Uso de Blueprint en Flask

¿Que es un Blueprint?

Un Blueprint en Flask es una herramienta que permite organizar y estructurar una aplicacion web de forma modular.

Funciona como una especie de "mini-aplicacion" que contiene sus propias rutas, vistas, formularios y logica, y que luego puede ser registrada dentro de una aplicacion Flask principal.

Ventajas de usar Blueprint:

- Facilita la separacion de funcionalidades (por ejemplo: modulos de alta, baja, modificacion).
- Mejora la legibilidad y mantenibilidad del codigo.
- Permite reutilizar codigo en diferentes proyectos.
- Ayuda a escalar aplicaciones grandes dividiendolas por secciones.

¿Cómo se usa un Blueprint?

1. Crear un archivo Python y definir el blueprint:

```
# archivo: alta.py
from flask import Blueprint, render_template, request

alta_bp = Blueprint('alta_bp', __name__)

@alta_bp.route('/alta', methods=['GET', 'POST'])
def alta():
return render_template('alta.html')
```

2. Registrar el blueprint en app.py:

```
from flask import Flask

from alta import alta_bp


app = Flask(__name__)

app.register_blueprint(alta_bp)
```

¿Cómo fue útil Blueprint en este proyecto?

En el proyecto CRUD Empresarial, Blueprint se utilizó para dividir las operaciones CRUD en diferentes módulos:

- alta.py para el modulo de registro de personal.
- modificacion.py para el modulo de edicion.
- baja.py para el modulo de eliminacion de registros.

Cada archivo blueprint:

- Define su propia logica de rutas.
- Usa sus propias plantillas (.html).
- Se conecta con la base de datos.

Esto hizo posible que el archivo principal app.py quedara limpio, simplemente registrando cada modulo.

Esto no solo hace el codigo mas limpio, sino tambien mas escalable y profesional.

Conclusion

El uso de Blueprint en Flask es una practica recomendada cuando se trabaja en proyectos medianos o

grandes.

En este proyecto fue clave para estructurar el CRUD de forma clara, modular y reutilizable.