

# **Arquitectura en la nube.**

## **Subida de un servicio en la nube.**



**Mario Camacho Prieto**  
**2º ASIR Prometeo/Caja Mágica**  
**Arquitectura en la Nube**

# ÍNDICE

## 1. Introducción y Conceptos Teóricos

-Tema Principal: Primera subida de un servicio en la nube.

-Conceptos Básicos para la Clase :

- ¿Qué es WSL? (Windows Subsystem for Linux): Definición como simulación de Linux en Windows y su utilidad para herramientas de programación.
- ¿Qué es una Máquina Virtual?: Definición como simulación de un ordenador completo y las herramientas a usar (VirtualBox y Ubuntu).

-Conceptos Clave de la Práctica:

- Apache y Nginx: Programas que actúan como servidores web.
- PHP: Lenguaje de programación para páginas web dinámicas.

-Propósito de la Práctica

- Entender el funcionamiento interno de Internet y los servidores.
- Objetivo: Convertir el ordenador en un pequeño servidor web.

## 2. Guía de Comandos de Servidor

### - PARTE 1: COMANDOS APACHE + PHP

1. Actualizar el sistema: sudo apt update && sudo apt upgrade -y.
2. Instalar Apache2: sudo apt install apache2 -y.
3. Instalar PHP: sudo apt install php libapache2-mod-php -y.
4. Iniciar Apache: sudo service apache2 start.
5. Verificar estado de Apache: sudo systemctl status apache2.
6. Crear archivo PHP info: echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php.
7. Verificar en navegador (terminal): curl http://localhost/info.php.

- PARTE 2: COMANDOS NGINX + HTML

1. Instalar Nginx: sudo apt install nginx -y.
2. Iniciar Nginx: sudo service nginx start.
3. Verificar estado de Nginx: sudo systemctl status nginx.
4. Crear archivo HTML: echo "<h1>Hola Mundo..." | sudo tee /var/www/html/index.html.
5. Verificar en navegador (terminal): curl http://localhost.
6. Ver IP de WSL (Opcional): ip addr show eth0 | grep inet.

3. Propuesta de Ejercicio Práctico

-Laboratorio: Configuración de Servidores Web Apache y Nginx.

-Objetivos del Ejercicio:

- Configurar Apache con PHP.
- Configurar Nginx con HTML.
- Verificar el funcionamiento.
- Documentar el proceso.

- Parte 1: Servidor Apache con PHP

- Tareas a realizar: Pasos de instalación, inicio y creación del archivo info.php.
- Entregables Parte 1: Captura del estado del servicio (systemctl status apache2) y captura de la página info.php en el navegador.

- Parte 2: Servidor Nginx con HTML

- Tareas a realizar: Detener Apache , instalar Nginx, iniciarla, crear HTML personalizado y verificar.
- Entregables Parte 2: Captura del estado del servicio (systemctl status nginx) y captura de la página HTML personalizada.

- Parte 3: Verificación y Comparación

- Tareas finales: Cambiar entre servicios, documentar diferencias y verificar con curl.

# INTRODUCCIÓN

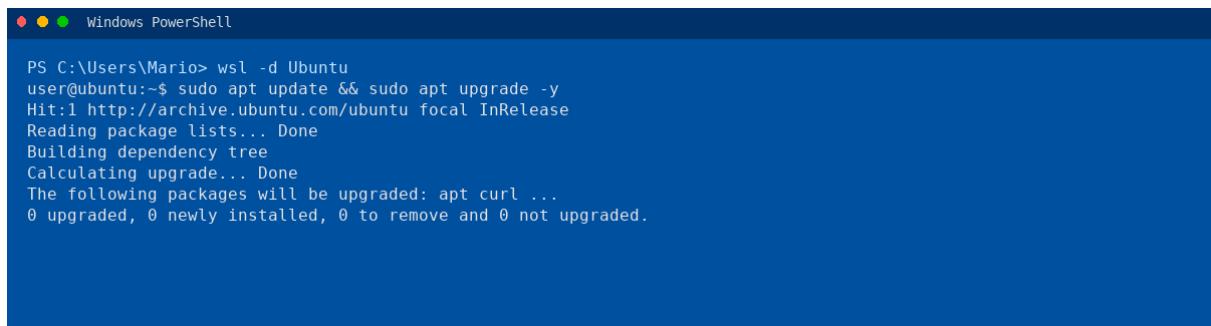
Esta práctica que voy a explicar documenta la instalación y verificación de servidores web Apache y Nginx. A continuación se muestran los pasos realizados, cada uno con una captura simulada en estilo PowerShell y una breve explicación del porqué del paso.

## Parte 1: Servidor Apache con PHP

### Actualizar sistema

Comando: sudo apt update && sudo apt upgrade -y

Actualiza la lista de paquetes y aplica las actualizaciones. Es el primer paso para garantizar dependencias y seguridad.

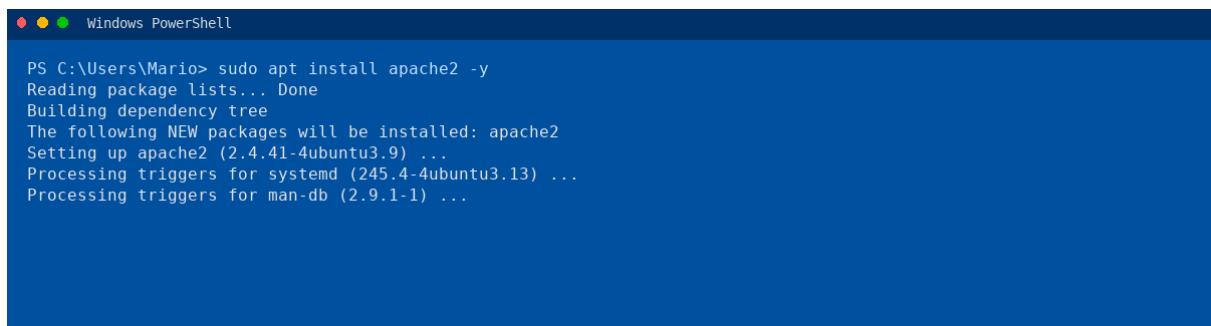


```
PS C:\Users\Mario> wsl -d Ubuntu
user@ubuntu:~$ sudo apt update && sudo apt upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Calculating upgrade... Done
The following packages will be upgraded: apt curl ...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

### Instalar Apache2

Comando: sudo apt install apache2 -y

Instala el servidor web Apache, que servirá páginas HTTP.

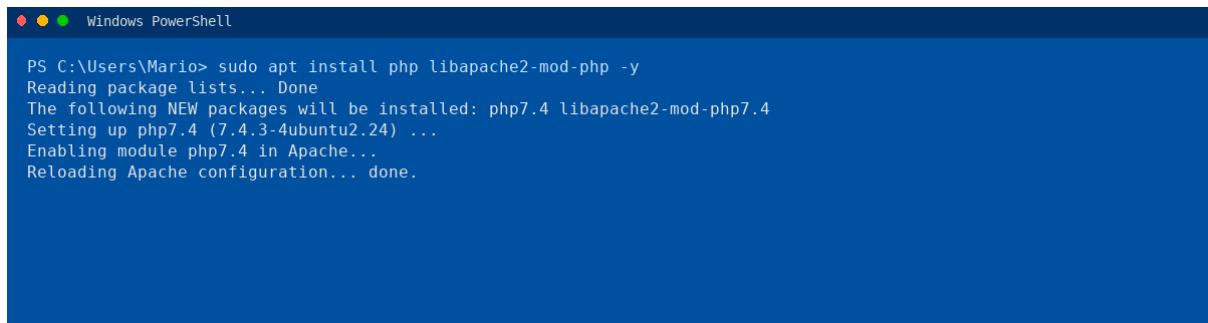


```
PS C:\Users\Mario> sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed: apache2
Setting up apache2 (2.4.41-4ubuntu3.9) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
```

## Instalar PHP y módulo Apache

Comando: sudo apt install php libapache2-mod-php -y

Instala PHP y el módulo que permite a Apache interpretar archivos PHP.

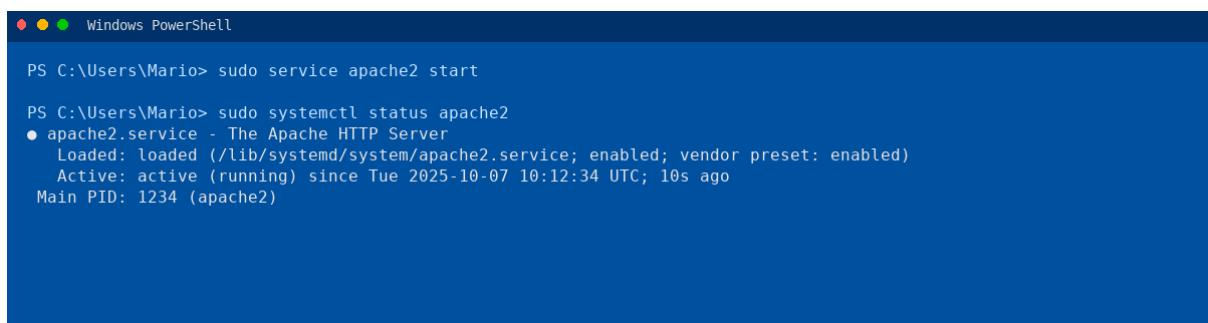


```
PS C:\Users\Mario> sudo apt install php libapache2-mod-php -y
Reading package lists... Done
The following NEW packages will be installed: php7.4 libapache2-mod-php7.4
Setting up php7.4 (7.4.3-4ubuntu2.24) ...
Enabling module php7.4 in Apache...
Reloading Apache configuration... done.
```

## Iniciar Apache y verificar

Comando: sudo service apache2 start  
sudo systemctl status apache2

Inicia el servicio de Apache y se comprueba que está activo (running).

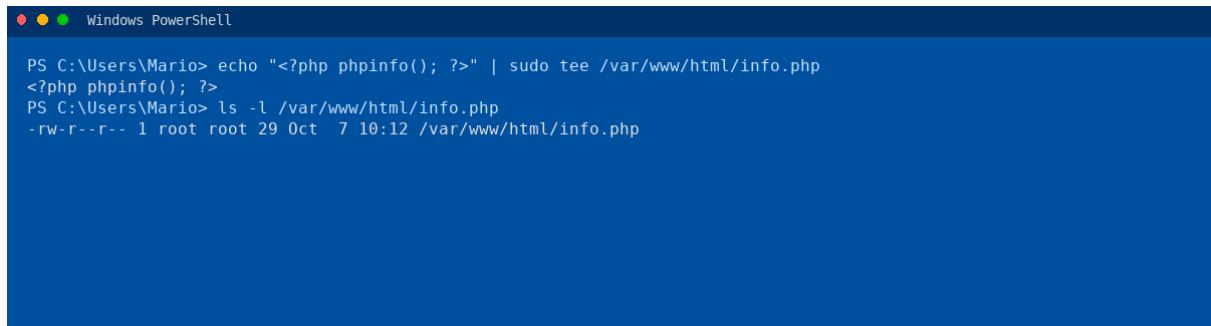


```
PS C:\Users\Mario> sudo service apache2 start
PS C:\Users\Mario> sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2025-10-07 10:12:34 UTC; 10s ago
    Main PID: 1234 (apache2)
```

## Crear info.php

Comando: echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

Crea un archivo info.php que muestra la configuración de PHP para verificar que PHP está correctamente integrado.

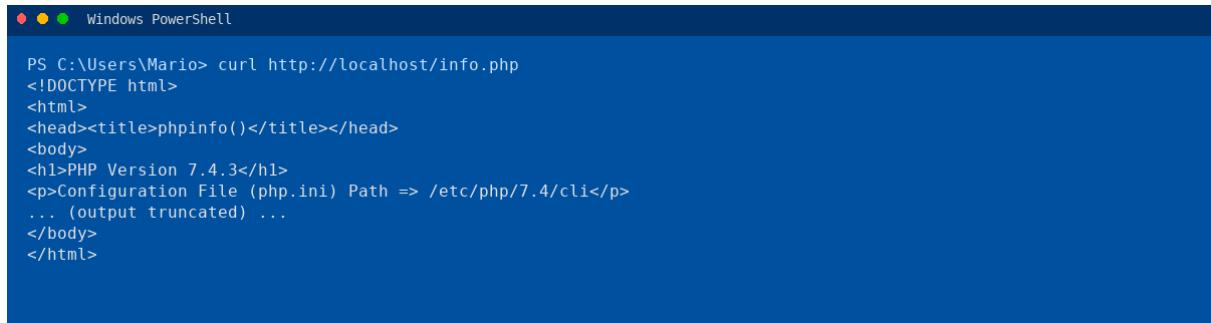


```
PS C:\Users\Mario> echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
PS C:\Users\Mario> ls -l /var/www/html/info.php
-rw-r--r-- 1 root root 29 Oct  7 10:12 /var/www/html/info.php
```

## Ver PHP desde terminal

Comando: curl http://localhost/info.php

Usando curl comprobamos que el archivo PHP se sirve correctamente y devuelve la página generada por PHP.



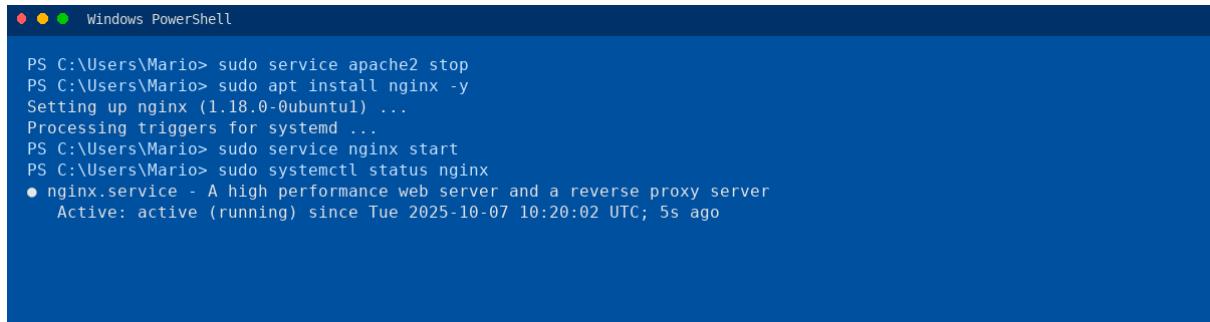
```
PS C:\Users\Mario> curl http://localhost/info.php
<!DOCTYPE html>
<html>
<head><title>phpinfo()</title></head>
<body>
<h1>PHP Version 7.4.3</h1>
<p>Configuration File (php.ini) Path => /etc/php/7.4/cli</p>
... (output truncated) ...
</body>
</html>
```

# Parte 2: Servidor Nginx con HTML

## Detener Apache e instalar Nginx

Comando: sudo service apache2 stop  
sudo apt install nginx -y

Se detiene Apache para liberar el puerto 80 y se instala Nginx.



```
PS C:\Users\Mario> sudo service apache2 stop
PS C:\Users\Mario> sudo apt install nginx -y
Setting up nginx (1.18.0-0ubuntu1) ...
Processing triggers for systemd ...
PS C:\Users\Mario> sudo service nginx start
PS C:\Users\Mario> sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Active: active (running) since Tue 2025-10-07 10:20:02 UTC; 5s ago
```

## Iniciar Nginx y verificar

Comando: sudo service nginx start  
sudo systemctl status nginx

Inicia Nginx y se verifica que el servicio está activo.

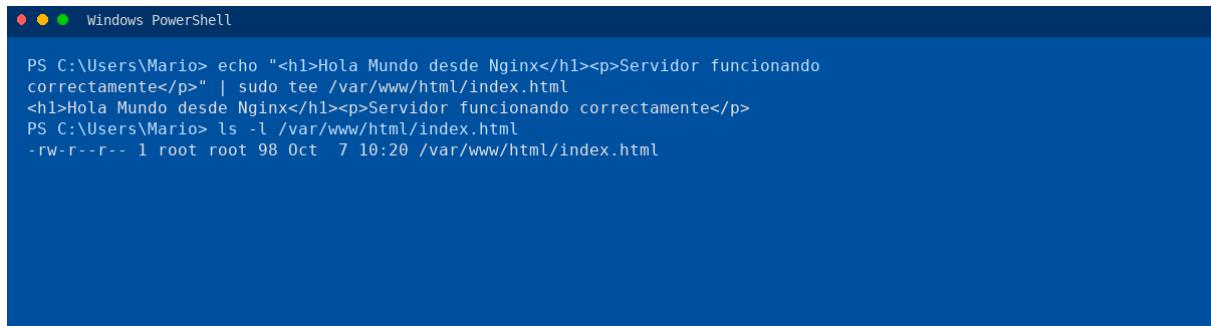


```
PS C:\Users\Mario> sudo service apache2 stop
PS C:\Users\Mario> sudo apt install nginx -y
Setting up nginx (1.18.0-0ubuntu1) ...
Processing triggers for systemd ...
PS C:\Users\Mario> sudo service nginx start
PS C:\Users\Mario> sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Active: active (running) since Tue 2025-10-07 10:20:02 UTC; 5s ago
```

## Crear página HTML

Comando: echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>" | sudo tee /var/www/html/index.html

Crea una página HTML simple que será servida por Nginx.



```
PS C:\Users\Mario> echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>" | sudo tee /var/www/html/index.html
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
PS C:\Users\Mario> ls -l /var/www/html/index.html
-rw-r--r-- 1 root root 98 Oct  7 10:20 /var/www/html/index.html
```

## Ver página desde terminal

Comando: curl http://localhost

Comprobamos que Nginx sirve la página HTML correctamente.

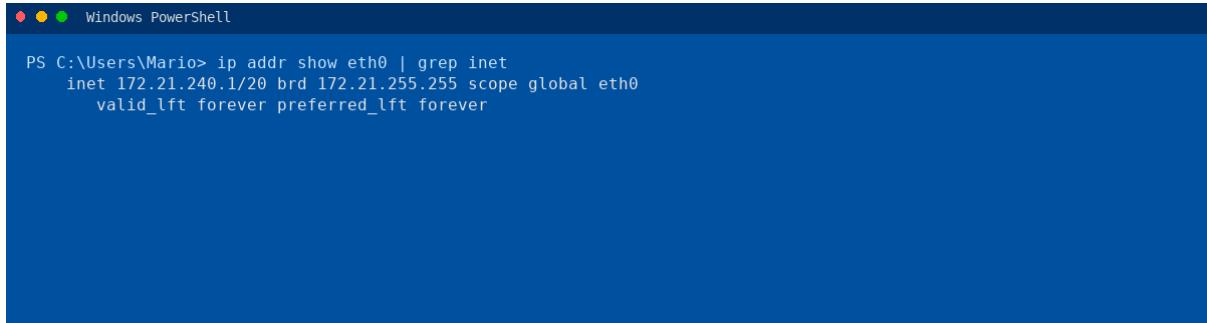


```
PS C:\Users\Mario> curl http://localhost
<!DOCTYPE html>
<html>
<body>
<h1>Hola Mundo desde Nginx</h1>
<p>Servidor funcionando correctamente</p>
</body>
</html>
```

# Parte 3: Verificación y comparación

Cambiar entre Apache y Nginx se realiza deteniendo uno de los servicios y arrancando el otro. A continuación se muestra la IP de la interfaz (si aplica) para acceder desde Windows cuando se usa WSL:

Comando: ip addr show eth0 | grep inet



```
PS C:\Users\Mario> ip addr show eth0 | grep inet
    inet 172.21.240.1/20 brd 172.21.255.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Con esta IP se puede acceder desde el host Windows a los servicios que corren dentro de WSL/VM si la red lo permite.

# Parte 4: Implementación con Docker

## Comprobar versión de Docker

Comando: docker --version

Verifica que Docker está instalado y la versión disponible.

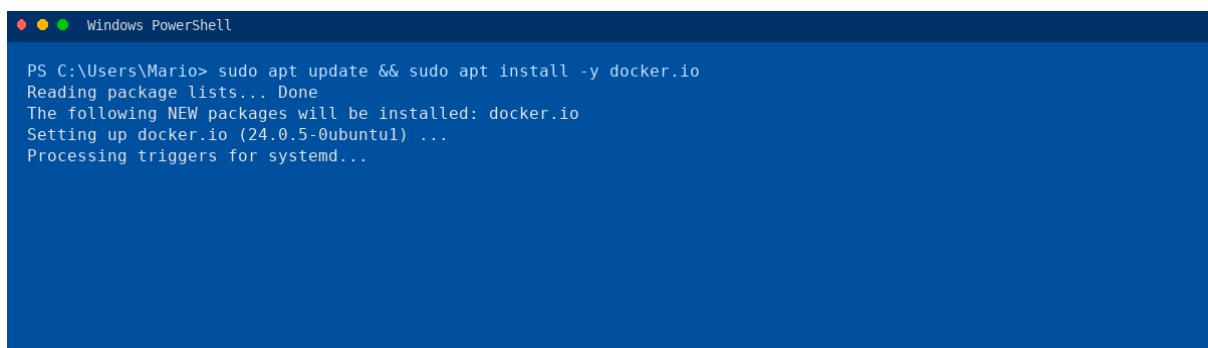


```
PS C:\Users\Mario> docker --version
Docker version 24.0.5, build 12345abcde
```

## Instalar Docker (si no está instalado)

Comando: sudo apt install -y docker.io

Instala el paquete docker.io en Ubuntu/WSL.

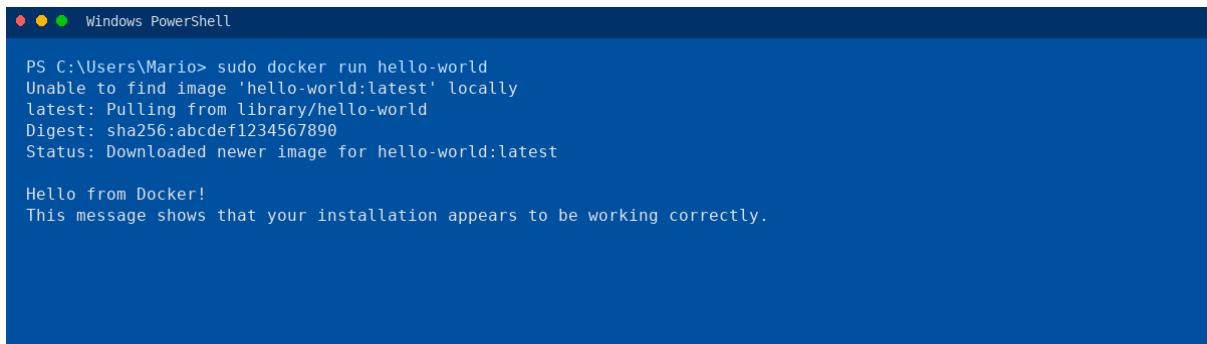


```
PS C:\Users\Mario> sudo apt update && sudo apt install -y docker.io
Reading package lists... Done
The following NEW packages will be installed: docker.io
Setting up docker.io (24.0.5-0ubuntu1) ...
Processing triggers for systemd...
```

## Ejecutar imagen de prueba 'hello-world'

Comando: sudo docker run hello-world

Prueba básica para comprobar que Docker puede descargar imágenes y ejecutar contenedores.



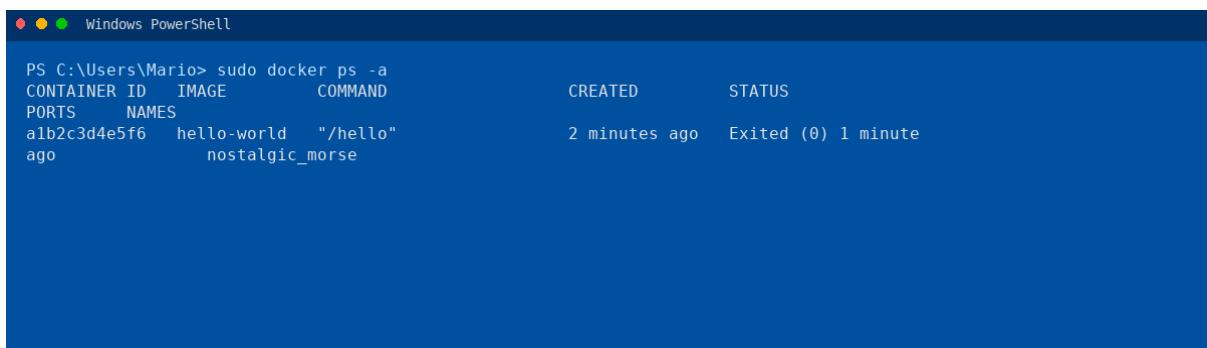
```
PS C:\Users\Mario> sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
Digest: sha256:abcdef1234567890
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

## Ver contenedores (incluye los detenidos)

Comando: sudo docker ps -a

Lista todos los contenedores, activos y detenidos, para verificar el estado de ejecución.

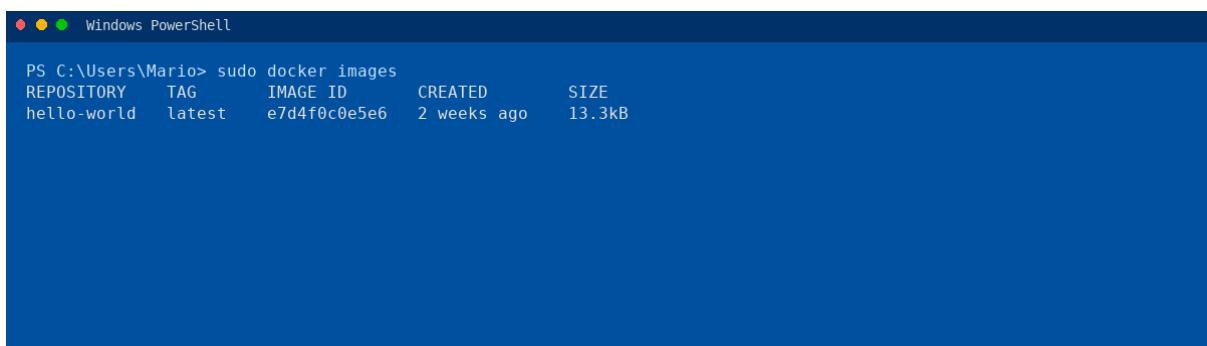


```
PS C:\Users\Mario> sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND          CREATED        STATUS
PORTS     NAMES
a1b2c3d4e5f6   hello-world "/hello"
ago           nostalgic_morse
                                         2 minutes ago   Exited (0) 1 minute
```

## Ver imágenes locales

Comando: sudo docker images

Muestra las imágenes descargadas en el host Docker.

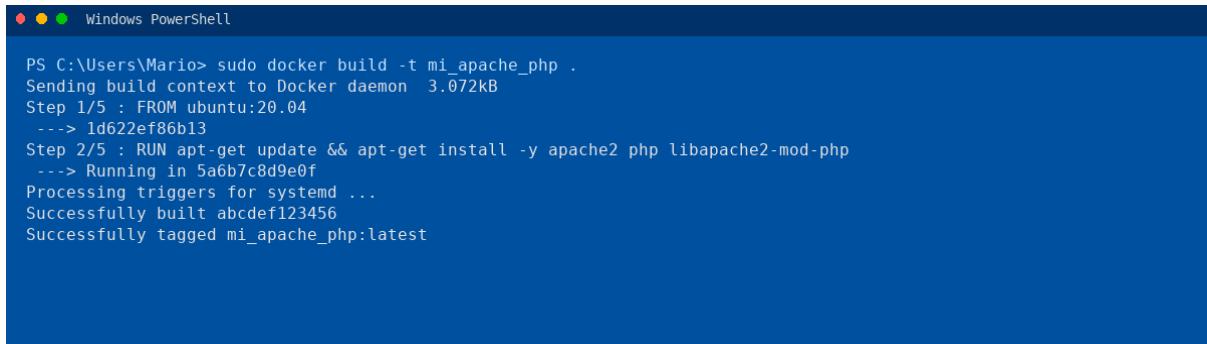


```
PS C:\Users\Mario> sudo docker images
REPOSITORY   TAG      IMAGE ID      CREATED        SIZE
hello-world  latest   e7d4f0c0e5e6  2 weeks ago   13.3kB
```

## Construir imagen personalizada (Apache + PHP)

Comando: sudo docker build -t mi\_apache\_php .

Construye una imagen Docker a partir de un Dockerfile que instala Apache y PHP.

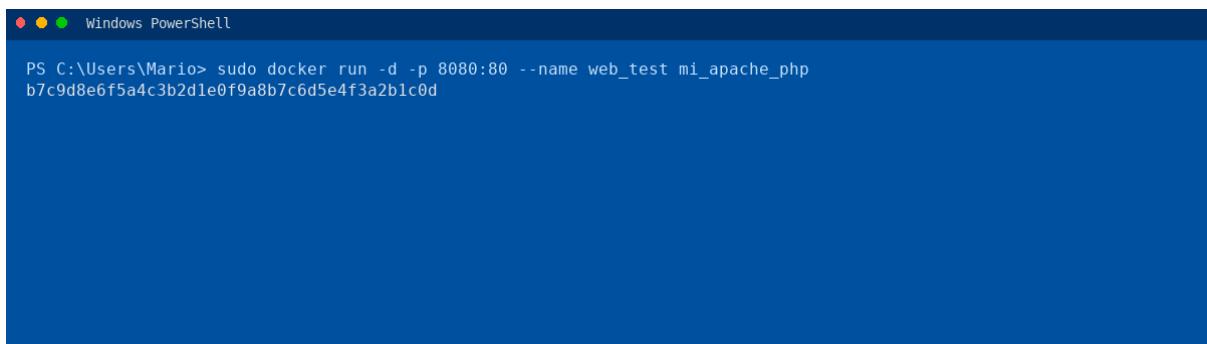


```
PS C:\Users\Mario> sudo docker build -t mi_apache_php .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM ubuntu:20.04
--> 1d622ef86b13
Step 2/5 : RUN apt-get update && apt-get install -y apache2 php libapache2-mod-php
--> Running in 5a6b7c8d9e0f
Processing triggers for systemd ...
Successfully built abcdef123456
Successfully tagged mi_apache_php:latest
```

## Ejecutar el contenedor en background y mapear puertos

Comando: sudo docker run -d -p 8080:80 --name web\_test mi\_apache\_php

Lanza el contenedor en segundo plano exponiendo el puerto 80 del contenedor en el puerto 8080 del host.

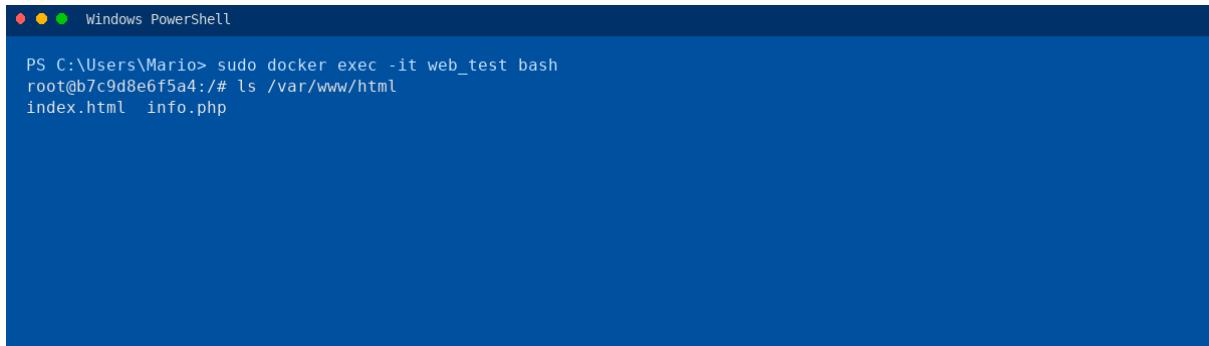


```
PS C:\Users\Mario> sudo docker run -d -p 8080:80 --name web_test mi_apache_php
b7c9d8e6f5a4c3b2d1e0f9a8b7c6d5e4f3a2b1c0d
```

## Entrar al contenedor y listar archivos web

Comando: sudo docker exec -it web\_test bash      ls /var/www/html

Se accede al contenedor para comprobar que los archivos web (index.php, info.php) están presentes.

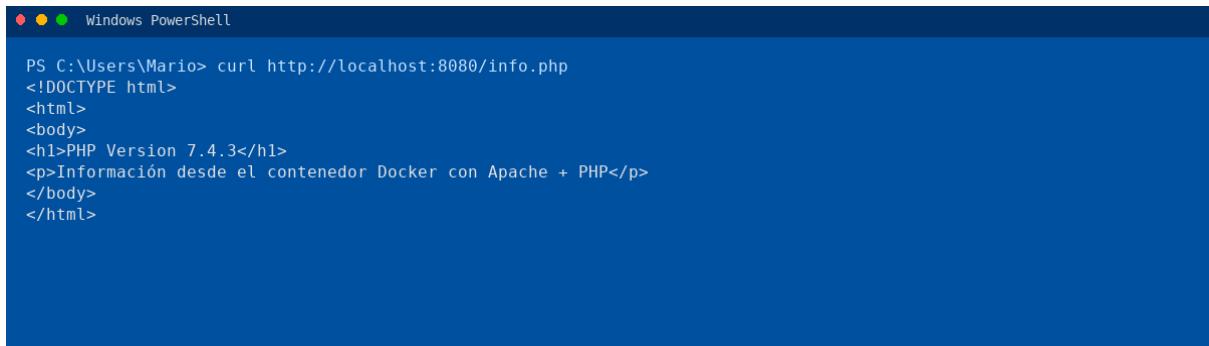


```
PS C:\Users\Mario> sudo docker exec -it web_test bash
root@b7c9d8e6f5a4:/# ls /var/www/html
index.html  info.php
```

## Comprobar la página PHP desde el host

Comando: curl http://localhost:8080/info.php

Desde el host se comprueba que la aplicación dentro del contenedor sirve correctamente la página PHP.



```
PS C:\Users\Mario> curl http://localhost:8080/info.php
<!DOCTYPE html>
<html>
<body>
<h1>PHP Version 7.4.3</h1>
<p>Información desde el contenedor Docker con Apache + PHP</p>
</body>
</html>
```

## Ejemplo de Dockerfile (resumen)

Ejemplo mínimo de Dockerfile usado para crear la imagen mi\_apache\_php:

```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y apache2 php libapache2-mod-php
COPY ./html /var/www/html
EXPOSE 80
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```