

# Mi primer proyecto MAVEN

Iris Pérez Aparicio

2º DAM

Campus FP Emprende Humanes

Abro docker y levanto un contenedor mysql:

## mysql-composer.yaml

```
services:
  mysql-db:
    image: mysql:latest
    container_name: mi-contenedor-mysql
    environment:
      MYSQL_ROOT_PASSWORD: root123
      MYSQL_DATABASE: testdb
      MYSQL_USER: user123
      MYSQL_PASSWORD: user123pw
      TZ: "Europe/Madrid"

    ports:
      - "3306:3306" # Puerto de MySQL

    volumes:
      - mysql_data:/var/lib/mysql
      - /etc/localtime:/etc/localtime:ro
      - /etc/timezone:/etc/timezone:ro

    networks:
      - mysql-network

networks:
  mysql-network:
    driver: bridge

volumes:
  mysql_data:
    driver: local
```

```

# (Abrir docker)

# Levantar los servidores
# docker-compose -f mysql-composer.yaml up -d

# verificar los contenedores:
# docker ps (la consola devuelve los contenedores que tenemos)


# Abrir MySql Workbench y crear una conexión
# Nombre: (el que quieras)
# Contraseña para acceder: root123


# detener los servicios
# docker-compose -f mysql-composer.yaml down


# reiniciar los servicios
# docker-compose -f mysql-composer.yaml up -d

```









(Ejecuto el comando para levantar los servidores desde el terminal de VSCode)

```

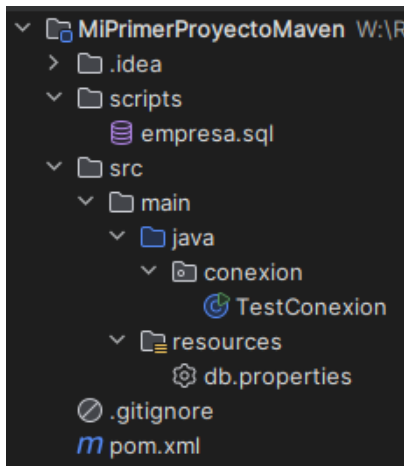
PS W:\REPOS\docker\mysql> docker-compose -f mysql-composer.yaml up -d
[+] Running 3/3
 ✓ Network mysql_mysql-network      Created
 ✓ Volume mysql_mysql_data          Created
 ✓ Container mi-contenedor-mysql    Started
PS W:\REPOS\docker\mysql> 

```

Contenedor levantado en docker:

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	mysql	-	-	-	0.41%	2 minutes ago	   
<input type="checkbox"/>	mi-contenedor-mysql	95042b79c970	<a href="#">mysql:latest</a>	<a href="#">3306:3306</a>	0.41%	2 minutes ago	   

## ESTRUCTURA DEL PROYECTO



Creo la base de datos desde MySQL Workbench:

### Archivo **scripts/empresa.sql**

```
CREATE DATABASE IF NOT EXISTS empresa;
USE empresa;

CREATE TABLE empleados (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50),
    salario DOUBLE
);

INSERT INTO empleados (nombre, salario)
VALUES ('Ana', 25000), ('Luis', 28000), ('Marta', 32000);
```

#	Time	Action
✓ 1	20:53:02	CREATE DATABASE IF NOT EXISTS empresa
✓ 2	20:53:02	USE empresa
✓ 3	20:53:02	CREATE TABLE empleados ( id INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(50), salario DOUBLE )
✓ 4	20:53:02	INSERT INTO empleados (nombre, salario) VALUES ('Ana', 25000), ('Luis', 28000), ('Marta', 32000)

SCHEMAS	
🔍	Filter objects
▶ 🗄️	<b>empresa</b>

### Archivo `src/main/java/resources/db.properties`

```
db.url=jdbc:mysql://localhost:3306/empresa
db.user=root
db.password=root123
```

### Archivo `src/main/java/conexion/TestConexion.java`

```
package conexion;

import java.io.InputStream;
import java.sql.*;
import java.util.Properties;

public class TestConexion {
    public static void main(String[] args) {
        // 1. Cargar configuración desde db.properties
        Properties props = new Properties();
        try (InputStream input =
TestConexion.class.getClassLoader().getResourceAsStream("db.pr
operties")) {
            if (input == null) {
                System.err.println("✗ No se encontró el
archivo db.properties");
                return;
            }
            props.load(input);
        } catch (Exception e) {
            e.printStackTrace();
            return;
        }
    }
}
```

```

// 2. Obtener datos de conexión
String url = props.getProperty("db.url");
String user = props.getProperty("db.user");
String password = props.getProperty("db.password");

// 3. Probar conexión
try (Connection con = DriverManager.getConnection(url,
user, password)) {
    System.out.println("✅ Conexión establecida con
éxito a la base de datos.");

    // Mostrar metadatos
    DatabaseMetaData meta = con.getMetaData();
    System.out.println("♦ Driver: " +
meta.getDriverName());
    System.out.println("♦ Versión del driver: " +
meta.getDriverVersion());
    System.out.println("♦ Base de datos: " +
meta.getDatabaseProductName());
    System.out.println("♦ Versión BD: " +
meta.getDatabaseProductVersion());
    System.out.println("♦ Usuario conectado: " +
meta.getUserName());
    System.out.println("♦ URL de conexión: " +
meta.getURL());

    // 4. Consulta de ejemplo
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("SELECT * FROM
empleados");

    System.out.println("\n=== EMPLEADOS ===");
    while (rs.next()) {

```

```

        System.out.printf("ID: %d | Nombre: %s |
Salario: %.2f €\n",
            rs.getInt("id"),
rs.getString("nombre"), rs.getDouble("salario"));
    }

    } catch (SQLException e) {
        System.err.println("✗ Error al conectar a la base
de datos: " + e.getMessage());
    }
}
}

```

Ejecuto el archivo **TestConexion.java**

(Se obtienen los datos de la base de datos y se muestran por consola)

```

✅ Conexión establecida con éxito a la base de datos.
💠 Driver: MySQL Connector/J
💠 Versión del driver: mysql-connector-j-8.0.33 (Revision: 7d6b0800528b6b25c68b52dc10d6c1c8429c100c)
💠 Base de datos: MySQL
💠 Versión BD: 9.5.0
💠 Usuario conectado: root@192.168.65.1
💠 URL de conexión: jdbc:mysql://localhost:3306/empresa

=== EMPLEADOS ===
ID: 1 | Nombre: Ana | Salario: 25000,00 €
ID: 2 | Nombre: Luis | Salario: 28000,00 €
ID: 3 | Nombre: Marta | Salario: 32000,00 €

Process finished with exit code 0

```