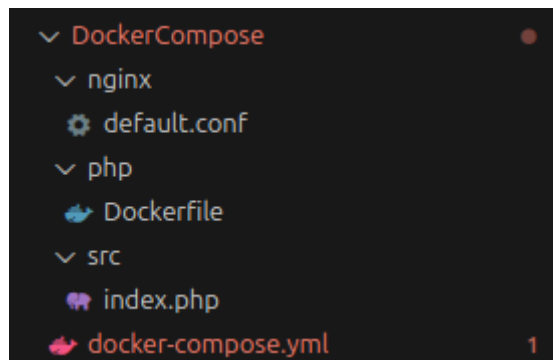


Objetivo

El objetivo de esta práctica es crear y gestionar un entorno de desarrollo multi-servicio utilizando Docker Compose, permitiendo que los distintos servicios (una aplicación PHP, un servidor Nginx y una base de datos MySQL) trabajen de forma coordinada. Además, se incluye una interfaz gráfica de administración de la base de datos con phpMyAdmin.

Estructura del proyecto

El proyecto se organiza en varias carpetas y archivos que representan cada uno de los servicios y sus configuraciones:



| | |
|----------------------|--|
| — nginx/ | |
| — default.conf | # Configuración personalizada de Nginx |
| — php/ | |
| — Dockerfile | # Imagen personalizada de PHP con mysqli |
| — src/ | |
| — index.php | # Código fuente de la aplicación PHP |
| — docker-compose.yml | # Archivo principal de orquestación |

Servicios definidos en docker-compose.yml

Se definieron tres servicios:

1. Nginx (Servidor web)

- Imagen: `nginx:latest`
- Funciona como servidor frontal, atendiendo las peticiones HTTP.
- Está configurado para reenviar peticiones PHP al contenedor `php`.
- Se expone al puerto `8080` del host.
- Usa una configuración personalizada (`default.conf`).

2. PHP (PHP-FPM)

- Imagen personalizada basada en `php:7.4-fpm`, definida con un `Dockerfile`.
- Se encarga de procesar el código PHP recibido desde Nginx.
- Incluye la extensión `mysqli` para conectar con MySQL.

3. MySQL (Base de datos)

- Imagen: `mysql:5.7`
- Contenedor persistente mediante un volumen (`db_data`).
- Configurado con variables de entorno para establecer una base de datos y credenciales de acceso.
- Se comunica internamente con el contenedor PHP.

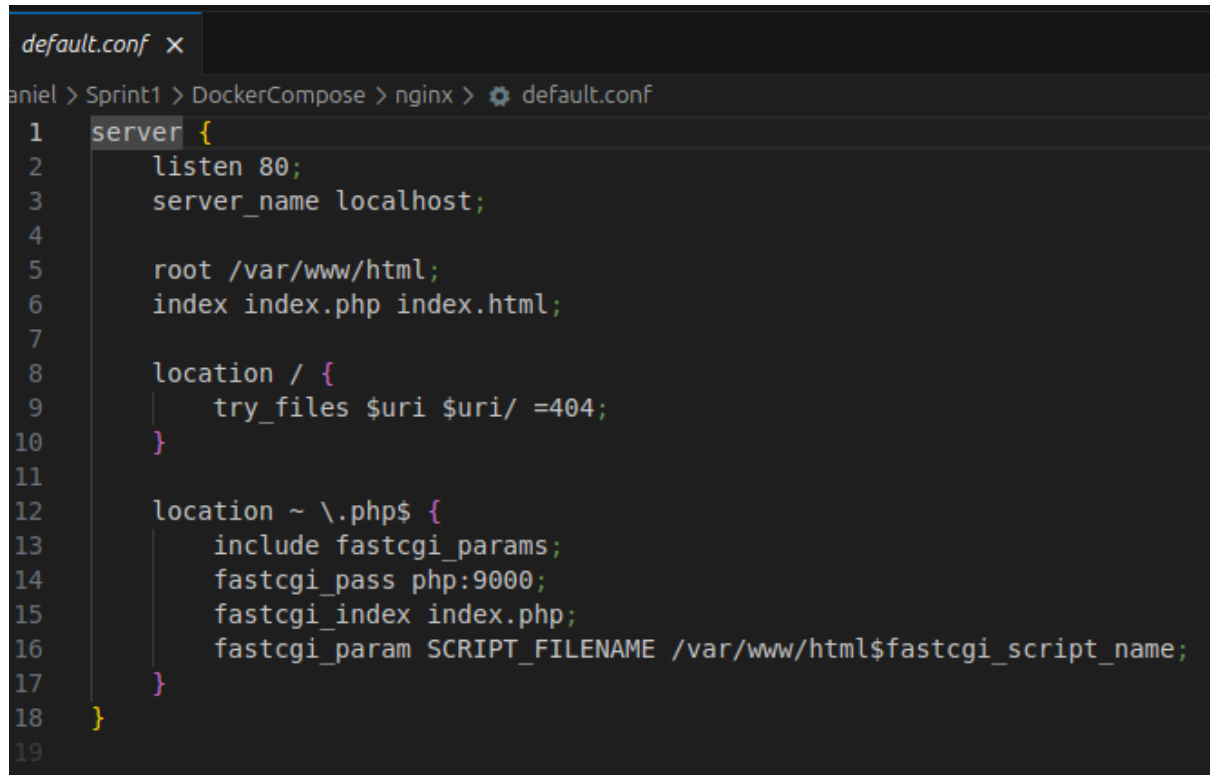
```
docker-compose.yml ~/LocalStack 1  docker-compose.yml ~/./DockerCompose 1 X
Daniel > Sprint1 > DockerCompose > docker-compose.yml > version
docker-compose.yml - The Compose specification establishes a standard for the definition of multi-container platform-agnostic applications (compose-spec.json)
1 version: '3.8'
2
3 services:
4   nginx:
5     image: nginx:latest
6     container_name: nginx
7     ports:
8       - "8080:80"
9     volumes:
10      - ./src:/var/www/html
11      - ./nginx/default.conf:/etc/nginx/conf.d/default.conf
12     depends_on:
13       - php
14     networks:
15       - app-network
16
17   php:
18     build: ./php
19     container_name: php
20     volumes:
21       - ./src:/var/www/html
22     networks:
23       - app-network
24
25   db:
26     image: mysql:5.7
27     container_name: mysql
28     restart: always
29     environment:
30       MYSQL_ROOT_PASSWORD: rootpass
31       MYSQL_DATABASE: app_db
32       MYSQL_USER: user
33       MYSQL_PASSWORD: userpass
34     volumes:
35       - db_data:/var/lib/mysql
36     networks:
37       - app-network
38
39 volumes:
40   db_data:
41
42 networks:
43   app-network:
44     driver: bridge
45
```

Se utilizan volúmenes para persistencia de datos. Esto asegura que los datos de la base de datos se mantengan incluso si el contenedor se reinicia o elimina.

Todos los servicios están conectados mediante una red bridge personalizada llamada **app-network**, lo que permite que se comuniquen usando sus nombres de contenedor como hostnames.

Configuración personalizada de Nginx

En el archivo `nginx/default.conf`, se define cómo manejar los archivos PHP:



```
1 server {
2     listen 80;
3     server_name localhost;
4
5     root /var/www/html;
6     index index.php index.html;
7
8     location / {
9         try_files $uri $uri/ =404;
10    }
11
12    location ~ /\.php$ {
13        include fastcgi_params;
14        fastcgi_pass php:9000;
15        fastcgi_index index.php;
16        fastcgi_param SCRIPT_FILENAME /var/www/html$fastcgi_script_name;
17    }
18 }
19
```

Esto permite que Nginx reenvíe peticiones PHP al contenedor `php` correctamente.

Dockerfile de php

El contenedor de PHP no se utiliza directamente desde una imagen oficial sin cambios, sino que construimos una imagen personalizada mediante un `Dockerfile`. Esto nos permite añadir extensiones específicas o realizar configuraciones necesarias para nuestra aplicación.

```
Dockerfile x
daniel > Sprint1 > DockerCompose > php > Dockerfile > FROM
1 FROM php:7.4-fpm
2 RUN docker-php-ext-install mysqli
3
```

1. **FROM php:7.4-fpm**

Se basa en la imagen oficial de PHP en modo FPM (FastCGI Process Manager), que es ideal para ser usado con servidores web como Nginx.

2. **RUN docker-php-ext-install mysqli**

Instala la extensión **mysqli**, necesaria para que PHP pueda conectarse a una base de datos MySQL.

Sin esta línea, cualquier intento de conexión MySQL desde PHP fallaría porque la función `new mysqli()` no estaría disponible.

¿Por qué no usar directamente php:7.4-fpm sin Dockerfile?

Porque la imagen base de PHP no incluye todas las extensiones por defecto. Al construir una imagen personalizada, tenemos el control de qué funcionalidades queremos incluir, en este caso, la extensión **mysqli**.

Index.php

En el archivo `src/index.php`, se realiza una conexión a la base de datos para verificar que todo el sistema funciona correctamente:

```
index.php x
daniel > Sprint1 > DockerCompose > src > index.php
1  <?php
2  $mysqli = new mysqli("db", "user", "userpass", "app_db");
3
4  if ($mysqli->connect_error) {
5      die("Conexión fallida: " . $mysqli->connect_error);
6  }
7
8  echo "Conexión exitosa a la base de datos!";
9  ?>
10
```

Ejecución de la aplicación

Para levantar el entorno, se ejecuta el siguiente comando desde el directorio del proyecto:

```
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# ls
docker-compose.yml  nginx  php  src
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose up -d --build
Creating network "dockercompose_app-network" with driver "bridge"
Creating volume "dockercompose_db_data" with default driver
Building php
Step 1/2 : FROM php:7.4-fpm
7.4-fpm: Pulling from library/php
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Pull complete
fb5a4c8af82f: Pull complete
972155ae644b: Pull complete
a8e3b94fe6c1: Pull complete
93346a3f46bc: Pull complete
b922b67ca46b: Pull complete
6137f893bda6: Pull complete
79b1a1b78461: Pull complete
Digest: sha256:3ac7c8c74b2b047c7cb273469d74fc0d59b857aa44043e6ea6a0084372811d5b
Status: Downloaded newer image for php:7.4-fpm
--> 38f2b691dcb8
Step 2/2 : RUN docker-php-ext-install mysqli
--> Running in 147116a4c307
Configuring for:
PHP Api Version:      20190902
Zend Module Api No:   20190902
Zend Extension Api No: 320190902
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for cc... cc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether cc accepts -g... yes
checking for cc option to accept ISO C89... none needed
```

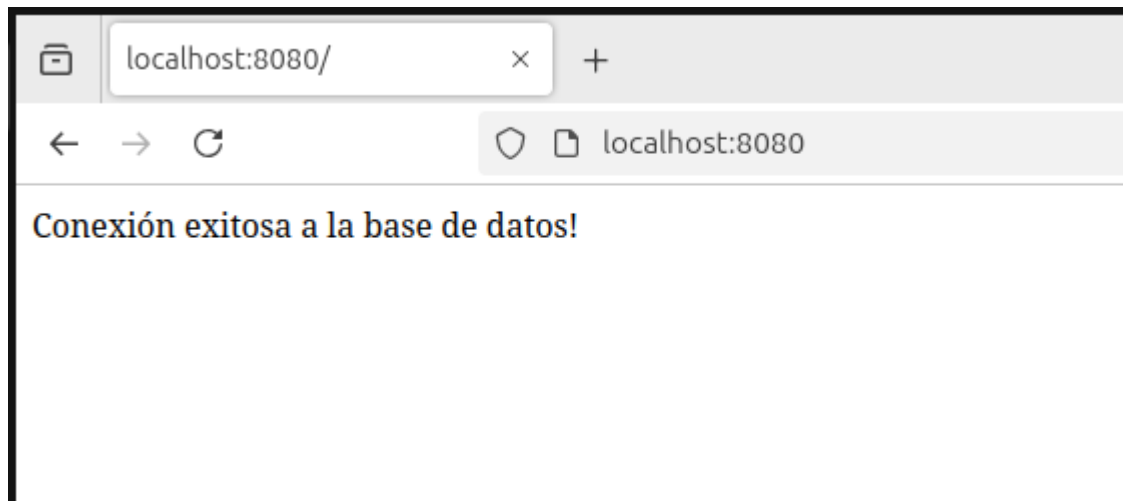
```
Creating php ... done
Creating mysql ... done
Creating nginx ... done
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker ps
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|-------------------|--------------------------|----------------|---------------|---|-------|
| fc88e64913ba | nginx:latest | "/docker-entrypoint..." | 10 minutes ago | Up 10 minutes | 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp | nginx |
| 3ae5b4fb1157 | mysql:5.7 | "docker-entrypoint.s..." | 10 minutes ago | Up 10 minutes | 3306/tcp, 33060/tcp | mysql |
| 21196cdf854b | dockercompose php | "docker-php-entrypoi..." | 10 minutes ago | Up 10 minutes | 9000/tcp | php |

```
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose#
```

Una vez iniciado:

- La aplicación PHP está disponible en: <http://localhost:8080>



Comprobaciones adicionales

Para comprobar que todo funciona correctamente, vamos a comprobar los logs de las tres aplicaciones que hemos implementado:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose logs nginx
Attaching to nginx
nginx /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx 10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
nginx /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
nginx /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx /docker-entrypoint.sh: Configuration complete; ready for start up
nginx 2025/06/16 06:41:25 [notice] 1#1: using the "epoll" event method
nginx 2025/06/16 06:41:25 [notice] 1#1: nginx/1.27.5
nginx 2025/06/16 06:41:25 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
nginx 2025/06/16 06:41:25 [notice] 1#1: OS: Linux 6.11.0-26-generic
nginx 2025/06/16 06:41:25 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker processes
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 28
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 29
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 30
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 31
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 32
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 33
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 34
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 35
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 36
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 37
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 38
nginx 2025/06/16 06:41:25 [notice] 1#1: start worker process 39
nginx 172.21.0.1 - - [16/Jun/2025:06:47:37 +0000] "GET / HTTP/1.1" 200 48 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0" "-"
nginx 172.21.0.1 - - [16/Jun/2025:06:47:37 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://localhost:8080/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0" "-"
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose#
```

```
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose logs php
Attaching to php
php [16-Jun-2025 06:41:25] NOTICE: fpm is running, pid 1
php [16-Jun-2025 06:41:25] NOTICE: ready to handle connections
php 172.21.0.4 - [16/Jun/2025:06:47:37 +0000] "GET /index.php 200
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose logs db
Attaching to mysql
mysql 2025-06-16 06:41:25+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.44-1.el7 started.
mysql 2025-06-16 06:41:25+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql 2025-06-16 06:41:25+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.44-1.el7 started.
mysql 2025-06-16 06:41:25+00:00 [Note] [Entrypoint]: Initializing database files
mysql 2025-06-16 06:41:25.4205842 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
mysql 2025-06-16 06:41:25.5643382 0 [Warning] InnoDB: New log files created, LSN=45790
mysql 2025-06-16 06:41:25.5894342 0 [Warning] InnoDB: Creating foreign key constraint system tables.
mysql 2025-06-16 06:41:25.6444192 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: ec92f311-4a7c-11f0-be2b-064037199e10.
mysql 2025-06-16 06:41:25.6462882 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
mysql 2025-06-16 06:41:25.6885912 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
mysql 2025-06-16 06:41:25.6885912 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
mysql 2025-06-16 06:41:25.6886642 0 [Warning] CA certificate ca.pem is self signed.
mysql 2025-06-16 06:41:25.6884122 1 [Warning] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
mysql 2025-06-16 06:41:27.6272022 0 [Note] [Entrypoint]: Database files initialized
mysql 2025-06-16 06:41:27.6382862 0 [Note] [Entrypoint]: Starting temporary server
mysql 2025-06-16 06:41:27.6401612 0 [Note] [Entrypoint]: Waiting for server startup
mysql 2025-06-16 06:41:27.6401722 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
mysql 2025-06-16 06:41:27.6401742 0 [Note] InnoDB: Uses event mutexes
mysql 2025-06-16 06:41:27.6401762 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
mysql 2025-06-16 06:41:27.6401802 0 [Note] InnoDB: Compressed tables use zlib 1.2.13
mysql 2025-06-16 06:41:27.6401822 0 [Note] InnoDB: Using Linux native AIO
mysql 2025-06-16 06:41:27.6403402 0 [Note] InnoDB: Number of pools: 1
mysql 2025-06-16 06:41:27.6404102 0 [Note] InnoDB: Using CPU crc32 instructions
mysql 2025-06-16 06:41:27.6414052 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
mysql 2025-06-16 06:41:27.6456372 0 [Note] InnoDB: completed initialization of buffer pool
mysql 2025-06-16 06:41:27.6460132 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of setpriority().
mysql 2025-06-16 06:41:27.6484542 0 [Note] InnoDB: Highest supported file format is Barracuda.
mysql 2025-06-16 06:41:27.6671492 0 [Note] InnoDB: Creating shared tablespace for temporary tables
mysql 2025-06-16 06:41:27.6671872 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
mysql 2025-06-16 06:41:27.6811802 0 [Note] InnoDB: file './ibtmp1' size is now 12 MB.
mysql 2025-06-16 06:41:27.6815532 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
mysql 2025-06-16 06:41:27.6815562 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
mysql 2025-06-16 06:41:27.6818442 0 [Note] InnoDB: 5.7.44 started; log sequence number 2768291
mysql 2025-06-16 06:41:27.6819842 0 [Note] InnoDB: Loading buffer pool(s) from ./var/lib/mysql/ib_buffer_pool
mysql 2025-06-16 06:41:27.6820532 0 [Note] Plugin 'FEDERATED' is disabled.
mysql 2025-06-16 06:41:27.6833852 0 [Note] InnoDB: Buffer pool(s) load completed at 250616 6:41:27
mysql 2025-06-16 06:41:27.6862192 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them.
mysql 2025-06-16 06:41:27.6862262 0 [Note] Skipping generation of SSL certificates as certificate files are present in data directory.
mysql 2025-06-16 06:41:27.6862282 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
mysql 2025-06-16 06:41:27.6862302 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
mysql 2025-06-16 06:41:27.6866222 0 [Warning] CA certificate ca.pem is self signed.
mysql 2025-06-16 06:41:27.6866702 0 [Note] Skipping generation of RSA key pair as key files are present in data directory.
mysql 2025-06-16 06:41:27.6903372 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
mysql 2025-06-16 06:41:27.6998522 0 [Note] Event Scheduler: Loaded 0 events
mysql 2025-06-16 06:41:27.6991792 0 [Note] mysqld: ready for connections.
mysql Version: 5.7.44 socket: /var/run/mysqld/mysqld.sock port: 0 MySQL Community Server (GPL)
mysql 2025-06-16 06:41:28+00:00 [Note] [Entrypoint]: Temporary server started.
mysql /var/lib/mysql/mysql.sock' -> /var/run/mysqld/mysqld.sock'
mysql 2025-06-16 06:41:28.5208962 3 [Note] InnoDB: Stopping purge
mysql 2025-06-16 06:41:28.5145392 3 [Note] InnoDB: Resuming purge
mysql 2025-06-16 06:41:28.5155132 3 [Note] InnoDB: Stopping purge
mysql 2025-06-16 06:41:28.5172402 3 [Note] InnoDB: Resuming purge
mysql 2025-06-16 06:41:28.5182492 3 [Note] InnoDB: Stopping purge
mysql 2025-06-16 06:41:28.5208962 3 [Note] InnoDB: Resuming purge
mysql 2025-06-16 06:41:28.5218562 3 [Note] InnoDB: Stopping purge
mysql 2025-06-16 06:41:28.5228962 3 [Note] InnoDB: Resuming purge
mysql Warning: Unable to load './usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
mysql Warning: Unable to load './usr/share/zoneinfo/leapseconds.tab' as time zone. Skipping it.
mysql Warning: Unable to load './usr/share/zoneinfo/ztdata.zi' as time zone. Skipping it.
mysql Warning: Unable to load './usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
mysql Warning: Unable to load './usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
mysql 2025-06-16 06:41:29+00:00 [Note] [Entrypoint]: Creating database app_db
mysql 2025-06-16 06:41:29+00:00 [Note] [Entrypoint]: Creating user user
mysql 2025-06-16 06:41:29+00:00 [Note] [Entrypoint]: Giving user user access to schema app_db
```

Y a mayores, comprobamos que podemos entrar al contenedor de PHP y probar conexión a MySQL manualmente:

```
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker exec -it php bash
root@21196cdf854b:/var/www/html#
```


Phpmyadmin

A mayores, vamos a añadir phpmyadmin a los servicios de docker-compose para poder gestionar la base de datos en un entorno gráfico.

```
phpmyadmin:
  image: phpmyadmin/phpmyadmin
  container_name: phpmyadmin
  restart: always
  ports:
    - "8081:80"
  environment:
    PMA_HOST: db
    PMA_USER: user
    PMA_PASSWORD: userpass
  depends_on:
    - db
  networks:
    - app-network
```

Imagen: `phpmyadmin/phpmyadmin`

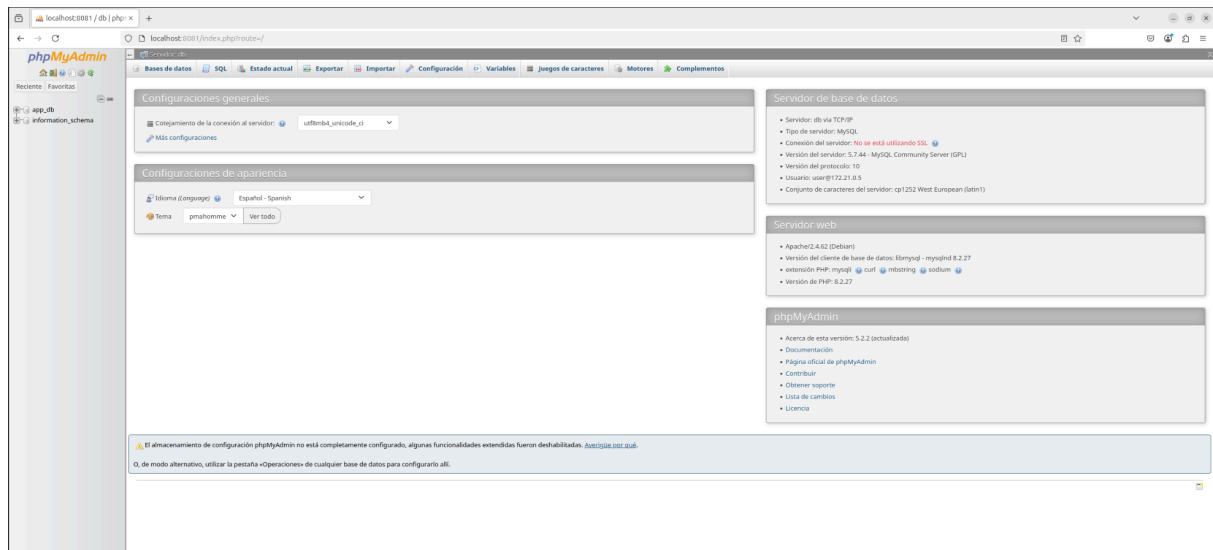
Proporciona una interfaz web para gestionar MySQL.

Se expone en el puerto `8081`.

Se conecta al contenedor `db`.

```
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker-compose up -d --build
Building php
Step 1/2 : FROM php:7.4-fpm
--> 38f2b691dcb8
Step 2/2 : RUN docker-php-ext-install mysqli
--> Using cache
--> a41b301e6b4b
Successfully built a41b301e6b4b
Successfully tagged dockercompose_php:latest
Pulling phpmyadmin (phpmyadmin/phpmyadmin):...
latest: Pulling from phpmyadmin/phpmyadmin
af302e5c37e9: Pull complete
71a74e0b3aeb: Pull complete
3ef0d0774deb: Pull complete
11d17388a3b8: Pull complete
0814cbbf72a2: Pull complete
3a28acedadff8: Pull complete
2ab7ef40feaf: Pull complete
80324ccb20a1: Pull complete
ad5f2fca9132: Pull complete
9df2a6231627: Pull complete
b3207e60ff9a: Pull complete
d18c9f420b35: Pull complete
073faad72ba8: Pull complete
4f4fb700ef54: Pull complete
a5c74661bb0e: Pull complete
1cf5cbfd971f: Pull complete
e92d8472eb26: Pull complete
7753344c0dda: Pull complete
b0f9d0503cef: Pull complete
2ee0fe041682: Pull complete
Digest: sha256:95e01f723b5e5fabf16d0473f1df2354c4c6352b35902b51d6aa245e074aee4
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
mysql is up-to-date
php is up-to-date
nginx is up-to-date
Creating phpmyadmin ... done
root@daniel-MS-7E28:/home/daniel/Sprint1/DockerCompose# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                NAMES
7a7b6aa437c8   phpmyadmin/phpmyadmin              "/docker-entrypoint..." 19 seconds ago Up 18 seconds 0.0.0.0:8081->80/tcp, [::]:8081->80/tcp phpmyadmin
fc88e64913ba   nginx:latest                       "/docker-entrypoint..." 35 minutes ago Up 35 minutes 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp nginx
5ae5b4f01157   mysql:5.7                          "docker-entrypoint.s..." 35 minutes ago Up 35 minutes 3306/tcp, 33060/tcp mysql
21196cdf854b   dockercompose_php                  "docker-php-entrypoi..." 35 minutes ago Up 35 minutes 9080/tcp php
```

phpMyAdmin está disponible en: <http://localhost:8081>



Conclusión

Esta práctica demuestra cómo utilizar **Docker Compose** para orquestar múltiples servicios de forma eficiente, permitiendo el desarrollo y pruebas de aplicaciones web en entornos reproducibles y aislados. Se configuraron redes internas, persistencia de datos, dependencias entre servicios y una interfaz de administración amigable, logrando así un entorno de desarrollo completo y funcional.