

Build1

Creamos un archivo en php para la página, el cual tendrá el siguiente contenido:

```
<?php
echo "¡Hola desde un contenedor PHP!";
?>
```

Y creamos un dockerfile con el siguiente contenido:

```
FROM php:8.2-cli
WORKDIR /usr/src/myapp
COPY . /usr/src/myapp
CMD ["php", "-S", "0.0.0.0:8000"]
```

Tienen que tener esta estructura para que el COPY funcione, ya que copia archivos del directorio actual.

```
usuario@pps:~/Parte1/Practica4/Build1$ tree
.
├── dockerfile
└── index.php

0 directories, 2 files
usuario@pps:~/Parte1/Practica4/Build1$
```

Lo buildeamos con `docker build -t mi_php_app .`

Siendo el comando: `docker build -t nombre_imagen ubicacion_dockerfile`

```
usuario@pps:~/Parte1/Practica4/Build1$ docker build -t mi_php_app .
[+] Building 1.9s (8/8) FINISHED
=> [internal] load build definition from dockerfile
=> == transferring dockerfile: 136B
=> [internal] load metadata for docker.io/library/php:8.2-cli
=> [internal] load .dockerignore
=> == transferring context: 2B
=> [1/3] FROM docker.io/library/php:8.2-cli@sha256:ed4385b854a7ef4aeee1108c75333443d64c937faaf7c7d28bf63a436df06428
=> [internal] load build context
=> == transferring context: 223B
=> CACHED [2/3] WORKDIR /usr/src/myapp
=> [3/3] COPY . /usr/src/myapp
=> exporting to image
=> == exporting layers
=> == writing image sha256:0808bf620bfb7e0fb3d95b80fdb089944c6d1d5cb2ad686e1c2fc97bfb4c12
=> == naming to docker.io/library/mi_php_app
usuario@pps:~/Parte1/Practica4/Build1$
```

Y creamos el contenedor con la imagen con el siguiente comando: `docker run -d -p 8000:8000 --name php_app mi_php_app`

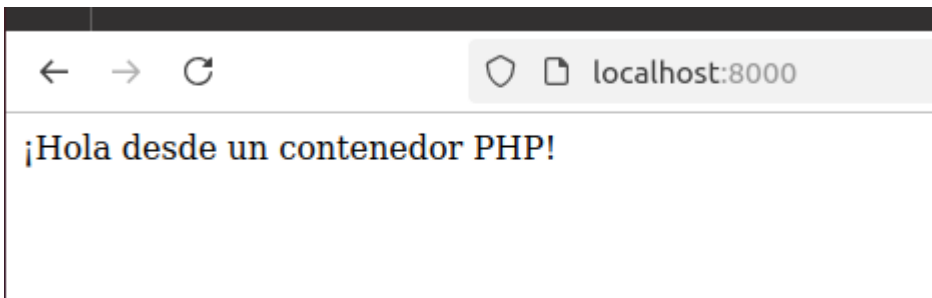
Siendo el comando: `docker run -d -p puerto_propio:puerto_contenedor --name nombre_contenedor nombre_imagen .`

```

usuario@pps:~/Parte1/Practica4/Build1$ docker run -d -p 8000:8000 --name php_app mi_php_app
1889ba040466e5c6795825d828a3811a5461666a2d13919e660c5aa7e2e47a13
usuario@pps:~/Parte1/Practica4/Build1$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
1889ba040466   mi_php_app    "docker-php-entrypoi..." 4 seconds ago  Up 4 seconds  0.0.0.0:8000->8000/tcp, :::8000->8000/tcp  php_app
usuario@pps:~/Parte1/Practica4/Build1$

```

Comprobamos un nuestro puerto 8000 que, efectivamente, el contenedor esta funcionando.



Build2

Creamos el archivo dockerfile

```

FROM ubuntu:22.04

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && \
    apt-get install -y apache2 php git unzip && \
    apt-get clean

RUN rm -rf /var/www/html/* && \
    git clone https://github.com/blueimp/Gallery.git /var/www/html/

RUN chown -R www-data:www-data /var/www/html

EXPOSE 80

CMD ["apachectl", "-D", "FOREGROUND"]

```

Contruimos la imagen con docker build:

```

usuario@pps:~/Parte1/Practica4/Build2$ docker build -t mi_php_app2 .
[+] Building 1.8s (8/8) FINISHED
=> [internal] load build definition from dockerfile
=> == transferring dockerfile: 383B
=> [internal] load metadata for docker.io/library/ubuntu:22.04
=> [internal] load .dockerignore
=> == transferring context: 2B
=> [1/4] FROM docker.io/library/ubuntu:22.04@sha256:67cadaff1dca187079fce41360d5a7eb6f7dcd3745e53c79ad5efd8563118240
=> CACHED [2/4] RUN apt-get update && apt-get install -y apache2 php git unzip && apt-get clean
=> CACHED [3/4] RUN rm -rf /var/www/html/* && git clone https://github.com/blueimp/Gallery.git /var/www/html/
=> CACHED [4/4] RUN chown -R www-data:www-data /var/www/html
=> exporting to image
=> == exporting layers
=> == writing image sha256:b6c2045da1d0703e898e640600d611711a17057a9d19ba9253459622c03a1579
=> == naming to docker.io/library/mi_php_app2
usuario@pps:~/Parte1/Practica4/Build2$

```

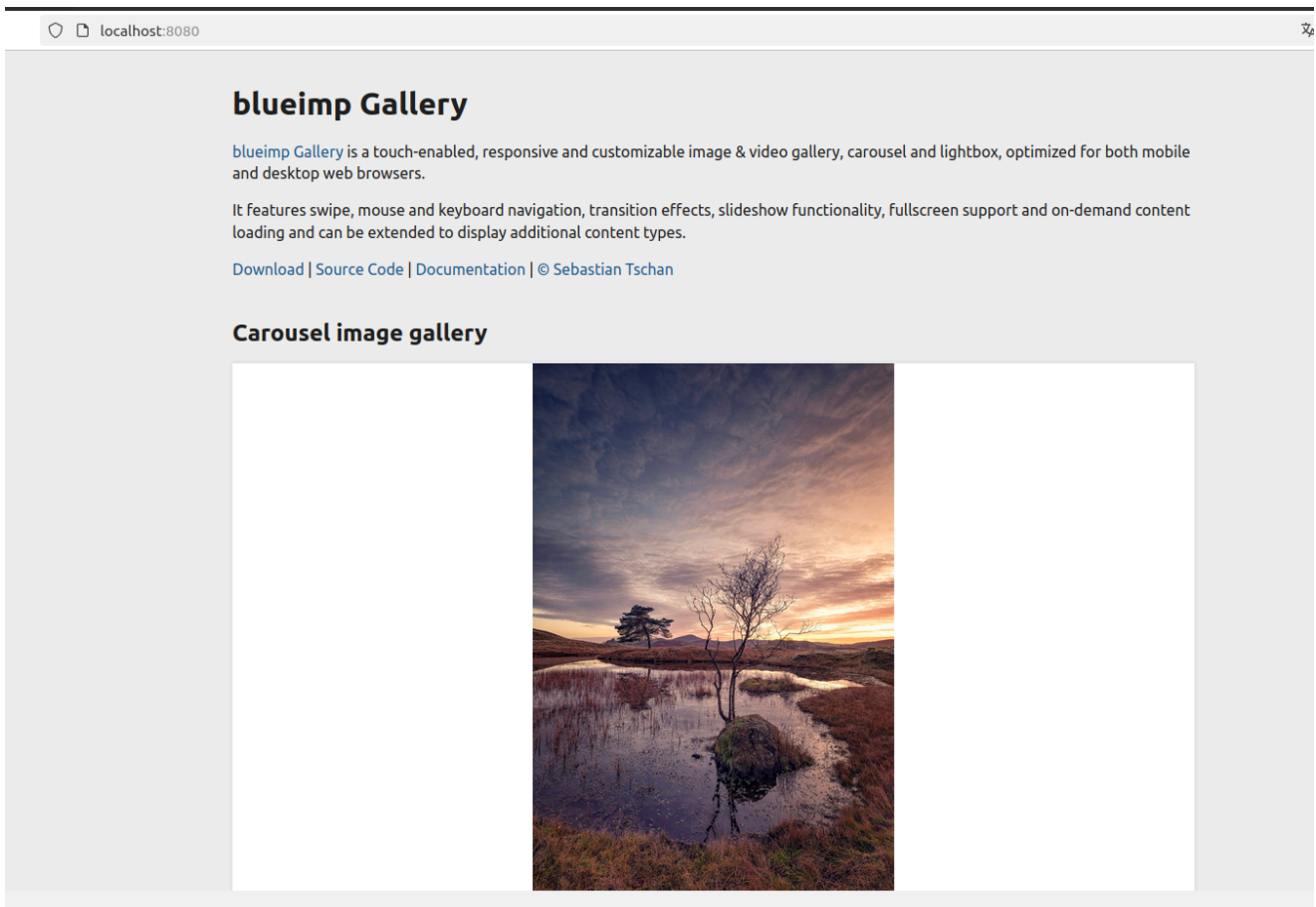
Creamos el contenedor con el comando `docker run -d -p 8080:80 --name ubuntu_app mi_php_app2 :`

```

usuario@pps:~/Parte1/Practica4/Build2$ docker run -d -p 8080:80 --name ubuntu_app mi_php_app2
9b84d29626c237f3a6b01c4b8265dd93a361b15b6599fc8cf610a5687de209b7
usuario@pps:~/Parte1/Practica4/Build2$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
9b84d29626c2   mi_php_app2    "apachectl -D FOREGR..." 4 seconds ago  Up 3 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  ubuntu_app
231cfd1d968e   mi_php_app     "docker-php-entrypoi..." About an hour ago  Up About an hour  0.0.0.0:8000->8000/tcp, ::8000->8000/tcp  php_app

```

Accedemos a la página web para comprobar su funcionamiento.



Verificamos el nombre del contenedor para borrarlo, lo detenemos, lo borramos y comprobamos que se ha borrado.

```

usuario@pps:~/Parte1/Practica4$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
9b84d29626c2   mi_php_app2    "apachectl -D FOREGR..." 3 minutes ago  Up 3 minutes  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  ubuntu_app
231cfd1d968e   mi_php_app     "docker-php-entrypoi..." About an hour ago  Up About an hour  0.0.0.0:8000->8000/tcp, ::8000->8000/tcp  php_app
usuario@pps:~/Parte1/Practica4$ docker stop ubuntu_app
ubuntu_app
usuario@pps:~/Parte1/Practica4$ docker rm ubuntu_app
ubuntu_app
usuario@pps:~/Parte1/Practica4$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
231cfd1d968e   mi_php_app     "docker-php-entrypoi..." About an hour ago  Up About an hour  0.0.0.0:8000->8000/tcp, ::8000->8000/tcp  php_app
usuario@pps:~/Parte1/Practica4$

```

Para lanzar 20 veces la imagen lo automatizaremos con un script en bash

```

echo "Que imagen vamos usar?"
read app
for i in {1..20}; do
    port=$((8080 + i))
    docker run -d -p $port:80 --name apache_php_$i $app
done

```

Y lo comprobamos ejecutando un `docker ps` tras ejecutar el script.

```

usuario@pps:~/Parte1/Practica4$ bash crearlos_a_todos.sh
Que imagen vamos usar?
mi_php_app2
4208aad9f12764596db6b3c52871a8d3f673f36797a5a6bf7dc2550f754bfcfb
7bc944d703b89347fdd0ad60a2e2e477c93f2e18803526e7380c6347f5314e8e
9293788cc0cca06c35c03fc89cb29da12432eaa8e92dbd9be4103caf0cbe53a
5427003439c98fd8c028fa7b05cf2f3a7075facf8139a25eaf0ad4fd3ca265cc
b1938a52d0d2738e9f2419dbd3e59c57691b079e70915b1781a9f231f4fcd5e4
8c4dbcecf9b4a81d480f3b90144559babf871a4a50daa2a5787e16a06e730977
570a089d6760cf445f8b52fcd1cd75a8445ca890187bb735f44322ae7815f4f
351b24165888d0290543c7436bb475e7e4ea9a70426e47f3be1b339e250bfe6b
d29c0b46c786dc42de997e048450b93c984bae7e51a5db677442f1d692825284
3d7aa4d455ec48fa3e10846e912f3aa1631a6180c56e6ad1eeec1ef65126da50
09f22de0fa5d0d7ab470cd5a9d3786ec305d97132735125b3801cf4d6b4981
3ed307a58e91ce73290660a18127241a70b047192d2dcabe14a1d743e167db1c
728ef818541231453d11d2b33bba6b620bc8e228e15c8473f6df9849e7f511
e357fa3a3d66ddb0d7b8bc772382db39c6ca893f439e3b2ed5abb90883a0708
6266fd479a19b3cfa5f54a195eca03a348efaaa5619df6cddb3e46e6f8e16e74
6530384f619930f1d503ebf9f2a01470ea2f92019250aa9fc050f3a63ed57f5
042b23541399784a4d91f63e425b2cc44219e62666aae9411501129324e70145
80538c7c0e63d1f30681f9ac1b0b255ad180113483c19c6c7951a684d594c315
509c2dfafaf2271e9360f12e7548f47ba61a7e3c482ac1e0070ba3dcf4799ae4
6d50bc87b32c96bdee61271be47ef1226855d2bfe3c708c6b9b735859edaf01c
usuario@pps:~/Parte1/Practica4$ docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6d50bc87b32c	mi_php_app2	"apachectl -D FOREGR..."	8 seconds ago	Up 7 seconds	0.0.0.0:8100->80/tcp, [::]:8100->80/tcp	apache_php_20
509c2dfafaf2	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8099->80/tcp, [::]:8099->80/tcp	apache_php_19
8d538c7c06e3	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8098->80/tcp, [::]:8098->80/tcp	apache_php_18
042b23541399	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8097->80/tcp, [::]:8097->80/tcp	apache_php_17
6530384f6199	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8096->80/tcp, [::]:8096->80/tcp	apache_php_16
6266fd479a19	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8095->80/tcp, [::]:8095->80/tcp	apache_php_15
e357fa3a3d66	mi_php_app2	"apachectl -D FOREGR..."	9 seconds ago	Up 8 seconds	0.0.0.0:8094->80/tcp, [::]:8094->80/tcp	apache_php_14
728ef8185412	mi_php_app2	"apachectl -D FOREGR..."	10 seconds ago	Up 9 seconds	0.0.0.0:8093->80/tcp, [::]:8093->80/tcp	apache_php_13
3ed307a58e91	mi_php_app2	"apachectl -D FOREGR..."	10 seconds ago	Up 9 seconds	0.0.0.0:8092->80/tcp, [::]:8092->80/tcp	apache_php_12
09f22de0fa5d	mi_php_app2	"apachectl -D FOREGR..."	10 seconds ago	Up 9 seconds	0.0.0.0:8091->80/tcp, [::]:8091->80/tcp	apache_php_11
3d7aa4d455ec	mi_php_app2	"apachectl -D FOREGR..."	10 seconds ago	Up 9 seconds	0.0.0.0:8090->80/tcp, [::]:8090->80/tcp	apache_php_10
d29c0b46c786	mi_php_app2	"apachectl -D FOREGR..."	10 seconds ago	Up 9 seconds	0.0.0.0:8089->80/tcp, [::]:8089->80/tcp	apache_php_9
351b24165888	mi_php_app2	"apachectl -D FOREGR..."	11 seconds ago	Up 10 seconds	0.0.0.0:8088->80/tcp, [::]:8088->80/tcp	apache_php_8
570a089d6760	mi_php_app2	"apachectl -D FOREGR..."	11 seconds ago	Up 10 seconds	0.0.0.0:8087->80/tcp, [::]:8087->80/tcp	apache_php_7
8c4dbcecf9b4	mi_php_app2	"apachectl -D FOREGR..."	11 seconds ago	Up 10 seconds	0.0.0.0:8086->80/tcp, [::]:8086->80/tcp	apache_php_6
b1938a52d0d2	mi_php_app2	"apachectl -D FOREGR..."	11 seconds ago	Up 10 seconds	0.0.0.0:8085->80/tcp, [::]:8085->80/tcp	apache_php_5
5427003439c9	mi_php_app2	"apachectl -D FOREGR..."	11 seconds ago	Up 10 seconds	0.0.0.0:8084->80/tcp, [::]:8084->80/tcp	apache_php_4
9293788cc0c	mi_php_app2	"apachectl -D FOREGR..."	12 seconds ago	Up 11 seconds	0.0.0.0:8083->80/tcp, [::]:8083->80/tcp	apache_php_3
7bc944d703b8	mi_php_app2	"apachectl -D FOREGR..."	12 seconds ago	Up 11 seconds	0.0.0.0:8082->80/tcp, [::]:8082->80/tcp	apache_php_2
4208aad9f127	mi_php_app2	"apachectl -D FOREGR..."	12 seconds ago	Up 11 seconds	0.0.0.0:8081->80/tcp, [::]:8081->80/tcp	apache_php_1
231cfdd968e	mi_php_app	"docker-php-entrypoi..."	About an hour ago	Up About an hour	0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp	php_app

```

usuario@pps:~/Parte1/Practica4$

```

Y Ya que deterlos y borrarlos uno a uno lo haremos todo junto con el siguiente script:

```

for i in {1..20}; do
    docker rm apache_php_$i -f
done

```

Git

Para gestionar git desde la línea de comandos tenemos que añadir el nombre de usuario y el correo, y si queremos no introducir el token en cada modificación tenemos que decirle que lo almacene.

```

git config --global user.name "nombre_usuario"
git config --global user.email "correo_electronico"
git config --global credential.helper store

```

Con esto ya configurado creamos el repositorio en GIT y lo clonamos a la ruta donde queramos trabajar con `git clone https://github.com/ppsrafa/Repositorio1.git`.

Con esto ya hecho lo que haremos será desde el directorio raíz ir actualizando el repositorio añadiendo modificaciones con:

- `git add .`
- Siendo `.` el directorio raíz del proyecto.

Añadir comentarios para cada actualización que hagamos con:

- `git commit -m "comentario"`

Y finalmente para actualizar el repositorio usaremos:

- git push

```
usuario@pps:~/Parte1/Practica4$ git add .
usuario@pps:~/Parte1/Practica4$ git commit -m "Añadimos los scripts"
[main fcc7414] Añadimos los scripts
 2 files changed, 9 insertions(+)
 create mode 100644 borrarlos_a_todos.sh
 create mode 100644 crearlos_a_todos.sh
usuario@pps:~/Parte1/Practica4$ git push
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 2 hilos
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (4/4), 566 bytes | 566.00 KiB/s, listo.
Total 4 (delta 0), reusado 0 (delta 0)
To https://github.com/ppsrafa/Practica4.git
 0438a9b..fcc7414  main -> main
```

Rafa