Ejercicio práctico

Enunciado

Crear una aplicación SPRING BOOT que ofrezca un API REST para el mantenimiento de una entidad almacenada en una base de datos MySQL.

Alojar la aplicación y la base de datos en sendos contenedores Docker.

1. Crear contenedor MySQL con volumen (datos)

```
docker run --name movies_mysql -v mysql_data:/var/lib/MySQL -p
3306:3306 -e MYSQL_ROOT_PASSWORD=my_secret_pw -e
MYSQL_DATABASE=moviesdb -d mysql:latest
```

Por defecto, MySQL crea el usuario **root**.

2. Conectar con una herramienta de administración de MySQL con la base de datos

USAR ESTA CADENA DE CONEXIÓN:

jdbc:mysql://localhost:3306/moviesdb?allowPublicKeyRetrieval=true&useSSL=false

3. Crear aplicación WS CRUD

1. **Crear proyecto Maven con spring initializr.**

Dependencias:

- spring web
- spring boot devtools
- spring data jpa
- MySQL driver
- Rest Repositories

2. **Configurar application.properties:**

```
spring.datasource.url=jdbc:mysql://localhost:3306/moviesdb?allow
PublicKeyRetrieval=true&useSSL=false

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.My
SQLDialect
```

3. **Crear la entidad.**

Anotarla como @Entity y realizar el resto de anotaciones.

4. **Crear el repositorio:**

```
@RepositoryRestResource
```

```
public interface IMovieRepository  extends JpaRepository<Movie,
Long> {

}
```

5. **Crear la documentación del API.**

Dependencia de OpenAPI (Swagger):

```
<dependency>

      <groupId>org.springdoc</groupId>

      <artifactId>springdoc-openapi-starter-webmvc-
ui</artifactId>

      <version>2.5.0</version>

</dependency>
```

NOTA: La documentación del API se consulta en http://localhost:8088/swagger-ui/index.html

6. **Crear imagen DOCKER**

- Crear el fichero **dockerfile** en la carpeta raíz del proyecto:

```
# Usar una imagen base de JDK de Amazon Corretto
FROM amazoncorretto:17-alpine-jdk

# Establecer el directorio de trabajo
WORKDIR /app

# Copiar el archivo JAR generado en el contenedor
COPY target/*.jar app.jar

# Exponer el puerto en el que la aplicación se ejecutará
EXPOSE 8080

# Comando para ejecutar la aplicación
ENTRYPOINT ["java", "-jar", "app.jar"]
```

- Empaquetar el proyecto.

```
mvn clean package -DskipTests
```

- Crear el contenedor ejecutando desde la raíz del proyecto el siguiente comando:

```
docker build -t nombre-de-tu-imagen .
```