## 💻 Entorno de Desarrollo BBDD con Docker Compose (MVE)

Aquí tienes el archivo docker-compose.yml que define los servicios. Está diseñado para ser un **Entorno Mínimo Viable (MVE)**, fácil de configurar y ejecutar.

YAML

version: '3.8' # Versión de Docker Compose  
  
services:  
 # 1. Servicio de Base de Datos: MySQL  
 db:  
 image: mysql:8.0 # Utiliza la imagen oficial de MySQL 8.0  
 container\_name: mysql-dev-bbdd  
 # Reinicia automáticamente si falla o se detiene  
 restart: always   
 environment:  
 # Variables esenciales para la configuración inicial de MySQL  
 MYSQL\_ROOT\_PASSWORD: rootpassword # ¡IMPORTANTE! Cambiar en producción  
 MYSQL\_DATABASE: mi\_bbdd\_dev  
 MYSQL\_USER: devuser  
 MYSQL\_PASSWORD: devpassword  
 volumes:  
 # Mapea un volumen persistente para guardar los datos de la BBDD  
 - db\_data:/var/lib/mysql   
 # Mapea un archivo SQL para inicializar la BBDD (Opcional)  
 # - ./init\_db.sql:/docker-entrypoint-initdb.d/init\_db.sql  
 ports:  
 # Mapea el puerto 3306 del contenedor al puerto 3306 de tu máquina  
 - "3306:3306"   
  
 # 2. Servicio de Gestión de Base de Datos: phpMyAdmin  
 phpmyadmin:  
 image: phpmyadmin/phpmyadmin  
 container\_name: phpmyadmin-dev-gui  
 restart: always  
 environment:  
 # Configuración para conectar a nuestro servicio 'db' (MySQL)  
 PMA\_HOST: db   
 PMA\_PORT: 3306  
 MYSQL\_ROOT\_PASSWORD: rootpassword # Usa la misma contraseña ROOT que MySQL  
 ports:  
 # Mapea el puerto 80 del contenedor al puerto 8080 de tu máquina  
 - "8080:80"   
 depends\_on:  
 - db # Asegura que el servicio 'db' se inicie antes que phpMyAdmin  
  
# Definición de volúmenes persistentes  
volumes:  
 db\_data:

## 📝 Instrucciones

Sigue estos pasos para poner en marcha tu entorno de desarrollo.

### **Paso 1: Guardar la Configuración**

Guarda el código anterior en un archivo llamado **docker-compose.yml** en una carpeta vacía, por ejemplo, ~/proyectos/entorno-dev-bbdd/.

### **Paso 2: Levantar los Contenedores**

Abre tu terminal, navega hasta la carpeta donde guardaste el archivo y ejecuta el siguiente comando:

Bash

docker compose up -d

* **docker compose up**: Construye y levanta los servicios definidos en el archivo docker-compose.yml.
* **-d**: Ejecuta los contenedores en **modo *detached*** (segundo plano) para que puedas seguir usando tu terminal.

### **Paso 3: Verificar el Estado**

Verifica que ambos contenedores (MySQL y phpMyAdmin) se han levantado correctamente:

Bash

docker compose ps

Deberías ver ambos servicios (db y phpmyadmin) con el estado **running**.

### **Paso 4: Acceder a phpMyAdmin**

Abre tu navegador web y ve a la siguiente dirección:

http://localhost:8082

Podrás iniciar sesión usando las credenciales definidas en el archivo:

* **Servidor:** db (ya debería estar seleccionado)
* **Nombre de usuario:** devuser
* **Contraseña:** devpassword
* *(También puedes acceder como root con Usuario: root y Contraseña: rootpassword)*

## 🛠️ Ejemplo Práctico: Acceso a MySQL desde la Terminal

Este es un ejemplo de cómo interactuar con tu base de datos directamente desde la terminal, simulando una conexión cliente.

### **1. Acceder al Contenedor MySQL**

Para acceder a la terminal del contenedor de MySQL, utiliza el comando exec.

Bash

docker exec -it mysql-dev-bbdd bash

**docker exec**: Ejecuta un comando en un contenedor en ejecución.

* **-it**: Mantiene la entrada interactiva (-i) y asigna un terminal (-t).
* **mysql-dev-bbdd**: Es el nombre del contenedor de la base de datos que definimos antes.
* **bash**: Es el comando que queremos ejecutar dentro del contenedor (abre un *shell*).

### **2. Conectarse al Servidor MySQL**

Una vez dentro del *shell* del contenedor, conéctate al servidor MySQL como el usuario **devuser**:

Bash

mysql -u devuser -p mi\_bbdd\_dev

* Te pedirá la **contraseña**: introduce devpassword.

### **3. Ejecutar Comandos SQL**

Ahora puedes ejecutar comandos SQL directamente:

#### **Crear una tabla de ejemplo:**

SQL

CREATE TABLE productos (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 nombre VARCHAR(100) NOT NULL,  
 precio DECIMAL(10, 2)  
);

#### **Insertar un registro:**

SQL

INSERT INTO productos (nombre, precio) VALUES ('Laptop Dev', 1200.50);

#### **Consultar los datos:**

SQL

SELECT \* FROM productos;

#### **Salir del cliente MySQL:**

SQL

exit

#### **Salir del contenedor (volver a tu terminal principal):**

Bash

exit

## 🗑️ Para Detener y Eliminar el Entorno

Cuando termines tu sesión de trabajo, puedes detener y eliminar los contenedores y, opcionalmente, los datos.

### **Detener los servicios (manteniendo los datos):**

Bash

docker compose stop

### **Detener y eliminar contenedores y red (manteniendo los datos):**

Bash

docker compose down

### **Detener y eliminar contenedores, red Y los datos persistentes (volumen db\_data):**

**¡Cuidado!** Este comando eliminará todos los datos de tu base de datos.

Bash

docker compose down -v