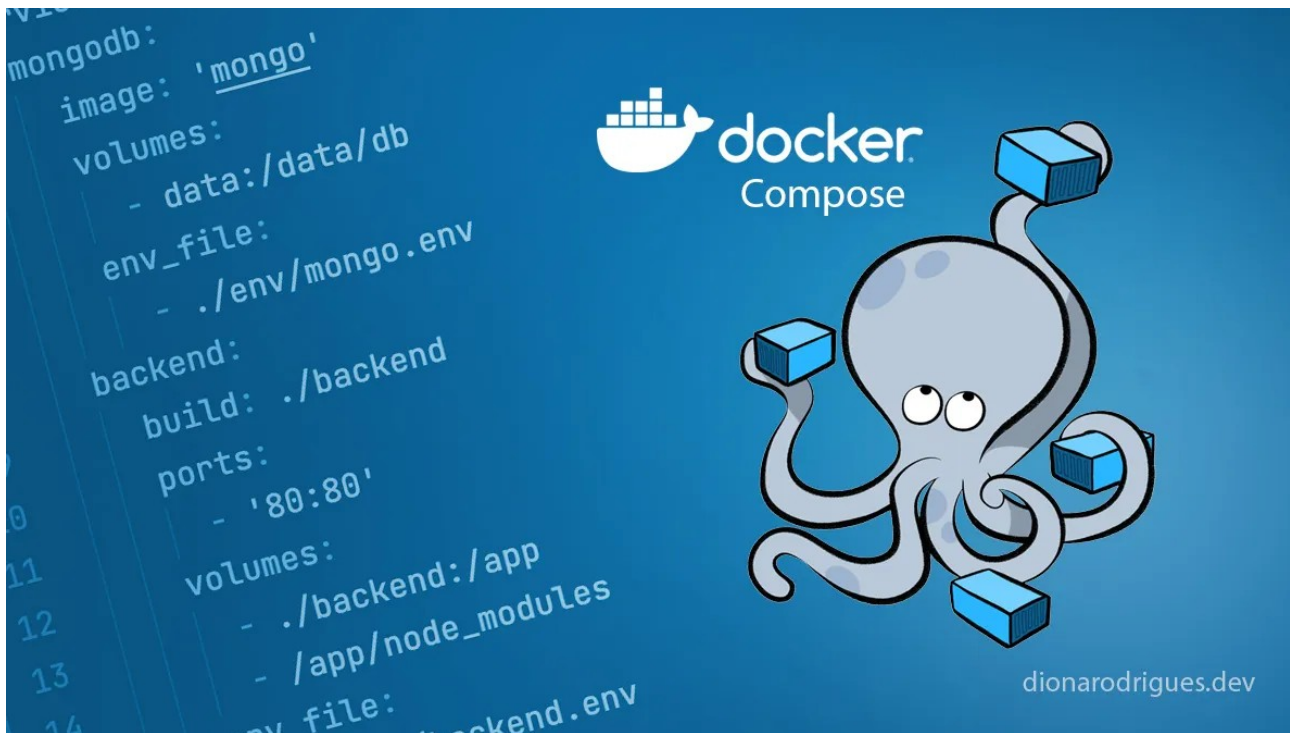


# Creació d'un entorn LAMP basat en contenidors



Nom i Cognom: Carles Molina Espinós

Curs: 2CFGS DAW

# Index

Objectius.....	3
Enunciat.....	3
Creació d'un entorn LAMP amb Docker.....	3
1. Descripció general.....	3
2. Fitxer docker-compose.yaml.....	3
phpmyadmin:: Aquest és el servei que executa phpMyAdmin, una aplicació web que permet gestionar bases de dades MySQL de manera gràfica.....	5
3. Accedir al phpMyAdmin.....	6
4. Creació de la base de dades i taula.....	6
4.1. Crear la taula amb SQL:.....	7
4.2. Inserta els registres.....	7
5. Codi PHP per llistar la taula.....	8
6. Instruccions per a la configuració i execució.....	9
Conclusions.....	9

# Objectius

- Utilitzar Docker compose per a gestionar múltiples contenidors.
- Desplegar una aplicació web dintre dels contenidors.
- Comunicar els contenidors entre si

## Enunciat

### Creació d'un entorn LAMP amb Docker

#### 1. Descripció general

Aquest document descriu els passos per configurar un entorn LAMP (Linux, Apache, MySQL i PHP) amb Docker, així com la creació d'una aplicació PHP que interactua amb una base de dades MySQL a través de PDO.

#### 2. Fitxer docker-compose.yml

Crearem un fitxer que configure els serveis necessaris per executar l'entorn LAMP amb Docker.

##### 1. Servei Apache-PHP:

```
apache-php:
  build: .
  container_name: apache-php
  volumes:
    - ./src:/var/www/html
  ports:
    - "8080:80"
  depends_on:
    - db
  networks:
    - lamp_network
```

- **apache-php:** Aquest és el nom del servei que s'executa amb Apache i PHP. El contenidor que executa aquest servei servirà l'aplicació PHP.
- **build: .:** Aquest paràmetre indica que Docker ha de construir una imatge basada en el Dockerfile que es troba al directori actual (.). Si no utilitzes un Dockerfile personalitzat, aquest paràmetre es pot substituir per una imatge prèvia.
- **container\_name: apache-php:** Aquesta línia assigna un nom al contenidor per identificar-lo fàcilment.

- **volumes:**
  - **./src:/var/www/html:** Aquesta línia vincula la carpeta local ./src a la carpeta /var/www/html dins del contenidor. Això significa que qualsevol arxiu PHP dins de ./src es copiarà a aquesta carpeta dins del contenidor i serà accessible per Apache.
- **ports:**
  - **"8080:80":** Exposar el port 80 del contenidor (per defecte en Apache) al port 8080 del teu ordinador local. Així, podràs accedir a la teva aplicació a través de http://localhost:8080.
- **depends\_on:**
  - **- db:** Això indica que el contenidor de apache-php depèn del contenidor db (la base de dades MySQL). Docker se n'assegurarà que el contenidor de la base de dades estigui en marxa abans que s'iniciï apache-php.
- **networks:**
  - **- lamp\_network:** Defineix que el contenidor s'integrarà a una xarxa anomenada lamp\_network per permetre la comunicació amb altres contenidors.

## 2. Servei de Base de Dades (MySQL):

```
db:
  image: mysql:5.7
  container_name: mysql-db
  environment:
    MYSQL_ROOT_PASSWORD: rootpassword
    MYSQL_DATABASE: db_proves
    MYSQL_USER: carles_proves
    MYSQL_PASSWORD: carles_proves
  ports:
    - "3306:3306"
  networks:
    - lamp_network
  volumes:
    - db_data:/var/lib/mysql
```

- **db:** Aquest és el servei de base de dades MySQL, utilitza la imatge oficial de MySQL.
- **image: mysql:5.7:** Utilitza la imatge de MySQL versió 5.7. Aquesta imatge conté la configuració i les dependències necessàries per executar MySQL.
- **container\_name: mysql-db:** Assigna un nom personalitzat al contenidor de MySQL per poder identificar-lo fàcilment.

- **environment:**
  - **MYSQL\_ROOT\_PASSWORD: rootpassword:** Assigna la contrasenya per al compte root de MySQL.
  - **MYSQL\_DATABASE: db\_proves:** Aquesta línia crea una base de dades anomenada db\_proves quan el contenidor es inicia.
  - **MYSQL\_USER: carles\_proves:** Assigna un usuari personalitzat anomenat carles\_proves.
  - **MYSQL\_PASSWORD: carles\_proves:** Assigna la contrasenya per a l'usuari carles\_proves.
- **ports:**
  - **"3306:3306":** Mapeja el port 3306 de MySQL al mateix port del teu ordinador local. Això permet que les aplicacions que s'executen al teu ordinador puguin connectar-se a la base de dades a través de localhost:3306.
- **networks:**
  - **- lamp\_network:** Indica que aquest contenidor es connectarà a la xarxa lamp\_network per poder comunicar-se amb altres contenidors com el de apache-php.
- **volumes:**
  - **db\_data:/var/lib/mysql:** Aquesta línia defineix un volum anomenat db\_data, que s'utilitza per emmagatzemar les dades de la base de dades. Això assegura que les dades de MySQL es mantinguin entre reinicis dels contenidors.

### 3.Servei de phpMyAdmin:

```
phpmyadmin:
image: phpmyadmin/phpmyadmin
container_name: phpmyadmin
environment:
  PMA_HOST: db
  PMA_USER: carles_proves
  PMA_PASSWORD: carles_proves
ports:
  - "8081:80"
depends_on:
  - db
networks:
  - lamp_network
```

- **phpmyadmin::** Aquest és el servei que executa phpMyAdmin, una aplicació web que permet gestionar bases de dades MySQL de manera gràfica.

- **image: phpmyadmin/phpmyadmin:** Utilitza la imatge oficial de phpMyAdmin per a Docker.
- **container\_name: phpmyadmin:** Assigna un nom personalitzat al contenidor de phpMyAdmin.
- **environment:**
  - **PMA\_HOST: db:** Defineix el nom del contenidor de la base de dades (en aquest cas, db) per a phpMyAdmin. Això permet a phpMyAdmin connectar-se a MySQL.
  - **PMA\_USER: carles\_proves:** Assigna l'usuari de la base de dades que es farà servir per a la connexió a MySQL a través de phpMyAdmin.
  - **PMA\_PASSWORD: carles\_proves:** Assigna la contrasenya de l'usuari de la base de dades per la connexió a MySQL.
- **ports:**
  - **"8081:80":** Mapeja el port 80 de phpMyAdmin al port 8081 del teu ordinador local. Això permet que puguis accedir a phpMyAdmin a través de <http://localhost:8081>.
- **depends\_on:**
  - **- db:** Aquesta línia indica que el contenidor phpmyadmin depèn del contenidor db (MySQL) i no es crearà fins que el contenidor de la base de dades estigui en marxa.
- **networks:**
  - **- lamp\_network:** Conecta phpMyAdmin a la mateixa xarxa per tal que pugui comunicar-se amb els altres contenidors (especialment amb la base de dades).

### 3. Accedir al phpMyAdmin

Per a accedir al phpMyAdmin la url es:

<http://localhost:8081>

### 4. Creació de la base de dades i taula

Per crear la taula, pots utilitzar la interfície de phpMyAdmin disponible a <http://localhost:8081> o executar un script SQL a través del terminal de MySQL.

## 4.1. Crear la taula amb SQL:

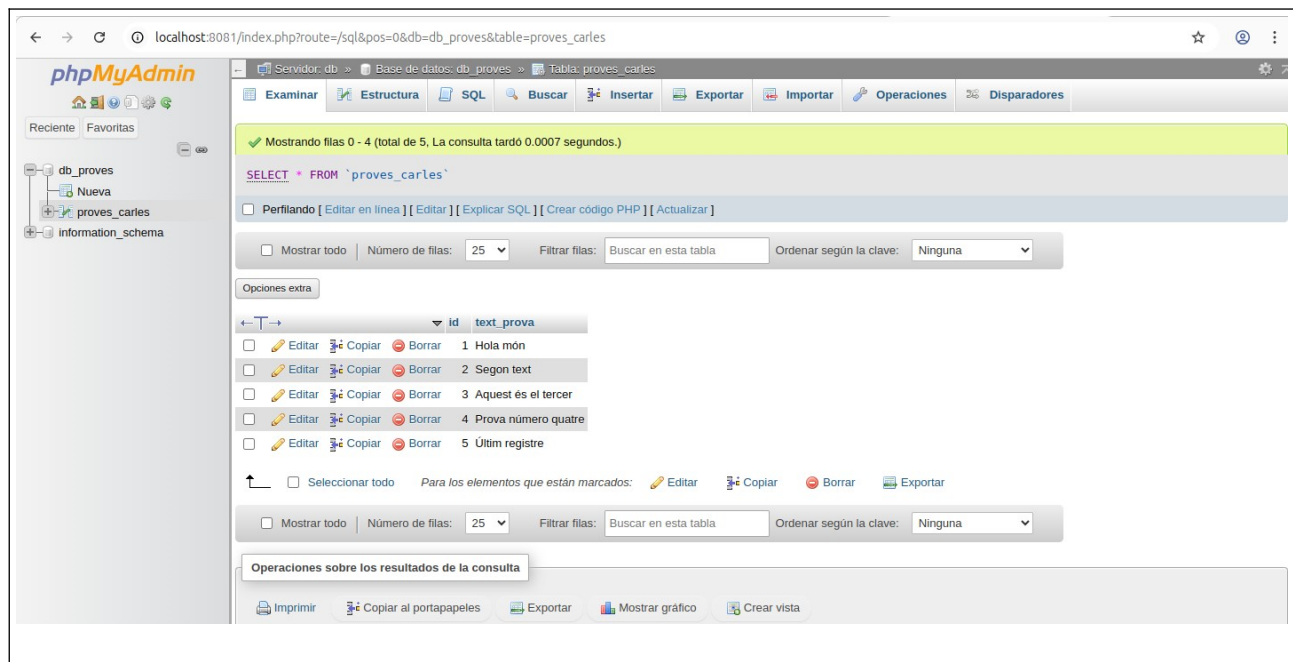
```
CREATE TABLE proves_carles (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  text VARCHAR(255) NOT NULL  
);
```

Estic creant una taula en phpMyAdmin que es diga proves\_carles.

## 4.2. Inserta els registres

```
INSERT INTO proves_carles (text_prova) VALUES  
(  
'Hola món',  
'Segon text',  
'Aquest és el tercer',  
'Prova número quatre',  
'Últim registre');
```

Insertar 5 registres como demana la activitat:



The screenshot shows the phpMyAdmin interface for a database named 'db\_proves'. The table 'proves\_carles' is selected, and the SQL query 'SELECT \* FROM `proves\_carles`' is executed. The results show 5 records with the following text values: 'Hola món', 'Segon text', 'Aquest és el tercer', 'Prova número quatre', and 'Últim registre'. The interface includes navigation tabs (Examinar, Estructura, SQL, etc.), a search bar, and various action buttons for each record (Editar, Copiar, Borrar).

	id	text_prova
<input type="checkbox"/>	1	Hola món
<input type="checkbox"/>	2	Segon text
<input type="checkbox"/>	3	Aquest és el tercer
<input type="checkbox"/>	4	Prova número quatre
<input type="checkbox"/>	5	Últim registre

## 5. Codi PHP per llistar la taula

Aquí tens el codi PHP que es connecta a la base de dades i llista els registres de la taula proves\_carles.

```
<?php
// Dades de connexió
$host = 'db'; // nom del servei MySQL al docker-compose
$db = 'db_proves';
$user = 'carles_proves';
$pass = 'carles_proves';
$charset = 'utf8mb4';

// DSN = Data Source Name
$dsn = "mysql:host=$host;dbname=$db;charset=$charset";

try {
    // Crear una connexió PDO
    $pdo = new PDO($dsn, $user, $pass);
    // Configurar PDO perquè llanci excepcions si hi ha errors
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Consulta SQL
    $stmt = $pdo->query("SELECT * FROM proves_carles");

    echo "<h1>Llistat de registres</h1>";
    echo "<ul>";
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        echo "<li>" . htmlspecialchars($row['text_prova']) . "</li>";
    }
    echo "</ul>";

} catch (PDOException $e) {
    echo "Error de connexió: " . $e->getMessage();
}
?>
```



## 6.Instruccions per a la configuració i execució

Crea un fitxer Dockerfile amb aquest contingut per afegir el mòdul PDO MySQL a la imatge de PHP:

```
FROM php:8.0-apache

# Instal·la el mòdul PDO per a MySQL
RUN docker-php-ext-install pdo pdo_mysql
```

Construeix i executa els contenidors Docker:

```
docker-compose up --build -d
```

Accedeix a la teua aplicació a través per el navegador per la url:

<http://localhost:8080>.



## Conclusions

- Docker Compose facilita la gestió de múltiples contenidors de manera senzilla i eficient, permetent definir tot l'entorn de desenvolupament amb un únic fitxer (docker-compose.yaml).
- Amb aquest projecte, hem pogut reproduir un entorn LAMP equivalent al que proporciona XAMPP, però amb la gran avantatge de ser portable, escalable i fàcilment desplegable en qualsevol màquina amb Docker.
- La utilització de volums ens ha permès mantenir les dades de MySQL persistents, evitant la pèrdua d'informació en cas de reinicis dels contenidors.