Añadiendo una base de datos “real”

Elementos avanzados en tu API REST con Spring Boot

Hasta ahora

- Hemos utilizado H2
- De hecho, hemos usado una base de datos *en memoria*
- Válido durante las primeras fases de desarrollo / testing
- Necesitamos una sistema gestor de base de datos que podamos utilizar en producción.

Alternativas

- Oracle (de pago)
- SQL Server (de pago)
- MySQL (gratis, pero de Oracle :S)
- **Postgresql (gratis, opensource)**
- MariaDB
- ….

Postgresql

- Gratuíta
- Opensource
- Multiplataforma
- Alta concurrencia
- Ofrecido gratuitamente en plataformas como heroku

Docker

- Los contenedores de aplicaciones son entornos ligeros de tiempo de ejecución que proporcionan a las aplicaciones los archivos, las variables y las bibliotecas que necesitan para ejecutarse, maximizando de esta forma su portabilidad.

Docker

- Nos permite desplegar algunas aplicaciones rápidamente.
- En lecciones posteriores lo haremos con nuestra propia aplicación de ejemplo.
- Ahora lo vamos a utilizar para desplegar de forma muy sencilla nuestro SGBD.
- Si quieres saber más, puedes visitar nuestro Curso de Introducción a Docker

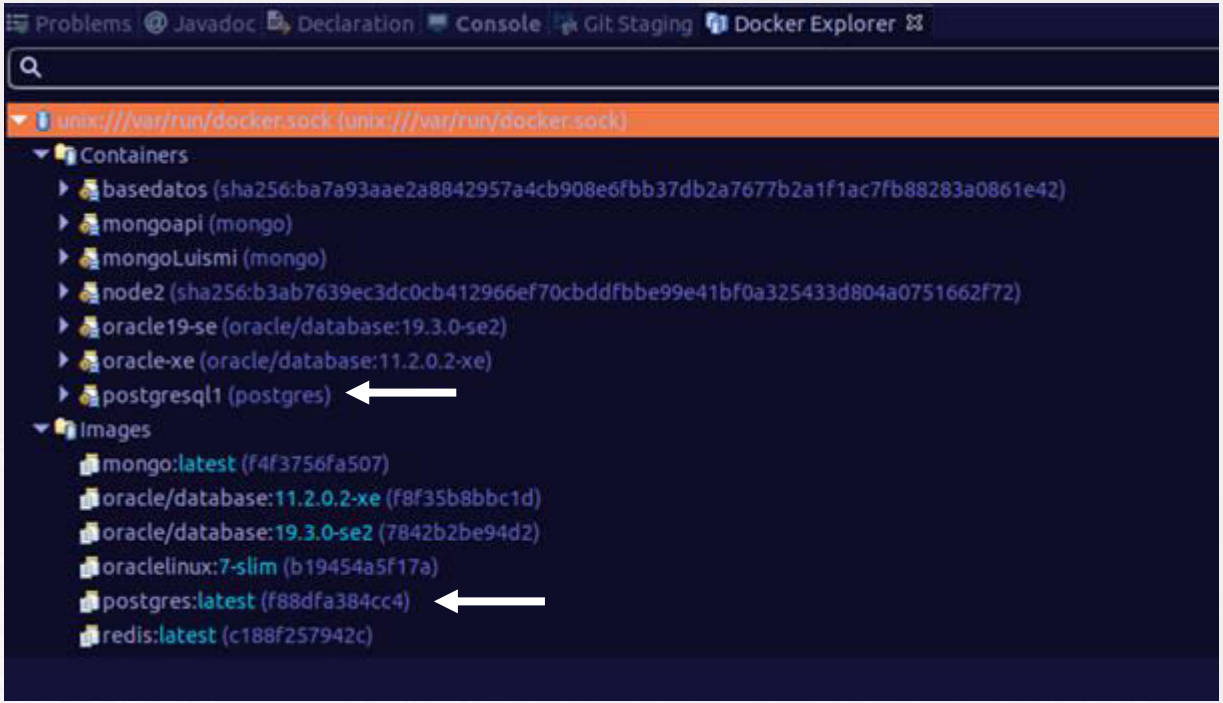
Docker y eclipse

- Help > Eclipse Marketplace
- Buscamos *docker*
- Instalamos *Eclipse Docker Tooling* y *Docker Editor*
- Una vez instalado (y reiniciado Eclipse), Window > Show View > Docker Explorer

Instalar Postgresql con Docker

- `docker pull postgres`
- `docker run --name postrgresql1 -p 5432:5432 -e POSTGRES_PASSWORD=postgresql -d postgres`
  - *--name* Nombre del contenedor
  - *-p* Conexión de un puerto externo con interno
  - *-e* Establecemos una variable de entorno
  - *-d* El contenedor corre en background

Instalar Postgresql con Docker



Cliente para Postgresql

- Desde el terminal
- docker exec -ti postgresql1 psql -U postgres -W postgres
  - *postgresql1:* Nombre del contenedor
  - *psql:* Cliente de Postgresql
  - *-U postgres:* Nombre de usuario
  - -W: nos solicitará contraseña
  - *postrgres*: Nombre de la base de datos

Cliente gráfico para Postgresql

- pgAdmin 4
- https://www.pgadmin.org
- Entorno web / Instalación vía docker

Spring Boot y Postgresql

- Ahora que tenemos el entorno, ¿cómo conectamos nuestra API REST con Postgresql?
- Añadimos dependencia en *pom.xml*
- Añadimos algunas *properties* para definir los parámetros del *Datasource*.

pom.xml

- Edit starters > Postgresql Driver
- O bien, añadir directamente:

```
<dependency>
    <groupId>org.postgresql</groupId>
        <artifactId>postgresql</artifactId
>
        <scope>runtime</scope>
</dependency>
```

application.properties

```
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres

spring.datasource.username=postgres

spring.datasource.password=postgresql

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=create-drop

spring.datasource.initialization-mode=always
```

Ejecutamos y comprobamos que funciona igual que antes.

Reto

- Realizar la misma configuración para Postgresql con el proyecto que hemos ido confeccionando en las primeras lecciones del curso (te puedes basar en el proyecto **ManyToMany** del repositorio).