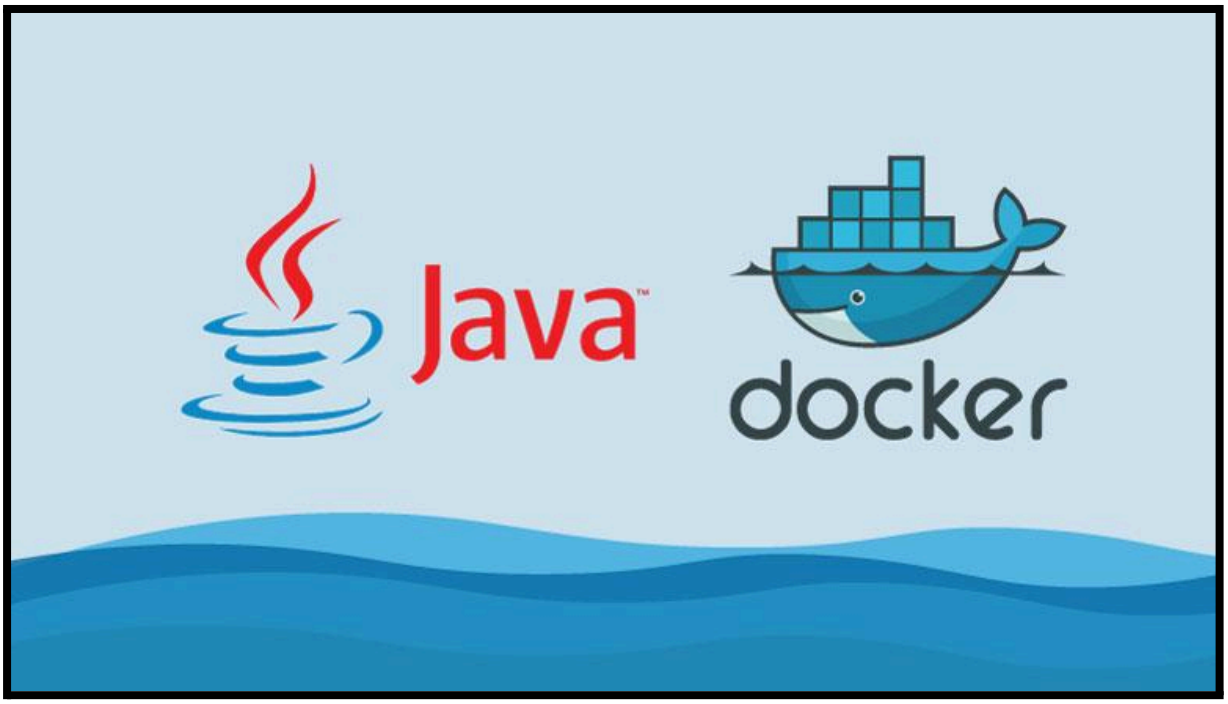
Despliegue de una mini aplicación JAVA



Nombre y Apellido	Laura San Román
Curso	2º - DAW
Fecha	12 / 02 / 2026
Asignatura	Despliegue de aplicaciones web

SERVIDOR WEB VIRTUALIZADO

La estructura es:

Carpeta llamada servidor-java

- app
    - DockerFile
    - index.html
    - InsertarUsuarioServlet.java
    - mariadb-java-client-3.5.6.jar

- db
    - init.sql

- docker-compose.yml

Sabiendo la estructura que tiene, pasamos a la configuración para luego construir el contenedor.

La arquitectura de IP es:

- Apache -> “8080:80”
- Tomcat-> “8081:80”
- MariaDB-> “3309:3306”
- PhpMyAdmin-> “5004:80”

CONFIGURACIÓN DE LA CARPETA APP

Paso 1. Entrar a la cmd -> cd C:\Users\dawmi\servidor-java\app

Paso 2. Crear un .txt llamado DockerFile donde se guardará la siguiente información
->

Dockerfile

```
# 1. Imagen base de Tomcat con JDK
FROM tomcat:10.1-jdk17

# 2. Crear carpetas para las clases del servlet
RUN mkdir -p /usr/local/tomcat/webapps/ROOT/WEB-INF/classes/com/ejemplo

# 3. Copiar archivos al contenedor
COPY InsertarUsuarioServlet.java /usr/local/tomcat/webapps/ROOT/WEB-INF/classes/com/ejemplo/
COPY index.html /usr/local/tomcat/webapps/ROOT/
COPY mariadb-java-client-3.5.6.jar /usr/local/tomcat/lib/

# 4. Compilar el servlet usando servlet-api.jar + MariaDB driver
RUN javac -cp "/usr/local/tomcat/lib/servlet-api.jar:/usr/local/tomcat/lib/mariadb-java-client-3.5.6.jar" \
    -d /usr/local/tomcat/webapps/ROOT/WEB-INF/classes \
    /usr/local/tomcat/webapps/ROOT/WEB-INF/classes/com/ejemplo/InsertarUsuarioServlet.java

# 5. Exponer puerto 8080 (Tomcat)
EXPOSE 8080

# 6. Arrancar Tomcat al iniciar el contenedor
CMD ["catalina.sh", "run"]

```

Y se guardará como *todos los archivos para que no tenga extensión ya que sino, no funciona.

Paso 3. Construir el index.html
->

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insertar Usuario</title>
</head>
<body>
    <h1>Agregar Usuario</h1>
    <!-- FORMULARIO: nombre y contraseña -->
    <form method="post" action="http://localhost:8081/InsertarUsuario">
        <label for="nombre">Nombre de usuario:</label>
        <input type="text" id="nombre" name="nombre" required>
        <br><br>
        <label for="password">Contraseña:</label>
        <input type="password" id="password" name="password" required>
        <br><br>
        <button type="submit">Insertar</button>
    </form>
</body>
</html>
```

Paso 4. Meter el código java (cogido del Aula Virtual) y añadir la parte de “password” ->

InsertarUsuarioServlet.java

app > InsertarUsuarioServlet.java

```java
package com.ejemplo;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/InsertarUsuario")
public class InsertarUsuarioServlet extends HttpServlet {

    // Configura estos valores según tu entorno
    private static final String URL = "jdbc:mariadb://mariadb-daw:3306/mi_bd";
    private static final String USER = "usuario";
    private static final String PASS = "password";

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html;charset=UTF-8");

        // Recoger los parámetros del formulario
        String nombre = request.getParameter("nombre");
        String password = request.getParameter("password"); // NUEVO

        try (PrintWriter out = response.getWriter()) {

            // 1. Cargar el driver de MariaDB
            try {
                Class.forName("org.mariadb.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                out.println("No se encontró el driver de MariaDB: " + e.getMessage());
                return;
            }

            // 2. Obtener conexión
            try (Connection conn = DriverManager.getConnection(URL, USER, PASS)) {

                // 3. Instrucción SQL preparada para INSERT (nombre + password)
                String sql = "INSERT INTO usuarios (nombre, password) VALUES (?, ?)";

                try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                    stmt.setString(1, nombre);
                    stmt.setString(2, password); // NUEVO

                    int filas = stmt.executeUpdate(); // ejecuta INSERT

                    if (filas > 0) {
                        out.println("Usuario insertado correctamente.");
                    } else {
                        out.println("No se pudo insertar el usuario.");
                    }
                }

            } catch (SQLException e) {
                out.println("Error de base de datos: " + e.getMessage());
            }
        }
    }
}
```

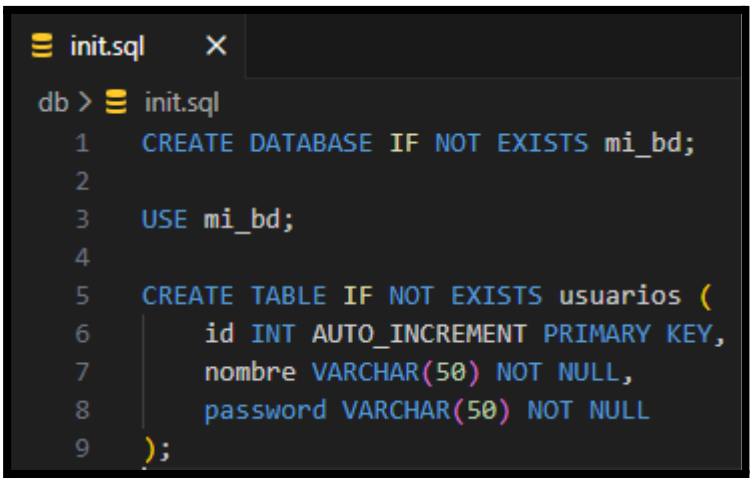
Paso 5. Descargar el .jar de mariadb y meterlo en la carpeta “app”

-> mariadb-java-client-3.5.6.jar

CONFIGURACIÓN DE LA CARPETA DB

Paso 1. Añadir el init.sql

->



CONFIGURACIÓN DEL DOCKER-COMPOSE.YML

Paso 1. El docker-compose.yml contiene lo siguiente

->

docker-compose.yml

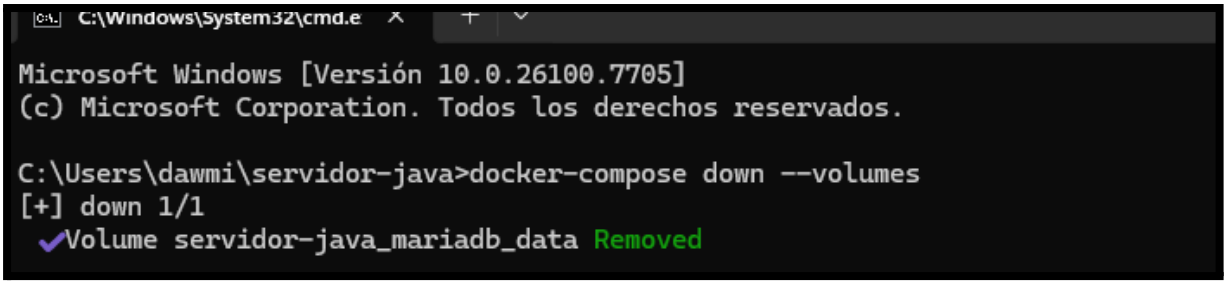
```yaml
services:
  apache-web:
    image: httpd:latest
    container_name: apache-web
    ports:
      - "8080:80"
    volumes:
      - ./app/index.html:/usr/local/apache2/htdocs/index.html

  tomcat:
    build: ./app
    container_name: servidor-java-app
    ports:
      - "8081:8080"
    depends_on:
      - mariadb

  mariadb:
    image: mariadb:latest
    container_name: mariadb-daw
    environment:
      MARIADB_ROOT_PASSWORD: root
      MARIADB_DATABASE: mi_bd
      MARIADB_USER: usuario
      MARIADB_PASSWORD: password
    ports:
      - "3309:3306"
    volumes:
      - mariadb_data:/var/lib/mysql
      - ./db:/docker-entrypoint-initdb.d

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    container_name: phpmyadmin-java
    environment:
      PMA_HOST: mariadb-daw
      PMA_USER: usuario
      PMA_PASSWORD: password
    ports:
      - "5004:80"
    depends_on:
      - mariadb

volumes:
  mariadb_data:
```

RESULTADOS

Paso 1. Desde la cmd de “servidor-java”. Haremos los dos siguiente comandos.

-> docker-compose down --volumes

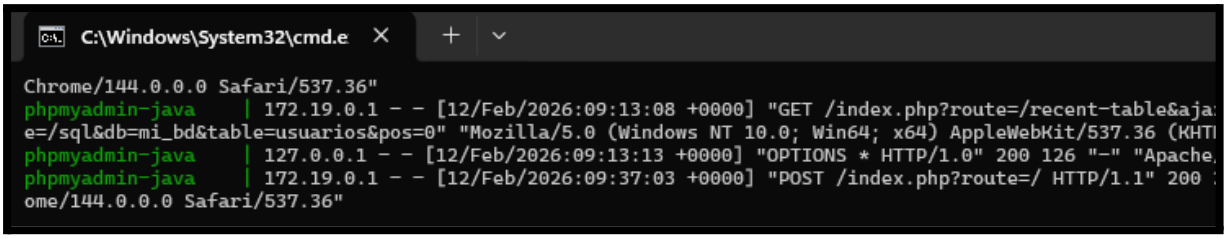
Para eliminar configuraciones previas.



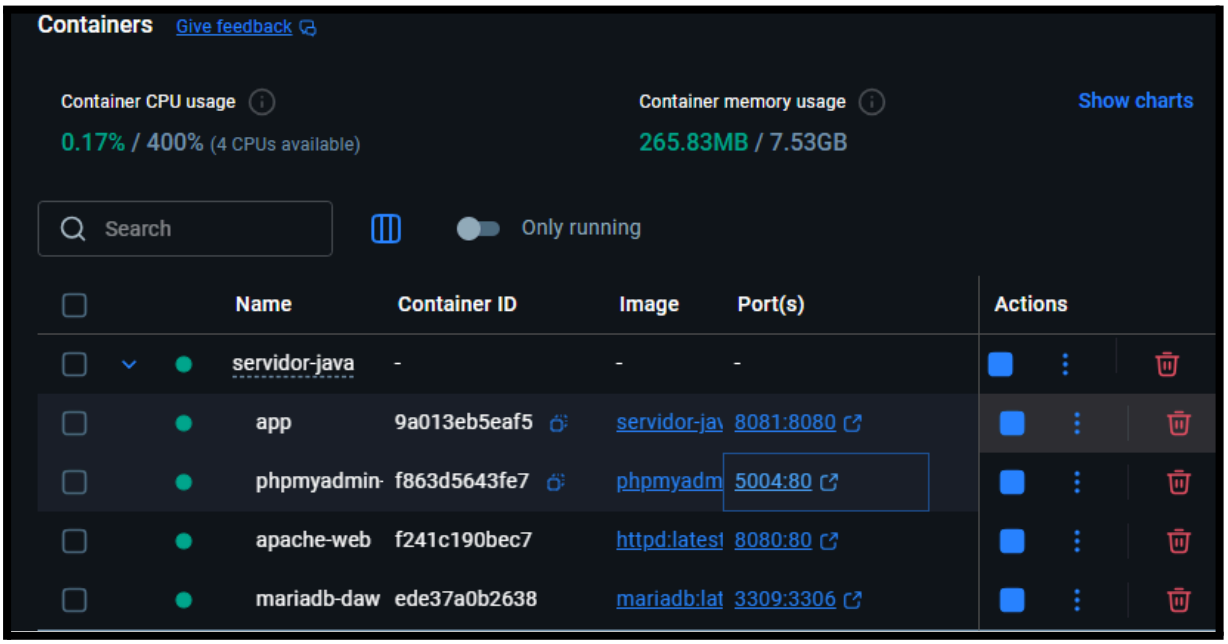
Paso 2. Desde la cmd de “servidor-java”. Segundo comando.

-> docker-compose up --build

Para construir y levantarlo.

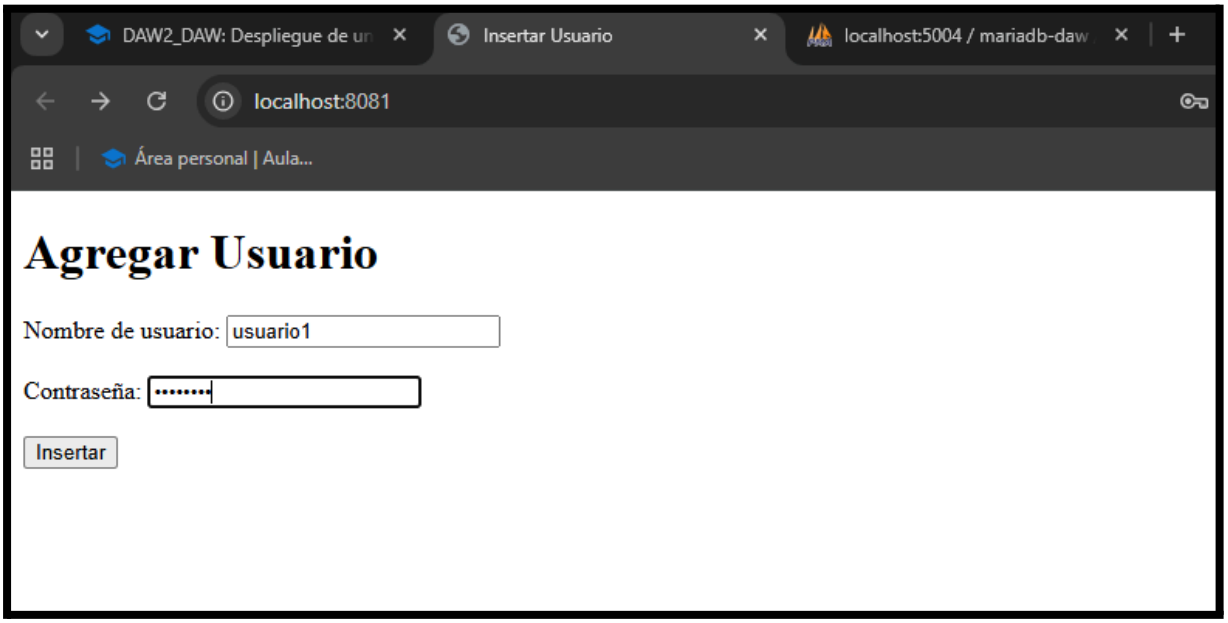


Así quedaría el docker:

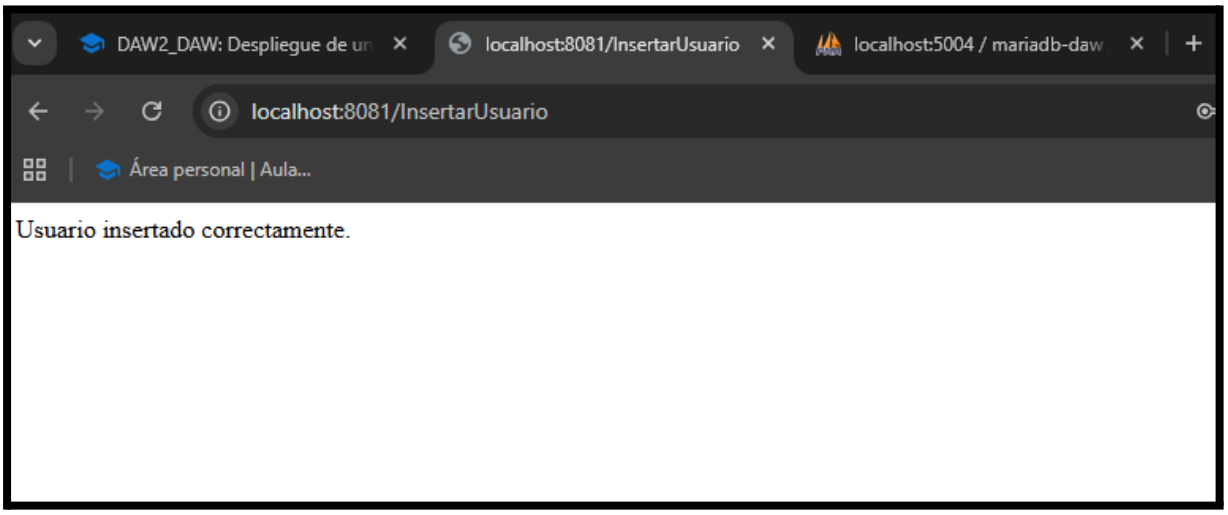


Al abrir el index.html

->

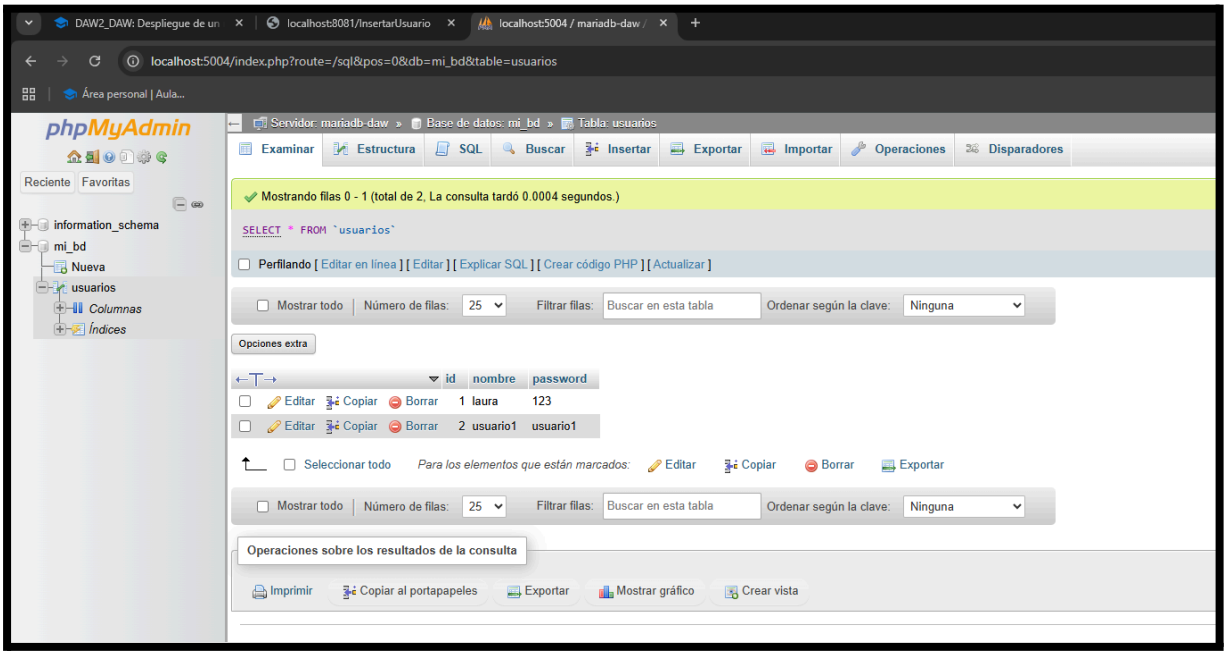


->



Y esto quedaría reflejado en la base de datos.

->



VERIFICACIÓN DEL ENTORNO

Paso Final. Para comprobar que ha sido desplegada en mi entorno local (mi ordenador).

->