

Plantilla Docker

Este documento describe la estructura de la plantilla y los pasos para inicializar el entorno de desarrollo.

1. Estructura de carpetas

```
PLANTILLA_DOCKER/
├── PROYECTO/
│   ├── pom.xml                # Configuración de Maven
│   └── target/                # Artefactos compilados (.WAR)
├── docker-compose.yml        # Orquestación de servicios
└── Docker
    ├── DB_DUMP/              # Script de inicialización de BD
    │   └── postgresqlCrearProyectoBase.sql
    ├── Docker/               # Dockerfiles por servicio
    │   ├── APP/
    │   │   └── Dockerfile     # Imagen WildFly 29.0.1
    │   └── DB/
    │       └── Dockerfile     # Imagen PostgreSQL 16
    └── data/                  # Volumen de datos crudos
                                #data/ (Se crea al desplegar los
                                contenedores)
```

2. Servicio definidos en `docker-compose.yml`

- **db:** PostgreSQL 16
 - **wildfly:** WildFly 29.0.1
 - **redes:** configuración de red interna para comunicación entre servicios
 - **volúmenes:**
 - **data/** → **/var/lib/postgresql/data**
 - **DB_DUMP/** → **/docker-entrypoint-initdb.d**
-

3.Despliegue

1. **Construir imágenes** - [Desde la raíz de la carpeta PLANTILLA DOCKER]

SHELL

```
docker-compose build
```

2. **Levantar contenedores**

SHELL

```
docker-compose up -d
```

3. **Verificar estado**

SHELL

```
docker-compose ps
```

Se deben listar los contenedores:

Pgsql_proy y **Wf_proy**

Accesos:

- Consola de administración WildFly:
<http://localhost:9990/console/index.html>
- **Usuario: admin**
- **Contraseña: 123456**
- Aplicación desplegada: <http://127.0.0.1:8081/proyectoBase/>

4. Personalización de credenciales y nombres

Por defecto, el servicio **db** crea:

```
environment:
  POSTGRES_USER: db_user
  POSTGRES_PASSWORD: 123456
  POSTGRES_DB: db_proy
container_name: Pgsql_proy #Para el contenedor de Postgres
container_name: Wf_proy #Para el contenedor de Wildfly
```

Debe sustituir: `db_user`, `123456`, `db_proy`, `Wf_proy` y `Pgsql_proy` por los valores del proyecto en curso.

5. Validación de la base de datos tras el despliegue

SQL

```
--1.- Acceder al contenedor Postgres:
docker exec -it Pgsql_proy bash

-- 2.- Conectarse con psql:
psql -U db_user -d db_proy -W

--3.- Listar esquemas:
\dn
```

Debe aparecer:

Nombre		Dueño
-----+-----		
esquema_prueba_spring_boot		db_user

Con estos pasos, el entorno queda listo para desplegar la aplicación Java sobre WildFly con PostgreSQL como base de datos.