@app.route("/")
def inicio():
   return render_template("notas.html", notas=notas)

@app.route("/agregar", methods=["POST"])
def agregar():
   nueva = request.form["nota"]
   notas.append(nueva)
   return redirect("/")   # Redirige a la página principal (método GET)

if __name__ == "__main__":
   app.run(debug=True)
```

10. Explicación del flujo GET/POST

1. El usuario abre / →
   Flask ejecuta la función `inicio()` con un **GET**
   → se muestran las notas.
2. El usuario envía el formulario →
   se hace un **POST** a **/agregar**
3. Flask recibe el dato en `request.form[...]` →
   lo añade a la lista
4. Flask hace un **redirect("/")** →
   el navegador vuelve a cargar la página con un **GET**.

Este patrón se llama:

PRG: Post – Redirect – Get

y evita que el formulario se re-envíe si el usuario recarga.

11. Introducción mínima a Jinja2

Jinja2 es un lenguaje de plantillas que permite:

✓ Insertar valores

{{ nombre }}

✓ Hacer bucles

{% for x in lista %}
  {{ x }}
{% endfor %}

✓ Hacer condiciones

{% if edad >= 18 %} Mayor de edad {% endif %}

✓ Extender plantillas (`extends`, `block`)
→ permite no repetir código.

12. Resumen de lo aprendido

- Cómo crear una aplicación mínima con Flask
- Qué son las rutas y cómo se definen
- Diferencias entre GET y POST
- Uso de formularios y `request.form`
- Redirecciones
- Uso básico de plantillas con Jinja2 (`{{ }}`, `{% %}`, `extends`)
- Mini ejemplo práctico completamente funcional