

Resumen paso a paso - Proyecto Flask + PostgreSQL local

1. Verificar instalación de Python y pip

```
python3 --version
```

```
pip3 --version
```

Comprueba que Python 3 y pip están disponibles en el sistema.

2. Crear entorno virtual

```
python3 -m venv venv
```

```
source venv/bin/activate
```

Crea un entorno virtual y lo activa, para aislar las dependencias del proyecto.

3. Instalar dependencias del proyecto

```
pip install -r requirements.txt
```

Instala Flask y psycopg2-binary desde el archivo requirements.txt.

4. Instalar y configurar PostgreSQL

```
brew install postgresql@17
```

```
brew services start postgresql@17
```

Instala y arranca PostgreSQL como servicio con Homebrew.

5. Añadir PostgreSQL al PATH

```
export PATH="/opt/homebrew/opt/postgresql@17/bin:$PATH"
```

Agrega psql al entorno para poder usarlo en terminal.

6. Crear base de datos y usuario

```
psql optimec
```

```
ALTER SCHEMA public OWNER TO postgres;
```

```
\q
```

Accede como usuario local y transfiere propiedad del esquema public.

7. Ejecutar script SQL

```
psql -U postgres -d optimec -f ../db/init.sql
```

Crea la tabla 'contactos' desde el archivo SQL.

8. Modificar host de conexión en app.py

Cambiar host='db' por host='localhost'

Permite conectar con PostgreSQL localmente sin Docker.

9. Ajustar carpeta de uploads

```
UPLOAD_FOLDER = os.path.join(os.path.dirname(__file__), "uploads")
```

Permite guardar archivos adjuntos dentro del proyecto local.

10. Ejecutar app Flask

```
venv/bin/python app.py
```

Inicia el servidor en `http://localhost:5050`

11. Probar formulario

Abrir `frontend/index.html` en el navegador

Enviar datos al backend para guardar en la base.

12. Consultar datos desde PostgreSQL

```
psql -U postgres -d optimec
```

```
SELECT * FROM contactos;
```

Muestra todos los datos almacenados en la tabla.

¿Cómo funciona esto con Docker?

En Docker, los servicios (Flask, PostgreSQL, etc.) están definidos en `docker-compose.yml`.

El hostname 'db' se refiere al servicio PostgreSQL dentro del contenedor.

Usamos Docker para encapsular y ejecutar toda la app sin instalar nada en el sistema.