

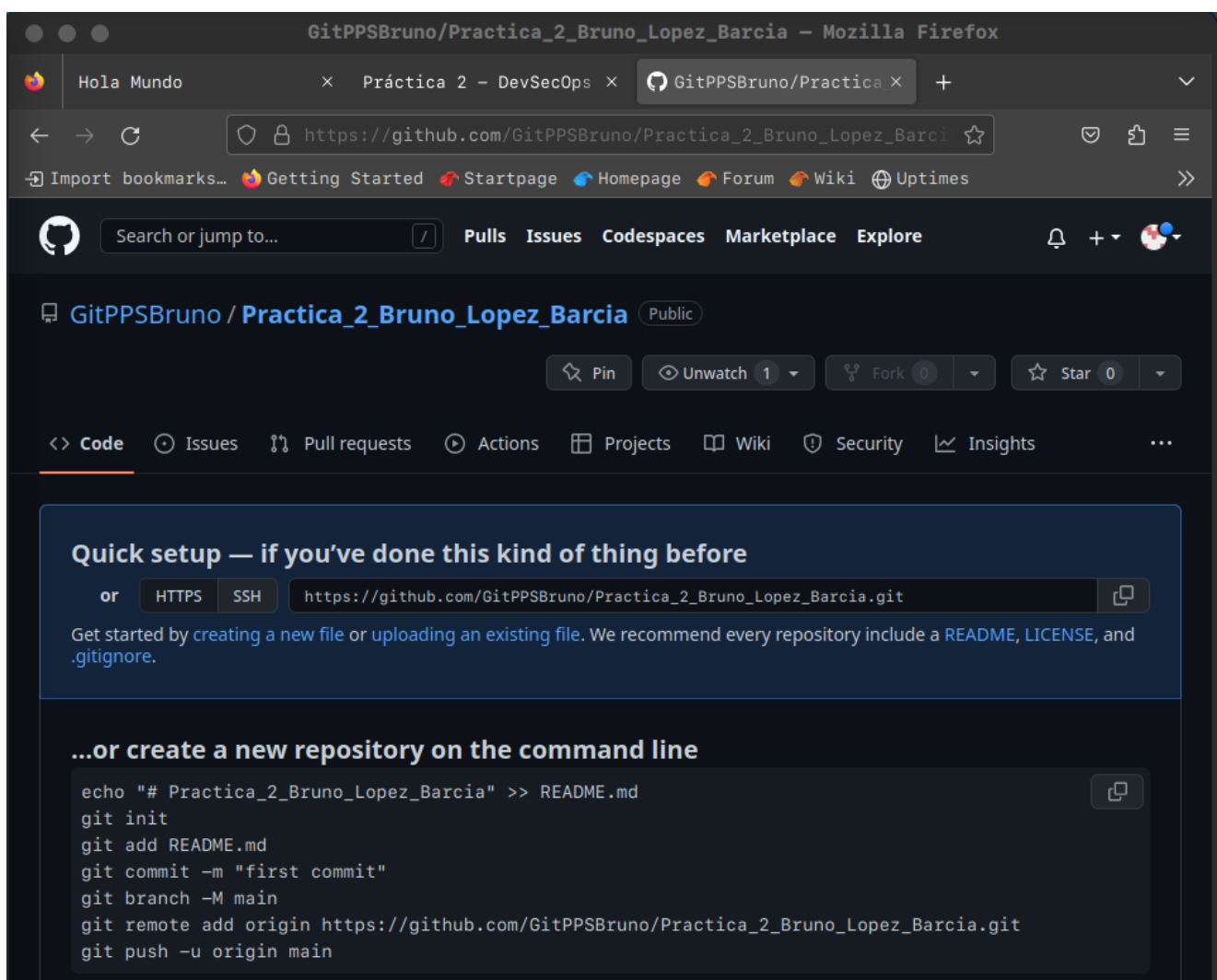
# Memoria Practica 2 Bruno Lopez Barcia

# Memoria Practica 2 Bruno Lopez Barcia

## PreRequisitos

Lo primero que debemos de hacer es crear es el repositorio de git e inicializar el repositorio en local.

Para crear un nuevo repositorio en GitHub debemos de ir a nuestro perfil de GitHub y darle a **Añadir un nuevo repositorio**. Una vez creada nos aparecera esta pantalla:



Una vez nos salga eso debemos de ir a la carpeta donde queremos inicializar nuestro repositorio en local y ejecutar los siguiente:

```
Practica_2_Bruno_Lopez_Barcia : fish - Konsole

brunolb@PCBRUNO in ~/Documentos/PPS took 5s
+ [Q] x mkdir Practica_2_Bruno_Lopez_Barcia

brunolb@PCBRUNO in ~/Documentos/PPS took 3ms
+ λ cd Practica_2_Bruno_Lopez_Barcia/

brunolb@PCBRUNO in ~/Documentos/PPS/Practica_2_Bruno_Lopez_Barcia took 15ms
+ λ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este coman
do:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/brunolb/Documentos/PPS/Practica_2_Bruno_Lopez_Bar
cia/.git/
```

## Ejercicio 1

1. Crea un Dockerfile que partiendo de una imagen PHP genera una imagen que:
  1. Copia una aplicación en PHP a un directorio del contenedor. Esta aplicación se debe copiar directamente desde un directorio del anfitrión. Para facilitar las cosas, debe de ser una aplicación sencilla que no emplee bases de datos (ya que sino también habría que instalar un Mysql)

Una vez que tenemos los requisitos hechos debemos de crear la carpeta del Ejercicio 1. En esta carpeta añadiremos el Dockerfile del ejercicio1, del que luego generaremos la imagen.

Contenido carpeta Ejercicio1:

```
Ejercicio_1 : fish - Konsole

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_1 on  master [?] took 35ms
+ λ ls
drwxr-xr-x - brunolb 25 ene 17:04 php
-rw-r--r-- 57 brunolb 25 ene 17:04 Dockerfile

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_1 on  master [?] took 9ms
+ λ ls -l php/
drwxr-xr-x - brunolb 25 ene 17:04 images
drwxr-xr-x - brunolb 25 ene 17:04 product-images
-rw-r--r-- 2,5k brunolb 25 ene 17:04 ajax-action.php
-rw-r--r-- 1,7k brunolb 25 ene 17:04 index.php
-rw-r--r-- 87k brunolb 25 ene 17:04 jquery-3.2.1.min.js
-rw-r--r-- 1,4k brunolb 25 ene 17:04 product-gallery.php
-rw-r--r-- 828 brunolb 25 ene 17:04 Product.php
-rw-r--r-- 1,2k brunolb 25 ene 17:04 style.css
```

El contenido del Dockerfile es el siguiente:

```
...brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_1 on ȳ master [?] took 36ms
+ ↳ cat Dockerfile

File: Dockerfile
1 FROM php:7.4-apache
2
3 COPY php/ /var/www/html/
4
5 EXPOSE 80
```

En la línea uno lo que hacemos es usar una imagen de PHP en su versión 7.4 con apache. En la línea 2 copiamos la carpeta PHP de nuestra máquina y por último debemos exponer el puerto 80.

Una vez que tenemos configurado el *Dockerfile* y las carpetas configuradas lo que tenemos que hacer es subir ambas carpetas al repositorio de Github. Debemos ejecutar los siguientes comandos

```
git add Ejercicio1/ images/ Memoria.md

git commit -m "Añadido ejercicio 1 y la parte correspondiente de la memoria"

git push -u origin main
```

```
Practica_2_Bruno_Lopez_Barcia : git - Konsole

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ȳ main [!?] took 19ms
↳ git add Ejercicio_1/ images/ Memoria.md

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ȳ main [++] took 7ms
↳ git commit -m "Añadido ejercicio 1 y la parte correspondiente de la memoria."
[main ed98ce4] Añadido ejercicio 1 y la parte correspondiente de la memoria.
2 files changed, 12 insertions(+), 4 deletions(-)
create mode 100644 images/imagen_7.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ȳ main [?:1] took 5ms
↳ git push -u origin main
Username for 'https://github.com': _
```

- 
2. Crea un DockerFile que partiendo de una imagen Ubuntu genera una imagen que:
    1. Instala Apache, de forma que se exponga el puerto 80
    2. Instala PHP

3. Copia una aplicación web de PHP al directorio de Apache que expone las paginas web. Esta aplicacion se debe descargar automáticamente mediante algún comando como *git clone* o *curl*. Para facilitar las cosas, debe de ser una aplicación sencilla que no emplee bases de datos(ya que si no también habría que instalar un MySQL).

Una vez que tenemos los prerequisites hechos debemos de crear la carpeta del Ejercicio 2. En esta carpeta añadiremos el Dockerfile del Ejercicio2, del que luego generaremos la imagen.

Contenido carpeta Ejercicio2:

```
Ejercicio_2 : fish - Konsole
brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_2 on ̣ main [!?:1] took 20ms
λ ls
.rw-r--r-- 487 brunolb 26 ene 20:16  Dockerfile
.rw-r--r-- 1,4k brunolb 26 ene 20:16  my-httpd.conf
```

El contenido del Dockerfile es el siguiente:

```
Ejercicio_2 : fish - Konsole
brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_2 on ̣ main [!?:1] took 5s
λ cat Dockerfile
File: Dockerfile
1  FROM ubuntu
2
3  RUN apt-get update -y
4  RUN DEBIAN_FRONTEND=noninteractive apt-get install -y tzdata
5  #Instalamos PHP y 7ZIP
6  RUN apt-get update -y && apt-get install apache2 php curl libapache2-mod-php -y && rm -rf /var/lib/ap
t/lists/*
7
8  WORKDIR /var/www/html
9
10 COPY ./my-httpd.conf /etc/apache2/apache2.conf
11
12 RUN curl -O https://gist.githubusercontent.com/galcober/1825545/raw/3521b89f6620ea2361d528fb7cbd31888
9177a34/holamundo.php
13
14 EXPOSE 80
15
16 CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

En la linea uno lo que hacemos es usar una imagen de Ubuntu en su ultima versión. En la linea 3 ejecutamos `apt-get update -y` que sirve para actualizar los repositorio de la maquina, para posteriormente instalar aplicaciones.

En la linea 4 y 5 instalamos los paquetes *tzdata* *apache2* *curl* y *libapache2-mod-php*, que son los paquetes necesarios para que funcionen los proximos pasos. Al final para que la imagen tenga un peso menor, ejecutare el comando `rm -rf /var/lib/apt/lists/*`.

En la linea 8 debemos de movernos a la carpeta */var/www/html* para luego posteriormente en las linea 12, realizar acciones que se realizan en esa carpeta.

En la línea 10 lo que hacemos es copiar el `my-httpd.conf` personalizado que tenemos en la carpeta local a la carpeta de `/etc/apache/apache2.conf`.

En la línea 12 lo que hacemos es descargar la aplicación PHP desde github con el comando `curl`.

Por último debemos de exponer el puerto **80** del contenedor y ejecutar en modo demonio el servicio de apache con el comando `CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]`

Una vez que tenemos configurado el *Dockerfile* y las carpetas configuradas lo que tenemos que hacer es subir ambas carpetas al repositorio de Github. Debemos de ejecutar los siguientes comandos

```
git add Ejercicio2/ images/ Memoria.md

git commit -m "Añaido ejercicio 2 y la parte correspondiente de la memoria"

git push -u origin main
```

```

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+:1] took 8ms
λ git add Ejercicio_2/ images/ Memoria.md ./obsidian/

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+:1] took 7ms
λ git status
En la rama main
Tu rama est  adelantada a 'origin/main' por 1 commit.
(usa "git push" para publicar tus commits locales)

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del  rea de stage)
nuevos archivos: .obsidian/app.json
nuevos archivos: .obsidian/appearance.json
nuevos archivos: .obsidian/core-plugins-migration.json
nuevos archivos: .obsidian/core-plugins.json
nuevos archivos: .obsidian/graph.json
nuevos archivos: .obsidian/hotkeys.json
nuevos archivos: .obsidian/workspace.json
nuevos archivos: Ejercicio_2/Dockerfile
nuevos archivos: Ejercicio_2/my-httpd.conf
modificados: Memoria.md
nuevos archivos: images/imagen_4.png
nuevos archivos: images/imagen_5.png
nuevos archivos: images/imagen_6.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+:1] took 8ms
λ git commit -m "A adido ejercicio 2 y la parte correspondiente de la memoria."
[main 78e408a] A adido ejercicio 2 y la parte correspondiente de la memoria.
13 files changed, 348 insertions(+), 1 deletion(-)
create mode 100644 .obsidian/app.json
create mode 100644 .obsidian/appearance.json
create mode 100644 .obsidian/core-plugins-migration.json
create mode 100644 .obsidian/core-plugins.json
create mode 100644 .obsidian/graph.json
create mode 100644 .obsidian/hotkeys.json
create mode 100644 .obsidian/workspace.json
create mode 100644 Ejercicio_2/Dockerfile
create mode 100644 Ejercicio_2/my-httpd.conf
create mode 100644 images/imagen_4.png
create mode 100644 images/imagen_5.png
create mode 100644 images/imagen_6.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+:2] took 10ms
λ git push -u origin main
Username for 'https://github.com': GitPPSBruno
Password for 'https://GitPPSBruno@github.com':
Enumerando objetos: 25, listo.
Contando objetos: 100% (25/25), listo.
Compresi n delta usando hasta 12 hilos
Comprimiendo objetos: 100% (20/20), listo.
Escribiendo objetos: 100% (22/22), 636.10 KiB | 28.91 MiB/s, listo.
Total 22 (delta 2), reusados 0 (delta 0), pack-reusados 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/GitPPSBruno/Practica_2_Bruno_Lopez_Barcia.git
 e897e75..78e408a main -> main
rama 'main' configurada para rastrear 'origin/main'.

```

3. Crea un contenedor para cada una de esas imagenes y verifica que funciona. Para y borra dicho contenedor.

Lo primero que hacer es un *docker build* del dockerfile de cada ejercicio.

Docker build Ejercicio 1:



```

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_1 on   main [!] took 6ms
  docker build -t php-ejercicio-1 .
Sending build context to Docker daemon 120.3kB
Step 1/3 : FROM php:7.4-apache
7.4-apache: Pulling from library/php
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Pull complete
fb5a4c8af82f: Pull complete
25f85b498fd5: Pull complete
9b233e420ac7: Pull complete
fe42347c4ecf: Pull complete
d14eb2ed1e17: Pull complete
66d98f73acb6: Pull complete
d2c43c5efbc8: Pull complete
ab590b48ea47: Pull complete
80692ae2d067: Pull complete
05e465aaa99a: Pull complete
Digest: sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905ff63fae3fe2e0ca6d
Status: Downloaded newer image for php:7.4-apache
----> 20a3732f422b
Step 2/3 : COPY php/ /var/www/html/
----> 067a534f5ed2
Step 3/3 : EXPOSE 80
----> Running in 28432d52da42
Removing intermediate container 28432d52da42
----> 601c8374c1f8
Successfully built 601c8374c1f8
Successfully tagged php-ejercicio-1:latest

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_1 on   main [!] took 10s
  docker image ls
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
php-ejercicio-1     latest      601c8374c1f8  6 seconds ago  453MB
php                 7.4-apache  20a3732f422b  2 months ago  453MB

```

Como podemos ver nos crea la imagen del contenedor; la imagen de php que aparece es la imagen que se ha descargado el *docker build* para poder crear el contenedor.

Ahora debemos de realizar lo mismo para el *Dockerfile* del ejercicio 2.

Docker build Ejercicio 2:

```
brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_2 on   main [!] took 21ms
  docker build -t php-ejercicio-2 .
Sending build context to Docker daemon 4.096kB
Step 1/9 : FROM ubuntu
latest: Pulling from library/ubuntu
6e3729cf69e0: Pull complete
Digest: sha256:27cb6e6ccef575a4698b66f5de06c7ecd61589132d5a91d098f7f3f9285415a9
Status: Downloaded newer image for ubuntu:latest
----> 6b7dfa7e8fdb
Step 2/9 : RUN apt-get update -y
----> Running in 28c84b170296
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [745 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [795 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [681 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1062 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8978 B]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [730 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1002 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7286 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
Fetched 25.3 MB in 4s (6008 kB/s)
Reading package lists...
Removing intermediate container 28c84b170296
----> 9d8d3e660681
Step 3/9 : RUN DEBIAN_FRONTEND=noninteractive apt-get install -y tzdata
----> Running in cee627626e2f
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
tzdata
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 333 kB of archives.
After this operation, 3857 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 tzdata all 2022g-0ubuntu0.22.04.1 [333 kB]
```



```

Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
info: Switch to mpm prefork for package libapache2-mod-php8.1
Module mpm_event disabled.
Enabling module mpm_prefork.
info: Executing deferred 'a2enmod php8.1' for package libapache2-mod-php8.1
Enabling module php8.1.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Setting up php8.1 (8.1.2-1ubuntu2.10) ...
Setting up libapache2-mod-php (2:8.1+92ubuntu1) ...
Setting up php (2:8.1+92ubuntu1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for ca-certificates (20211016ubuntu0.22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.10) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.10) ...
Removing intermediate container 7f70966e6dbd
--> 9395fbdec795
Step 5/9 : WORKDIR /var/www/html
--> Running in 14f19531cc2d
Removing intermediate container 14f19531cc2d
--> 6c952d5f0ee8
Step 6/9 : COPY ./my-httpd.conf /etc/apache2/apache2.conf
--> b14d4dd1cf7a
Step 7/9 : RUN curl -O https://gist.githubusercontent.com/galcober/1825545/raw/3521b89f6620ea2361d528fb7cbd318889177a34/holamundo.php
--> Running in 29e2cca03abb
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  106  100  106    0     0   275      0 --:--:-- --:--:-- --:--:--  275
Removing intermediate container 29e2cca03abb
--> 402a3ab31469
Step 8/9 : EXPOSE 80
--> Running in 85c57598cbf6
Removing intermediate container 85c57598cbf6
--> 7217f442b4b1
Step 9/9 : CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
--> Running in 593d67b12e53
Removing intermediate container 593d67b12e53
--> 522cdb3bc0a7
Successfully built 522cdb3bc0a7
Successfully tagged php-ejercicio-2:latest

```

Como podemos ver ahora si hacemos un *docker image ls* nos aparecen ambas imagenes:

```

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia/Ejercicio_2 on   main [!] took 41s
λ docker image ls

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
php-ejercicio-2	latest	522cdb3bc0a7	47 seconds ago	254MB
php-ejercicio-1	latest	601c8374c1f8	About a minute ago	453MB
ubuntu	latest	6b7dfa7e8fdb	7 weeks ago	77.8MB
php	7.4-apache	20a3732f422b	2 months ago	453MB

Ahora debemos de correr ambos contenedores a la vez.

Para ello usaremos usaremos los siguientes comandos:

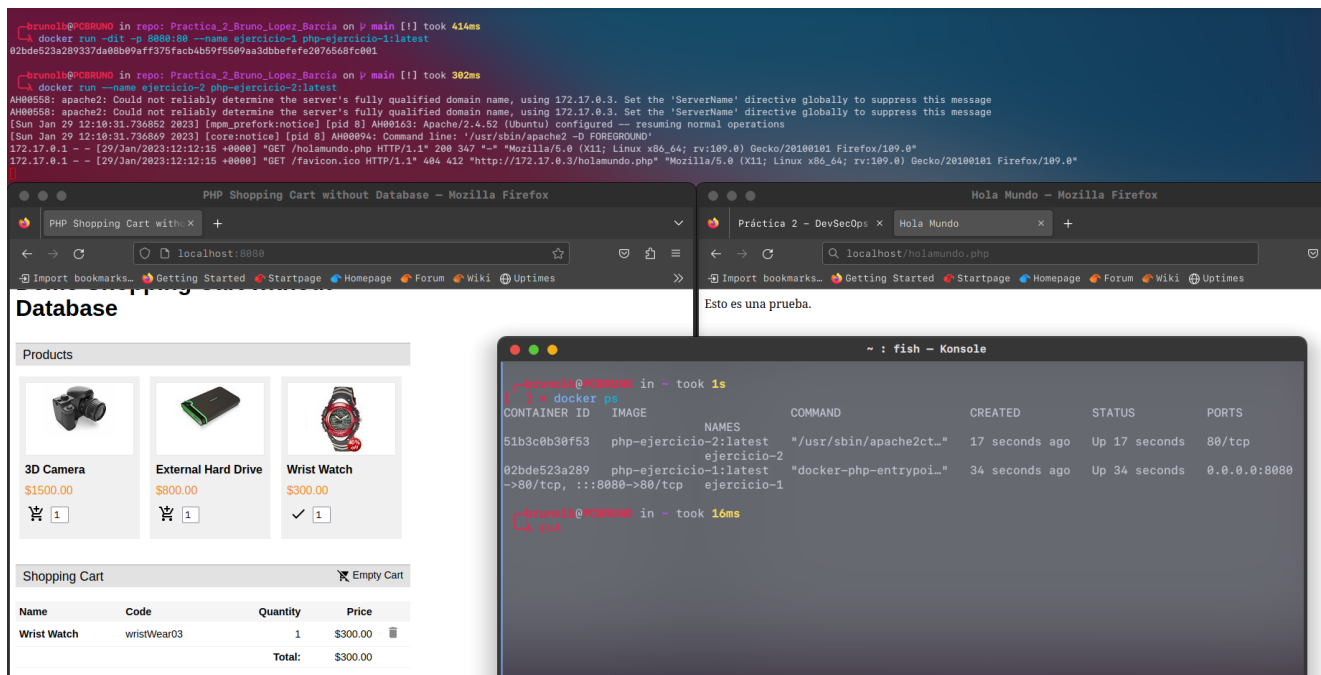
```

docker run -d --name ejercicio-1 php-ejercicio-1:latest #Primer ejercicio

docker run --name ejercicio-2 php-ejercicio-2:latest #Segundo ejercicio

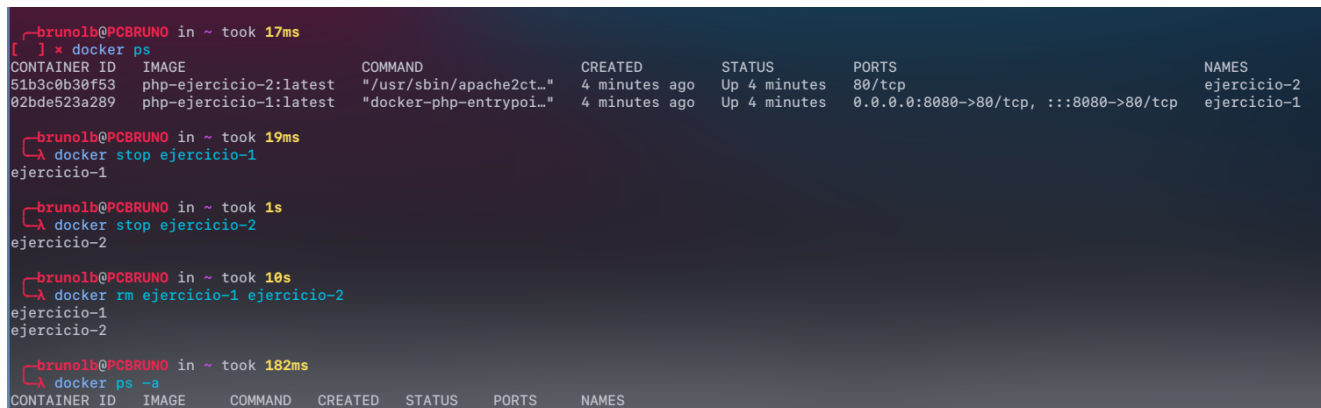
```

Una vez hayamos corrido esos comandos podemos comprobar que estan funcionando, entrando a las paginas web y haciendo un `docker ps`



Ahora para pararlos y borrarlos solo debemos de ejecutar los comandos:

```
docker stop NOMBRE_CONTENEDOR
docker rm NOMBRE_CONTENEDOR
```



Ahora debemos de subir los cambios realizados al github ejecutando los siguientes comandos:

```
git add Ejercicio2/ images/ Memoria.md
git commit -m "Añadiendo los comandos del ejercicio 3 a la memoria"
git push -u origin main
```

```

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [!+] took 11ms
λ git add images/ Memoria.md .obsidian/workspace.json

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+] took 7ms
λ git status
En la rama main
Tu rama est actualizada con 'origin/main'.

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del rea de stage)
modificados:      .obsidian/workspace.json
modificados:      Memoria.md
nuevos archivos:  images/imagen_10.png
nuevos archivos:  images/imagen_11.png
nuevos archivos:  images/imagen_12.png
nuevos archivos:  images/imagen_13.png
modificados:      images/imagen_7.png
nuevos archivos:  images/imagen_8.png
nuevos archivos:  images/imagen_9.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [+] took 8ms
λ git commit -m "Aadiendo los comandos del ejercicio 3 a la documentacion"
[main 402d5e8] Aadiendo los comandos del ejercicio 3 a la documentacion
9 files changed, 75 insertions(+), 9 deletions(-)
create mode 100644 images/imagen_10.png
create mode 100644 images/imagen_11.png
create mode 100644 images/imagen_12.png
create mode 100644 images/imagen_13.png
create mode 100644 images/imagen_8.png
create mode 100644 images/imagen_9.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on   main [1] took 25ms
λ git push -u origin main

```

- 
4. Emplea un comando para lanzar 20 contenedores de la segunda imagen, cada uno mapeado en un puerto distinto del anfitrin. Cuando veas que funcionan, para y borra dichos contenedores.
- 

Para realizar esto no hay ninguna forma nativa con docker de hacerlo, por lo que debemos de hacer con un loop *for* de bash.

El comando a ejecutar es el siguiente:

```

for i in `seq 1 20`; do
docker run -p $((900+$i)) --name ejercicio-2-contenedor-$i php-ejercicio-
2;
done

```

El resultado de ejecutar ese comando seria el siguiente:

```
[bruno1b@PCBRUNO ~]$ for i in `seq 1 20`; do docker run -dit -p $((900+$i)):80 --name ejercicio-2-contenedor-$i php-ejercicio-2; done
5b7be6445b44b34c4e614d3c087a76a321b915e75f2c8eedfe24420ce74364
569f46930f2b4e4667558ed97d0d354802cff5295984e21f3b0d9ca82207ab58
3c0976b75a72837c6fff79a8113adb01fe1c0b93d49b766e9b3e2f39a34e6eac
429be0c32edbe7bf296ab8504e64d0421a2b532247115fa46657c77de723245e
077e90286912829510eb448d83f7c5f68bfff77d23e61fe50e1d4291db5c42d6c
c3a1ec3e192d3b6ba370590380e2691985dfba675b71b744e494ac62d9d99a08
c92b5741ffab7a97220b24bf4c5f01ac930f60e810fdc7ef089d21b01b409754
5dbb80f2e1ccb51f0ae1e4667e673f632f8ad2b50b331c0ce057b78a84168d59
c07a4e34456bae5a6df8f5cc8b6d62b13a4722499773eb5513cfa28856eccc5
559fd2cb94b45db2b0e80f2bccfab0db800bf199b50de150c90ad28ba5cb57c
20fb9e60e2164a50a6abb300547c087aad31657953c7a15bf8f9233caaecf1
aa354e9c6f94cf646cd336d99802d5f3bbfc4f83edd3fc8b94cf43788318460f
f19960999315e71f029b527e1a358dc439d8941a03262f9636ea39ecddfbdda1
976a05525ef15f25ea9af34bd9c04d6a244d08b4a813ad626422881f46b1f305
120420a746cd351b7bdeba9f9ed1f7b3b946325a5b2fc2a78967cbae1f9a036d8
fa7b223a46587227c66227461b944005c153707a67722d9b97002e77e36f2ce8
20a8e0dfc2111dad0d1aada845d8987fca6206c89a30578d1410bea8aa01a9af
257920b27432e11302ba40099c3645f6fac336802b58e673fae0917c389cba1f
d6ea8a1b107b78ba279bd130a0787569af2d75e64e8555c10483169524dae1b
81bca1ac62227589e9ac0cc1a28349d25a34c05a049afd2f2a7b72f20def3c07
[bruno1b@PCBRUNO ~]$
```

Como se ve este comando saca una lista de los contenedores creados.

Ahora debemos de probar si se puede conectar a algun contenedor de los que hemos creado.

Para ello utilizaremos el navegador.

The screenshot displays a terminal window and two browser windows. The terminal window shows the command `docker ps` and its output, which lists 20 containers named `ejercicio-2-contenedor-1` through `ejercicio-2-contenedor-20`. Each container is running the `php-ejercicio-2` image and has port 80 mapped to a unique port starting from 900. The browser windows show the result of connecting to these ports: the first window shows `localhost:910/holamundo.php` and the second shows `localhost:905/holamundo.php`, both displaying the text "Esto es una prueba."

Como vemos en la captura anterior efectivamente se permite la conexión en distintos puertos.

Ahora debemos de parar los contenedores que hemos creado.

Para para los contenedores ejecutaremos el siguiente comando:

```
for i in `seq 1 20`; do
docker stop ejercicio-2-contenedor-$i;
done
```

Una vez parados podemos eliminarlos con el siguiente comando:

```
for i in `seq 1 20`; do
docker rm ejercicio-2-contenedor-$i;
done
```

Si los ejecutamos vemos lo siguiente:

```
[brunolb@PCBRUNO ~]$ for i in `seq 1 20`; do docker stop ejercicio-2-contenedor-$i; done
ejercicio-2-contenedor-1
ejercicio-2-contenedor-2
ejercicio-2-contenedor-3
ejercicio-2-contenedor-4
ejercicio-2-contenedor-5
ejercicio-2-contenedor-6
ejercicio-2-contenedor-7
ejercicio-2-contenedor-8
ejercicio-2-contenedor-9
ejercicio-2-contenedor-10
ejercicio-2-contenedor-11
ejercicio-2-contenedor-12
ejercicio-2-contenedor-13
ejercicio-2-contenedor-14
ejercicio-2-contenedor-15
ejercicio-2-contenedor-16
ejercicio-2-contenedor-17
ejercicio-2-contenedor-18
ejercicio-2-contenedor-19
ejercicio-2-contenedor-20
[brunolb@PCBRUNO ~]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
[brunolb@PCBRUNO ~]$ for i in `seq 1 20`; do docker rm ejercicio-2-contenedor-$i; done
ejercicio-2-contenedor-1
ejercicio-2-contenedor-2
ejercicio-2-contenedor-3
ejercicio-2-contenedor-4
ejercicio-2-contenedor-5
ejercicio-2-contenedor-6
ejercicio-2-contenedor-7
ejercicio-2-contenedor-8
ejercicio-2-contenedor-9
ejercicio-2-contenedor-10
ejercicio-2-contenedor-11
ejercicio-2-contenedor-12
ejercicio-2-contenedor-13
ejercicio-2-contenedor-14
ejercicio-2-contenedor-15
ejercicio-2-contenedor-16
ejercicio-2-contenedor-17
ejercicio-2-contenedor-18
ejercicio-2-contenedor-19
ejercicio-2-contenedor-20
[brunolb@PCBRUNO ~]$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
[brunolb@PCBRUNO ~]$ _
```

Como se puede primero se paran los contenedores y luego se procede a borrarlos.

Por ultimo debemos de subir la nueva version de la Memoria a Github para ello utilizaremos el siguiente comando:

```
git add Ejercicio2/ images/ Memoria.md

git commit -m "Añadiendo los comandos del ejercicio 4 a la memoria"

git push -u origin main
```



```
brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ̣ main [x!+] took 7ms
λ git add images/ Memoria\ Practica\ 2\ Bruno\ Lopez\ Barcia.md Memoria.md .obsidian/workspace.json

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ̣ main [»+] took 5ms
λ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del área de stage)
modificados:      .obsidian/workspace.json
renombrados:      Memoria.md -> Memoria Practica 2 Bruno Lopez Barcia.md
nuevos archivos:  images/imagen_15.png
nuevos archivos:  images/imagen_16.png
nuevos archivos:  images/imagen_17.png
nuevos archivos:  images/imagen_18.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ̣ main [»+] took 12ms
λ git commit -m "Añadiendo los comandos del ejercicio 4 a la memoria"
[main c216623] Añadiendo los comandos del ejercicio 4 a la memoria
6 files changed, 65 insertions(+), 9 deletions(-)
rename Memoria.md => Memoria Practica 2 Bruno Lopez Barcia.md (76%)
create mode 100644 images/imagen_15.png
create mode 100644 images/imagen_16.png
create mode 100644 images/imagen_17.png
create mode 100644 images/imagen_18.png

brunolb@PCBRUNO in repo: Practica_2_Bruno_Lopez_Barcia on ̣ main [+1] took 21ms
λ git push -u origin main
```