Manual de DevOps – CI/CD y Docker

# 1. Configuración del Pipeline CI/CD

* **CI:**

Se ejecuto desde visual studio code.

**Captura de pantalla con la imagen de una pantalla

El contenido generado por IA puede ser incorrecto.**

**Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.**

* **CD:**

**Dockerfile**

Texto

El contenido generado por IA puede ser incorrecto.

**Docker-compose**

**Texto

El contenido generado por IA puede ser incorrecto.**

# 2. Dockerización del Backend

Para poder dockerizarlo, se necesitó tener 2 documentos llamados Docker-compose.yml y el documento Dockerfile. Comenzando hay que modificar el Dockerfile, este define la variable de entorno LANG para que todo dentro del contenedor use esa configuración.

El document Docker-compose:

* **Levanta PostgreSQL** usando el Dockerfile anterior.
* **Configuraciones importantes**:
  + - POSTGRES\_DB=flight\_management → crea esa base al iniciar.
    - POSTGRES\_USER=postgres y POSTGRES\_PASSWORD=abcd1234 → credenciales por defecto.
    - ports: "5433:5432" → accedes desde tu máquina en el puerto **5433** (el contenedor escucha en el **5432** interno).
    - volumes:
    - postgres\_data → guarda los datos de la BD en tu host (persistencia).
    - /home/ubuntu/db\_backups:/backups/postgres → carpeta para respaldos en tu servidor.
    - healthcheck → verifica que PostgreSQL esté corriendo antes de que otros servicios dependan de él.
    - labels → simple metadata para organizar.

# 3. Guía práctica de despliegue

1. Clonar el repositorio del mismo GitHub.
2. Tener instalado Docker y WSL.
3. Entrar a el editor Visual Studio Code.
4. Entrar desde el editor de código a la carpeta correspondiente.
5. Entrar a la carpeta Postgres alojada en la dirección week4->sesión 2->material->architecture->devop.
6. Ejecutar el archivo “Docker-compose.yml” con la extensión: Interfaz de usuario gráfica, Texto

   El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

1. Configurar el Docker con las entidades usadas con terminal:

Primero:

docker exec -it serve-postgres-sena psql -U postgres -d flight\_management

Luego:

CREATE SCHEMA aircraft\_management;

CREATE SCHEMA flight;

CREATE SCHEMA geolocation;

CREATE SCHEMA human\_resources;

CREATE SCHEMA infrastructure;

CREATE SCHEMA notification;

CREATE SCHEMA parameterization;

CREATE SCHEMA passengersServices;

CREATE SCHEMA security;

Esto se hace para crear cada entidad y que el Backend pueda trabajar con normalidad.

1. Ejecutar el Backend por medio del editor de código teniendo esta extensión Interfaz de usuario gráfica, Texto

   El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

1. Entrar a la dirección http://localhost:8081/swagger-ui/index.html#/