

**Actividad 1 - Configuración y Ejecución de una Aplicación Web en PHP
con NGINX en Docker**

Fernando Martinez Ortiz

2º DAW 2023/24

Paso 1: Configuración de Docker

1-Actualización e instalación de paquetes

Primero voy a comprobar que todos los paquetes estén actualizados , y si no lo están voy a instalarlos, para eso voy a utilizar el comando `sudo apt-get update` y después para instalarlos el `sudo apt-get upgrade`

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo apt-get update
```

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo apt-get upgrade
```

2-Actualizar los paquetes necesarios para usar repositorios empleando https

Una vez actualizados los paquetes instalar los paquetes necesarios para permitir a la utilidad apt usar repositorios empleando HTTPS confiando en el certificado de Docker con el siguiente comando:

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

3-Descarga clave pública GPG y adición al repositorio

Luego descargo la clave pública GPG oficial de Docker y su adición al repositorio de claves con el siguiente comando:

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

4-Añadir a la lista de repositorios la correspondiente versión

Añadir a la lista de repositorios, el correspondiente a la versión “estable” para Ubuntu

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) stable"
Repositorio: «deb [arch=amd64] https://download.docker.com/linux/ubuntu \ stable jammy»
Descripción:
Archive for codename: \ components: stable,jammy
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Found existing deb-src entry in /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
E: Entrada 5 mal especificada en list fichero /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list (Component)
E: Tipo «Unable» desconocido en la línea 1 de la lista de fuentes /etc/apt/sources.list.d/docker.list
E: No se pudieron leer las listas de fuentes.
root@usuario-Standard-PC-i440FX-PIIX-1996:~#
```

5-Actualizar repositorios e instalar Docker CE

Actualización de los repositorios.

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo apt-get update
Obj:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:3 https://download.docker.com/linux/ubuntu \ InRelease
Obj:4 https://download.docker.com/linux/ubuntu xenial InRelease
Obj:5 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:6 https://download.docker.com/linux/ubuntu jammy InRelease
Obj:7 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Err:8 https://download.docker.com/linux/ubuntu \ Release
  404 Not Found [IP: 3.160.231.8 443]
Leyendo lista de paquetes... Hecho
W: https://download.docker.com/linux/ubuntu/dists/xenial/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
E: El repositorio «https://download.docker.com/linux/ubuntu \ Release» no tiene
```

Instalo la última versión estable de Docker CE, junto al demonio y cliente con el siguiente comando:

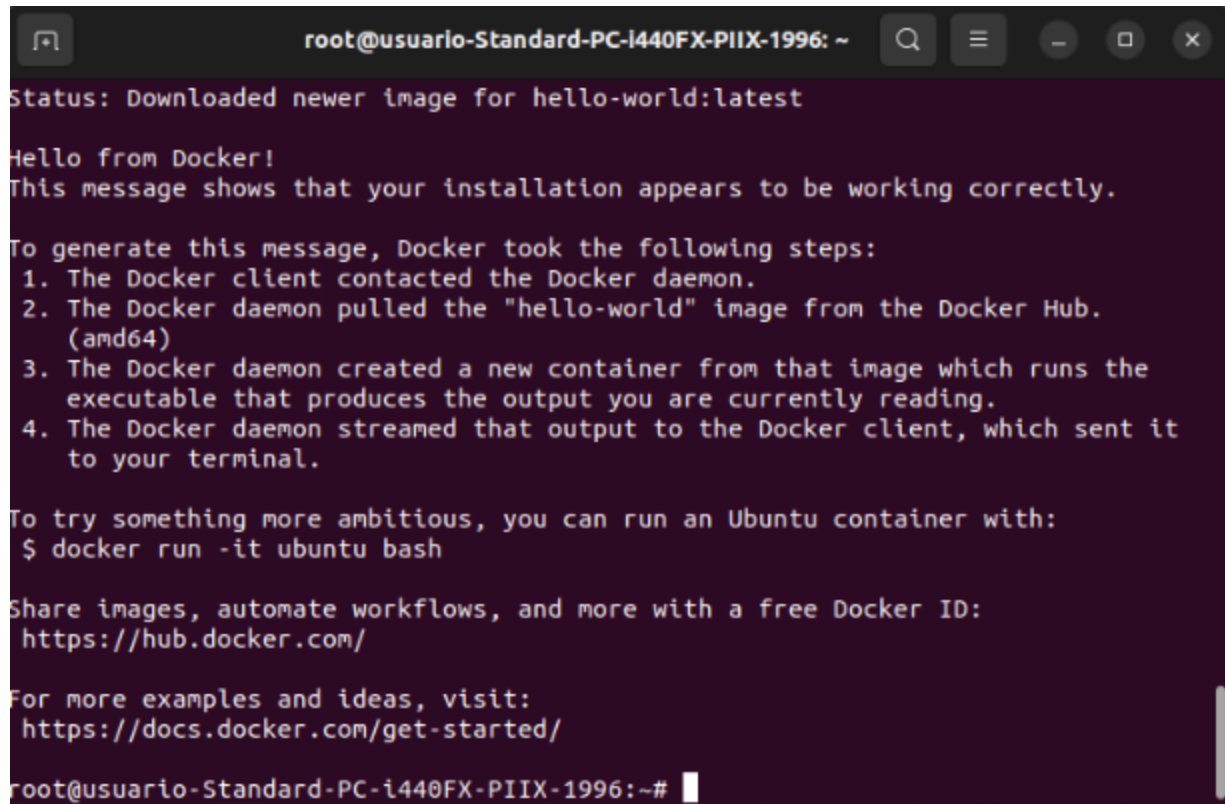
```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo apt-get install docker-ce dock
er-ce-cli containerd.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin git
  git-man liberror-perl libslirp0 pigz slirp4netns
Paquetes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl
  libslirp0 pigz slirp4netns
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 14 no actualizados.
Se necesita descargar 121 MB de archivos.
Se utilizarán 440 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io
```

6-Comprobar que Docker funcione

Comprobamos que Docker CE con un “hello world” con el siguiente comando:

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# sudo docker run hello-world
```

Este comando sirve para recoger una imagen de prueba y comprobar que el docker funciona, y con esto verificamos que esté funcionando correctamente



```
root@usuario-Standard-PC-i440FX-PIIX-1996: ~  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
root@usuario-Standard-PC-i440FX-PIIX-1996:~#
```

Paso 2: Creación del Contenedor de PHP

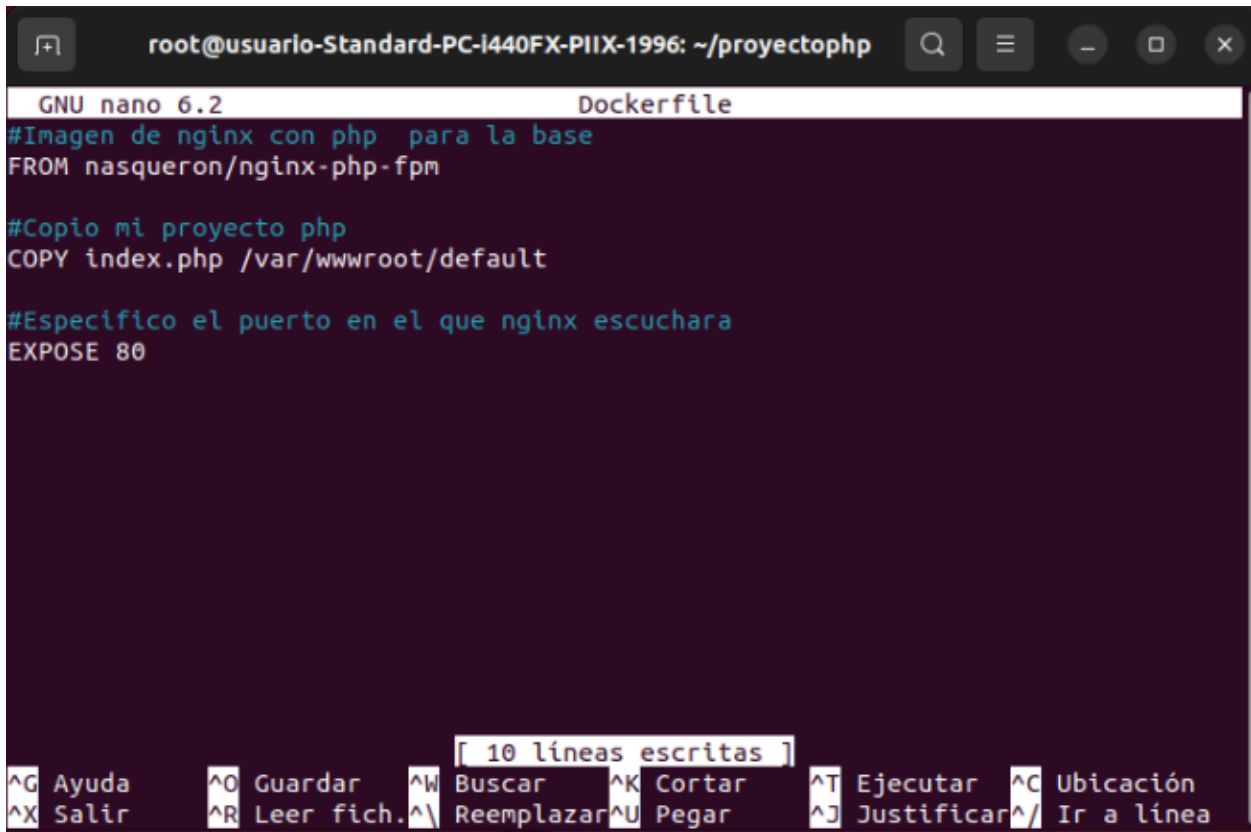
1-Cree un archivo Dockerfile para la aplicación PHP.

2-Configura el archivo Dockerfile para que tenga un servidor web Nginx.

Creo un directorio para el proyecto de php y me meto en el

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~# mkdir proyectophp  
root@usuario-Standard-PC-i440FX-PIIX-1996:~# cd proyectophp/  
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp#
```

Creo el archivo Dockerfile y lo relleno con las siguientes instrucciones



```
root@usuario-Standard-PC-l440FX-PIIX-1996: ~/proyectophp
GNU nano 6.2 Dockerfile
#Imagen de nginx con php para la base
FROM nasqueron/nginx-php-fpm

#Copio mi proyecto php
COPY index.php /var/wwwroot/default

#Especifico el puerto en el que nginx escuchara
EXPOSE 80

[ 10 líneas escritas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir   ^R Leer fich. ^_ Reemplazar ^U Pegar    ^J Justificar ^_ Ir a línea
```

“FROM nasqueron/nginx-php-fpm”

Esta línea establece la imagen base para el contenedor con nginx y php

“COPY index.php /var/wwwroot/default”

Copio el proyecto que tengo de php a la carpeta correspondiente

“EXPOSE 80”

Esta línea indica que el contenedor expone el puerto 80.

3-Configure el entorno de desarrollo de PHP dentro del contenedor.

Muevo mi proyecto de php a la carpeta correspondiente donde realizare el contenedor de docker ya que anteriormente indique la ruta de este

```
root@usuario-Standard-PC-i440FX-PIIX-1996:/home/usuario/Descargas# ls
'Actividad 1 - Configuración y Ejecución de una Aplicación Web en PHP con NGINX
en Docker-1.pdf'
'Actividad 1 - Configuración y Ejecución de una Aplicación Web en PHP con NGINX
en Docker.pdf'
index.php
root@usuario-Standard-PC-i440FX-PIIX-1996:/home/usuario/Descargas# mv index.php
~/proyectophp/
root@usuario-Standard-PC-i440FX-PIIX-1996:/home/usuario/Descargas# cd ..
root@usuario-Standard-PC-i440FX-PIIX-1996:/home/usuario# cd ..
root@usuario-Standard-PC-i440FX-PIIX-1996:/home# cd
root@usuario-Standard-PC-i440FX-PIIX-1996:~# cd proyectophp/
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp# ls
Dockerfile index.php nginx
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp#
```

4-Use el comando docker build para crear la imagen del contenedor de PHP.

Construyo la imagen que voy a utilizar para el contenedor de docker con el nombre de prueba2

```
root@usuario-Standard-PC-i440FX-PIIX-1996: ~/proyectophp
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp# docker build -t prueba2
.
[+] Building 22.3s (7/7) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile               0.0s
=> => transferring dockerfile: 229B                               0.0s
=> [internal] load metadata for docker.io/nasqueron/nginx-php-fpm:latest 7.0s
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 1.64kB                                    0.0s
=> [1/2] FROM docker.io/nasqueron/nginx-php-fpm:latest@sha256:fb9a28f3e 15.0s
=> => resolve docker.io/nasqueron/nginx-php-fpm:latest@sha256:fb9a28f3e6 0.0s
=> => sha256:fb9a28f3e60b22368a47bbbc56e7a8c0d35af7cffe0 1.37kB / 1.37kB 0.0s
=> => sha256:afc6b9026bf4869643aba73972e588e4c4d93d3bbfe 8.50kB / 8.50kB 0.0s
=> => sha256:f1f26f5702560b7e591bef5c4d840f76a232bf13f 31.41MB / 31.41MB 1.1s
=> => sha256:105300cf8caf343e90b0723b0f7ba09f8716005 161.04MB / 161.04MB 4.8s
=> => sha256:a38864cc16a8ca2119f1bc4ea94315355215a6328 47.90MB / 47.90MB 1.9s
=> => extracting sha256:f1f26f5702560b7e591bef5c4d840f76a232bf13fd5aefc4 1.4s
=> => sha256:34e3acc42eeb933cd31d2f8152c2c71c2e67cb194ef 2.04kB / 2.04kB 1.4s
=> => sha256:63e7826a05b9bffc5805555a74affbc3bc02d7159c1 3.30kB / 3.30kB 1.6s
=> => extracting sha256:105300cf8caf343e90b0723b0f7ba09f87160051e1811238 4.5s
=> => extracting sha256:a38864cc16a8ca2119f1bc4ea94315355215a6328af4a8f1 5.1s
=> => extracting sha256:34e3acc42eeb933cd31d2f8152c2c71c2e67cb194ef4e72f 0.0s
=> => extracting sha256:63e7826a05b9bffc5805555a74affbc3bc02d7159c12c3f8 0.0s
=> [2/2] COPY index.php /var/wwwroot/default                    0.2s
=> exporting to image                                             0.1s
=> => exporting layers                                              0.1s
=> => writing image sha256:4aca2e246ac5c76e8848951195d69d11ace857290b637 0.0s
=> => naming to docker.io/library/prueba2                        0.0s
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp#
```


5-Utilice el comando docker run para ejecutar el contenedor de PHP.

Ahora creo el contenedor , lo hecho correr y uso estas 2 opciones:

-d para ejecutar el contenedor en segundo plano

-p para enlazar el puerto del contenedor, a un puerto de la máquina host. En este caso, el puerto 80 del contenedor al puerto 8083 de la máquina host

```
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp# docker run -d -p 8083:80 prueba2
28929b6c112da8a17d259b07d2779c8278a2f81c235cb0226929593ff6c97d9a
root@usuario-Standard-PC-i440FX-PIIX-1996:~/proyectophp#
```

Paso 3: Subir una aplicación en PHP

Finalmente pongo en la barra del navegador <http://localhost:8083> y visualizo la aplicación de php



Teoria elemental de php dia: 24-01-2024

Mi primera pagina Curso 23-24

El resultado de sumar: 8 + 9 es: 17

3 es menor que 17

EL resultado del switch es: 17

Hola 1

Hola 2

Hola 3

Hola 4

Hola 5

Hola 6

Hola 7

Hola 8

Hola 9

Hola 1

Hola 2

Hola 3