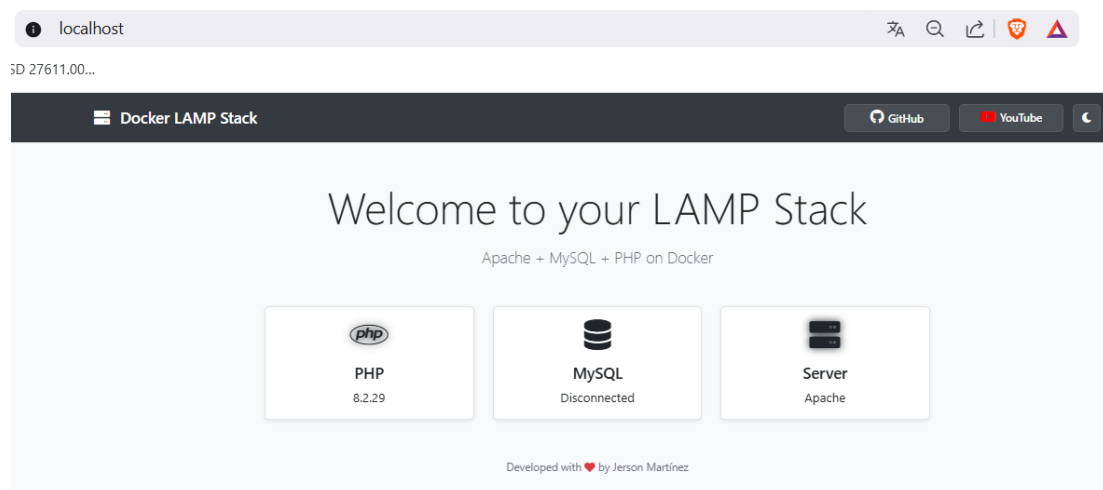


Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

Despliegue del aplicativo para gestión financiera en multi contenedor con Docker Compose

En esta actividad se utilizó el repositorio de [docker-lamp](https://github.com/Javier-P-C/Act2_Contenedores_UNIR_Docker_Compose) para levantar una aplicación con docker compose que tiene tres servicios: Una base de datos MySQL que tiene volumen persistente, servidor Apache con PHP y una interfaz web para gestionar el MySQL.

Para lanzarlo solo se tiene que usar el comando **docker compose up** e ir a <http://localhost> para acceder a la interfaz. En <http://localhost:8080> se encuentra PHPMysqlAdmin.



Repositorio:

https://github.com/Javier-P-C/Act2_Contenedores_UNIR_Docker_Compose

Docker

Compose:

```
1. version: "3.8"
2. services:
3.   db:
```

Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

```

4.     image: mysql:8.0
5.     restart: unless-stopped
6.     ports:
7.         - "127.0.0.1:${MYSQL_PORT}:3306"
8.     command:          --default-authentication-
plugin=mysql_native_password
9.     environment:
10.         MYSQL_DATABASE: ${MYSQL_DATABASE}
11.         MYSQL_USER: ${MYSQL_USER}
12.         MYSQL_PASSWORD: ${MYSQL_PASSWORD}
13.         MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
14.     volumes:
15.         - ./dump:/docker-entrypoint-initdb.d:ro
16.         - ./conf:/etc/mysql/conf.d:ro
17.         - persistent:/var/lib/mysql
18.     networks:
19.         - backend
20.     healthcheck:
21.         test: ["CMD", "mysqladmin", "ping", "-h",
"localhost", "-u", "root", "-p${MYSQL_ROOT_PASSWORD}"]
22.         interval: 10s
23.         timeout: 5s
24.         retries: 5
25.
26.     www:
27.     build:
28.         context: .
29.         args:
30.             PHP_MEMORY_LIMIT: ${PHP_MEMORY_LIMIT}
31.             PHP_MAX_EXECUTION_TIME:
32.                 ${PHP_MAX_EXECUTION_TIME}

```

Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

```

31.         PHP_UPLOAD_MAX_FILESIZE:
           ${PHP_UPLOAD_MAX_FILESIZE}
32.         PHP_POST_MAX_SIZE: ${PHP_POST_MAX_SIZE}
33.     restart: unless-stopped
34.     environment:
35.         MYSQL_HOST: ${MYSQL_HOST}
36.         MYSQL_DATABASE: ${MYSQL_DATABASE}
37.         MYSQL_USER: ${MYSQL_USER}
38.         MYSQL_PASSWORD: ${MYSQL_PASSWORD}
39.     ports:
40.         - "127.0.0.1:${APACHE_PORT}:80"
41.     volumes:
42.         - ./www:/var/www/html:ro
43.     depends_on:
44.         - db
45.     networks:
46.         - frontend
47.         - backend
48.     healthcheck:
49.         test: ["CMD", "curl", "-f",
           "http://localhost"]
50.         interval: 10s
51.         timeout: 5s
52.         retries: 5

53.     phpmyadmin:
54.         image: phpmyadmin/phpmyadmin:latest
55.         restart: unless-stopped
56.         environment:
57.             PMA_HOST: ${PMA_HOST}
58.             PMA_USER: ${PMA_USER}
59.             PMA_PASSWORD: ${PMA_PASSWORD}

```

Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

```

60.         UPLOAD_LIMIT: ${PHP_UPLOAD_MAX_FILESIZE}
61.     ports:
62.         - "127.0.0.1:${PHPMYADMIN_PORT}:80"
63.     depends_on:
64.         - db
65.     networks:
66.         - frontend
67.         - backend
68.     healthcheck:
69.         test: ["CMD", "curl", "-f",
70.             "http://localhost"]
71.         interval: 10s
72.         timeout: 5s
73.         retries: 5
74.     networks:
75.         frontend:
76.             driver: bridge
77.         backend:
78.             driver: bridge
79.             internal: true
80.     volumes:
81.         persistent:

```

1. Se especifica la versión del Docker Compose
2. Se inicia el bloque donde se definen los servicios
- 3-24. Se define la base de datos MySQL que se desplegará
4. Usa la imagen MySQL 8.0
5. Reinicia el contenedor al menos que sea detenido manualmente
7. Mapea el puerto 3306 del contenedor a un puerto del localhost

Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

8. Configura la autenticación de MySQL

9-13. Se declaran las variables de ambiente

14-17. Se establecen los volúmenes que usará el contenedor

15. Pone el script de la creación de la base de datos que se encuentra en dump/ en el directorio /docker-entrypoint-initdb.d como readonly

16. Pone la configuración que se encuentra en ./conf en /etc/mysql/conf.d para configurar MySQL

17. Hace que los datos que se guarden en /var/lib/mysql dentro del contenedor se vuelvan un volumen persistente

18-19: Se conecta a la red interna backend

20-24. Establece un healthcheck que le hace un ping a mysql

25-52. Se configura en la interfaz web

26-32. Se crea un contenedor con el Dockerfile del proyecto, para ellos se le pasa el contexto y argumentos

33. Reinicia el contenedor al menos que sea detenido manualmente

34-38. Se declaran las variables de ambiente

40. Mapea el puerto 80 del contenedor a un puerto del localhost

41-42. Establece la carpeta del host ./www como un bind mount

43-44. Indica que debe esperar a que el servicio db esté arriba para crearse

45-47. Se conecta a las redes internas backend y frontend

48-52. Se configura un healthcheck que hace un curl periódico a localhost

53-72. Se configura el servicio phpadmin

54. Usa la imagen de phpadmin latest

55. Reinicia el contenedor al menos que sea detenido manualmente

56-60. Se declaran las variables de ambiente

61-62. Mapea el puerto 80 del contenedor a un puerto del localhost

63-64. Indica que debe esperar a que el servicio db esté arriba para crearse

65-67. Se conecta a las redes internas backend y frontend

68-72. Se configura un healthcheck que hace un curl periódico a localhost

Asignatura	Datos del alumno	Fecha
Contenedores	Apellidos: Piña Camacho	25/10/2025
	Nombre: Javier	

73-77. Se crean las redes internas que usarán los contenedores

79-80. Se crea el volumen persistente para la base de datos