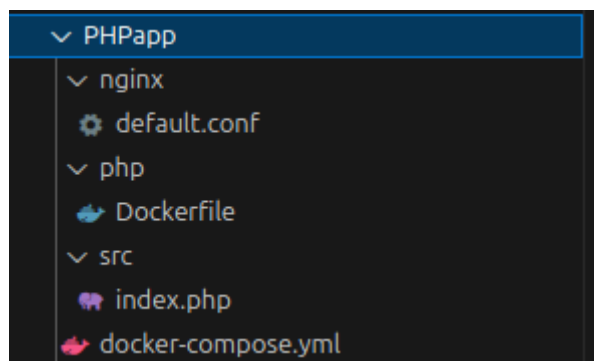


Objetivo

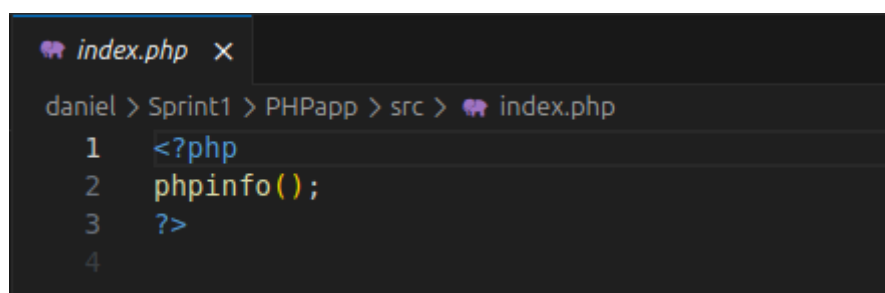
Configurar y desplegar una aplicación web mínima en PHP utilizando **Docker** y **Docker Compose**, combinando **PHP-FPM** y **Nginx** en contenedores separados conectados por una red interna.

Crear la estructura del proyecto

Crearemos la estructura del proyecto utilizando los siguientes archivos y carpetas:



Implementar el contenido de los archivos



Es una **aplicación PHP mínima**.

Muestra información del entorno PHP, útil para verificar que PHP y Nginx están funcionando correctamente.

```
default.conf x
daniel > Sprint1 > PHPapp > nginx > default.conf
1  server {
2      listen 80;
3      server_name localhost;
4
5      root /var/www/html;
6      index index.php index.html index.htm;
7
8      location / {
9          try_files $uri $uri/ =404;
10     }
11
12     location ~ /\.php$ {
13         include fastcgi_params;
14         fastcgi_pass php-fpm:9000;
15         fastcgi_index index.php;
16         fastcgi_param SCRIPT_FILENAME /var/www/html$fastcgi_script_name;
17     }
18
19     location ~ /\.ht {
20         deny all;
21     }
22 }
```

Es el archivo de configuración de **Nginx**.

Define:

- El puerto por donde escucha (**80**).
- La **raíz del sitio** (**/var/www/html**).
- Cómo manejar peticiones a archivos PHP:
 - Las redirige a PHP-FPM (**php-fpm:9000**) mediante FastCGI.
- Reglas básicas de seguridad (por ejemplo, negar acceso a **.ht***).

```
Dockerfile x
daniel > Sprint1 > PHPapp > php > Dockerfile > FROM
1 FROM php:8.2-fpm
2
3 WORKDIR /var/www/html
4
5
```

Crea una imagen basada en **PHP 8.2 con FPM** (FastCGI Process Manager), que es necesaria para que Nginx procese archivos PHP.

Establece `/var/www/html` como el directorio de trabajo, donde se ubicará el código fuente de la aplicación.

```
docker-compose.yml 1 x
daniel > Sprint1 > PHPapp > docker-compose.yml > {} networks > {} app-network
docker-compose.yml - The Compose specification establishes a standard for the definition of multi-con
1 version: '3.8'
2
3 services:
4   php-fpm:
5     build:
6       context: ./php
7     container_name: php-fpm
8     volumes:
9       - ./src:/var/www/html
10    networks:
11      - app-network
12
13   nginx:
14     image: nginx:latest
15     container_name: nginx
16     ports:
17       - "8080:80"
18     volumes:
19       - ./src:/var/www/html
20       - ./nginx/default.conf:/etc/nginx/conf.d/default.conf
21     depends_on:
22       - php-fpm
23     networks:
24       - app-network
25
26 networks:
27   app-network:
28     driver: bridge
```

Orquesta los contenedores usando **Docker Compose**.

Define dos servicios:

1. **php-fpm**:

- Construido desde el Dockerfile de **php/**.
- Procesa archivos PHP.

2. **nginx**:

- Usa una imagen oficial de Nginx.
- Sirve la aplicación y pasa las peticiones PHP a **php-fpm**.

Ambos comparten:

- El código fuente (**./src**) mediante volúmenes.
- Una **red interna app-network** que les permite comunicarse entre sí de forma segura.

Despliegue y comprobación

Iniciamos los contenedores con el siguiente comando:


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

root@daniel-MS-7E28:/home/daniel/Sprint1/PHPapp# docker-compose up -d --build
Building php-fpm
Step 1/2 : FROM php:8.2-fpm
8.2-fpm: Pulling from library/php
Digest: sha256:be60d05e522b390d2ece5f713a803bf50356763373ba1995a3edc940b986ae59
Status: Downloaded newer image for php:8.2-fpm
--> d70cbacd9e9
Step 2/2 : WORKDIR /var/www/html
--> Using cache
--> 753cad9e889d

Successfully built 753cad9e889d
Successfully tagged phpapp_php-fpm:latest
Pulling nginx (nginx:latest)...
latest: Pulling from library/nginx
dadd67da3f26b: Already exists
3b08567da964: Pull complete
5db01cfa947d: Pull complete
1bc5dc8b475d: Pull complete
979e6233a40a: Pull complete
d2a7ba8dbfee: Pull complete
32e44235e1d5: Pull complete
Digest: sha256:6784fb0834aa7dbbe12e3d7471e69c290df3e6ba810dc38b34ae33d3c1c05f7d
Status: Downloaded newer image for nginx:latest
php-fpm is up-to-date
Creating nginx ... done
root@daniel-MS-7E28:/home/daniel/Sprint1/PHPapp# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
28a99abe57b9   nginx:latest   "/docker-entrypoint. ..."  7 seconds ago Up 5 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   nginx
b8e6d0145b10   phpapp_php-fpm "docker-php-entrypoi ..."  2 minutes ago Up 32 seconds  9000/tcp                               php-fpm
```

<http://localhost:8080>

Como podemos ver, se muestra la página de `phpinfo()` correctamente renderizada por Nginx a través de PHP-FPM.

PHP Version 8.2.28	
	
System	Linux f8ad01d0-6110-6110-26-generic #29~24.04.1 Ubuntu SMP PREEMPT_DYNAMIC Thu Apr 17 19:20:47 UTC 2 68.64
Build Date	Jun 10 2025 23:32:12
Build System	Linux - Docker
Configure Command	<code>https://github.com/docker-library/php</code> `./configure --build-id=64-linux-gnu --with-config-file-path=/usr/local/etc/php --with-config-file-dir=/usr/local/etc/php/conf.d --enable-gzip-checking=fatal --with-mysql --with-pdo --enable-mbstring --enable-mcrypt --with-password-openssl --with-sodium=shared --with-yaml=libyaml --with-zip --with-curl --with-iconv --with-openssl --with-readline --with-zlib --disable-pharconfig --with-pspell --with-bcmath=lib,64-linux-gnu --disable-gmp --enable-fpm --with-fpm-user=www-data --with-fpm-group=www-data build_alias=x86_64-linux-gnu`
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini Files parsed	/usr/local/etc/php/conf.d/docker-fpm.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	APM420220829.NTS
PHP Extension Build	APM20220829.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
OpenSSL Support	disabled
Registered PHP Streams	tcp, udp, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udps, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, default