

Guía de Instalación para Proyecto PHP con Docker y Base de Datos

1. Requisitos Previos

Antes de comenzar, asegúrate de tener lo siguiente instalado en tu máquina:

- **Docker y Docker Compose.**
- **WSL (Windows Subsystem for Linux)** si estás utilizando Windows y la terminal de Visual Studio Code.
- **Proyecto PHP:** Ya tienes un proyecto PHP llamado `web_personal`.

2. Crear el Archivo `docker-compose.yml`

1. **Estructura del archivo `docker-compose.yml`:** En la raíz de tu proyecto PHP, crea un archivo llamado `docker-compose.yml` con el siguiente contenido:

```
version: '3.8'

services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8080:80"
    volumes:
      - ./web_personal:/var/www/html
    depends_on:
      - db
    networks:
      - webnet

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_ALLOW_EMPTY_PASSWORD: "yes" # Permite que MySQL funcione sin contraseña
      MYSQL_ROOT_PASSWORD: "" # No se establece una contraseña para el root
      MYSQL_DATABASE: project # Nombre de la base de datos
      MYSQL_USER: projectuser # Usuario de la base de datos sin contraseña
    ports:
      - "3306:3306"
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - webnet

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    ports:
      - "8081:80" # Accede a phpMyAdmin en http://localhost:8081
    environment:
      PMA_HOST: db # Nombre del servicio de la base de datos
      PMA_USER: projectuser # Usuario para acceder a la base de datos
      PMA_PASSWORD: "" # Contraseña vacía si no has configurado una
    depends_on:
      - db
    networks:
      - webnet

volumes:
  db_data:

networks:
  webnet:
    driver: bridge
```

Explicación:

- **php**: Contenedor con PHP y Apache, montando tu proyecto `web_personal` en `/var/www/html`.
- **mysql**: Contenedor con MySQL, configurado con las credenciales necesarias para tu base de datos.
- **depends_on**: Asegura que el contenedor de PHP espere a que MySQL esté listo antes de arrancar.

3. Crear la Base de Datos en Docker

- La base de datos se crea automáticamente cuando inicias el contenedor de MySQL gracias a la configuración en el archivo `docker-compose.yml`, donde especificamos `MYSQL_DATABASE: project`

4. Configuración de la Conexión en PHP (`config.php`)

Crea un archivo `config.php` en tu proyecto PHP para manejar la conexión a la base de datos. Asegúrate de tener los siguientes parámetros:

```
C:\Users\User\Documents\Asir 2\Aplicaciones Web\XAMPP\htdocs\docker-projects\Web-LAMP\web_personal > index.php
1 <?php
2 $servername = "db"; // Nombre del contenedor de la base de datos
3 $username = "projectuser"; // Usuario de la base de datos
4 $password = ""; // Sin contraseña
5 $dbname = "project"; // Nombre de la base de datos
6
7 // Crear la conexión
8 $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Verificar la conexión
11 if ($conn->connect_error) {
12     die("Conexión fallida: " . $conn->connect_error);
13 }
14 echo "Conexión exitosa";
15 ?>
16
17
```

- **Complicación**: Si hay problemas de conexión, asegúrate de que el contenedor de MySQL esté completamente inicializado antes de intentar la conexión. Puedes usar `depends_on` en `docker-compose.yml` para que PHP espere a MySQL.

5. Levantar los Contenedores

Con los archivos listos, abre una terminal en Visual Studio Code y navega a la carpeta de tu proyecto. Ejecuta el siguiente comando para iniciar los contenedores:

```
lec@DESKTOP-Q0N9833:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/Web-LAMP$ docker-compose up --build
```

Uso de docker build en el proyecto PHP

El comando `docker-compose up -d` no funcionó inicialmente porque no tenía instalada la herramienta necesaria para ejecutar `docker build`. Para solucionarlo, realicé la instalación de Docker Build utilizando los pasos indicados en la documentación oficial.

Después de instalarlo, construí la imagen del proyecto con:

`docker build -t web_personal .`

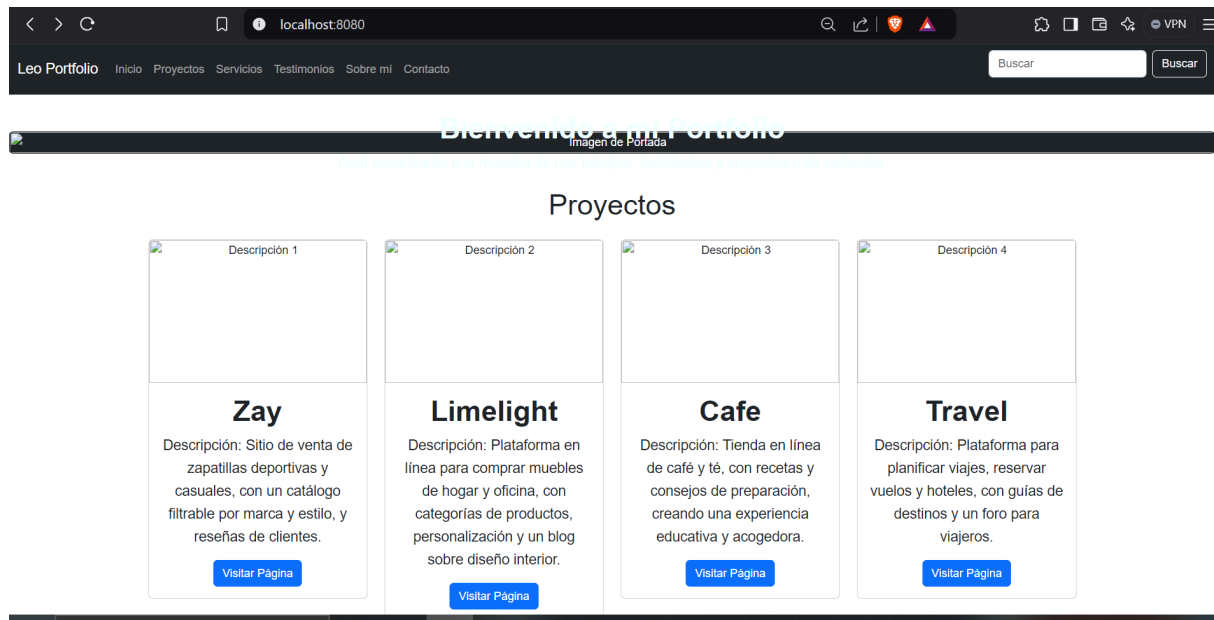
Esto creó la imagen personalizada del proyecto a partir del Dockerfile.

Complicación:

El error se debió a la falta de instalación de docker build, lo que retrasó el proceso. Una vez instalado, el problema se resolvió.

6. Acceder al Proyecto

Abre tu navegador y accede a <http://localhost:8080>. Aquí deberías ver tu proyecto PHP en ejecución.



7. Acceder a la Base de Datos MySQL

1. Creación de la Base de Datos en MySQL

Pasos:

Entra al contenedor MySQL:

```
docker exec -it web-lamp_db_1 bash
```

1.

Accede al cliente MySQL:

```
mysql -u root -p
```

```
leog@DESKTOP-Q8M9833:/mnt/c/Users/User/Documents/Asir 2/Aplicaciones Web/XAMPP/htdocs/docker-projects/Web-LAMP$ docker exec -it web-lamp_db_1 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Crea la base de datos:

```
CREATE DATABASE web_personal;
```

Crea una tabla para almacenar los contactos:

```
CREATE TABLE contactos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    phone VARCHAR(20),
    service VARCHAR(50),
    message TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. Conexión de PHP a la Base de Datos

Archivo `db.php` (Gestión de Conexión)

Crea un archivo llamado `db.php` en el directorio de tu proyecto con el siguiente contenido:

```
<?php
$servername = "db"; // Nombre del contenedor de la base de datos
$username = "projectuser"; // Usuario de la base de datos
$password = ""; // Sin contraseña
$dbname = "project"; // Nombre de la base de datos

// Crear la conexión
$conn = new mysqli($servername, $username, $password, $dbname);

// Verificar la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
echo "Conexión exitosa";
?>
```

3. Formulario de Contacto y Script de Inserción

Archivo `index.php/guardar_contacto`

Este archivo contiene el formulario y el código para guardar datos en la base de datos.

Código del Formulario:

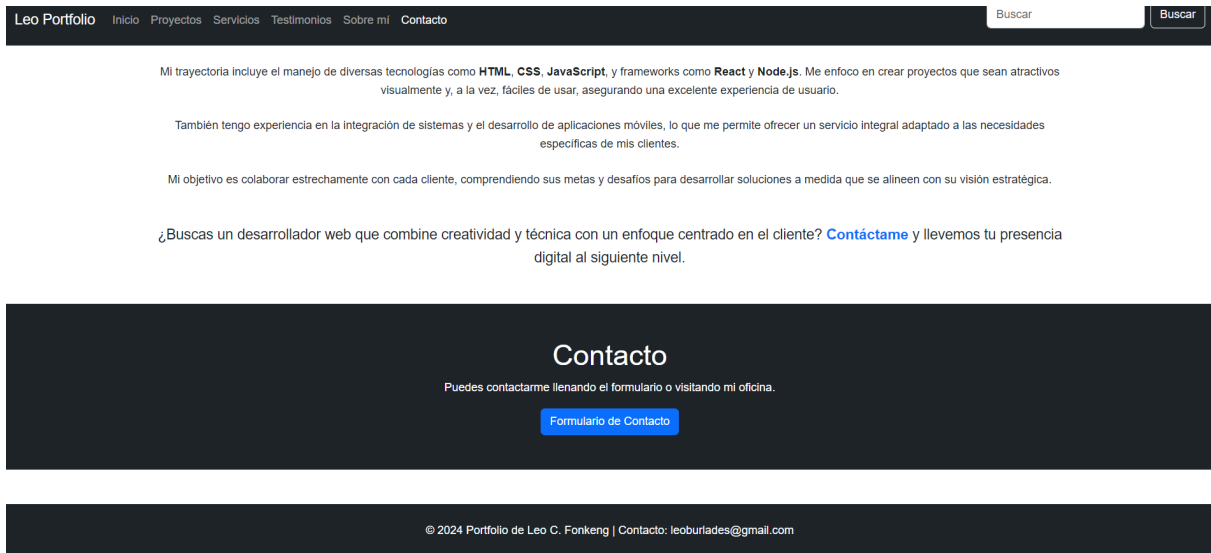
```

1  <?php
2  $servername = "db"; // Nombre del contenedor de la base de datos
3  $username = "projectuser"; // Usuario de la base de datos
4  $password = ""; // Sin contraseña
5  $dbname = "project"; // Nombre de la base de datos
6
7  // Crear la conexión
8  $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Verificar la conexión
11 ✓ if ($conn->connect_error) {
12     die("Conexión fallida: " . $conn->connect_error);
13 }
14
15 // Recoger los datos del formulario
16 $first_name = $_POST['first_name'];
17 $last_name = $_POST['last_name'];
18 $email = $_POST['email'];
19 $phone = $_POST['phone'];
20 $service = $_POST['service'];
21 $message = $_POST['message'];
22
23 // Preparar la consulta para insertar los datos en la base de datos
24 ✓ $sql = "INSERT INTO contactos (first_name, last_name, email, phone, service, message)
25     VALUES ('$first_name', '$last_name', '$email', '$phone', '$service', '$message')";
26
27 // Ejecutar la consulta
28 ✓ if ($conn->query($sql) === TRUE) {
29     echo "Mensaje enviado exitosamente";
30 } else {
31     echo "Error: " . $sql . "<br>" . $conn->error;
32 }
33
34 // Cerrar la conexión
35 $conn->close();
36 ?>
37

```

4. Pruebas

1. Accede a la aplicación en el navegador, por ejemplo,
<http://localhost:8000/index.php>



2. Completa el formulario y envíalo.

The image shows a modal form titled 'Formulario de Contacto' overlaid on the website. The form has a white background with a blue header and a blue 'Enviar Solicitud' button at the bottom. It contains the following fields: 'Nombre' (with value 'prueba1'), 'Apellido' (with value 'prueba'), 'Email' (with value 'prueba@prueba'), 'Teléfono' (with value '0v00000000'), 'Servicio de Interés' (a dropdown menu with 'Desarrollo Web' selected), and 'Mensaje' (with value 'prueba'). Below the message field is a checkbox labeled 'He leído y acepto la Política de Privacidad.' which is checked. The background of the website is visible through the semi-transparent modal.

3. Verifica que los datos se guardan correctamente en la tabla **contactos**.

Consulta en la Base de Datos:

Desde MySQL, puedes verificar los datos:

```
SELECT * FROM contactos;
```

phpMyAdmin

Reciente

Favoritas

information_schema

project

Nueva

contactos

Servidor db » Base de datos: project » Tabla: contactos

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Operaciones

Disparadores

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0008 segundos.)

SELECT * FROM `contactos`

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

first_name	last_name	email	phone	service	message
prueba1	prueba	prueba@prueba	0v00000000	web_development	prueba

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Operaciones sobre los resultados de la consulta

[Imprimir](#)

[Copiar al portapeles](#)

[Exportar](#)

[Mostrar gráfico](#)

[Crear vista](#)