

1. Creación del repositorio git
 - a. git init : creacion del repositorio git local
 - b. git remote add origin <https://github.com/Danielxd13/dockerfilePHP.git> : añadir repositorio y union del git local con el github
 - c. creas el documento php y lo metes en el repositorio
 - d. git add . : añadimos todo lo q tengamos para subir
 - e. git commit -m "subida del documento php" : hacemos un commit
 - f. git branch -M main : ponemos a main como master del git
 - g. git push -u origin main : subir las cosas al github
1. Crea un Dockerfile que partiendo de una imagen PHP genera una imagen que:
 - A. 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 si no también habría que instalar un MySQL).
 1. Creamos una carpeta llamada dockerfilephp q tendra nuestro repositorio git con nuestros documentos a subir de la actividad.(se crea el repositorio como esta arriba al principio de la actividad)
 2. creamos o documento php, lo metemos adentro del fichero Dockerfilephp
 3. y creamos el Dockerfile (adentro tienen la explicación de todo los comandos)

```

usuario@pps:~/dockerfilephp$ cat Dockerfile
#partiendo de una imagen de php
from php:latest
#copiar proyecto php en la carpeta de servicios html
COPY documento.php /var/www/html/
#ejecutamos el proyecto -S modo silencio -t directorio q se ejecutara el php
CMD ["php", "-S ", "0.0.0.0:8080", "-t", "/var/www/html/"]

```

4. subida de todo con push

5. sudo docker build -t imagenphp1 . Para iniciar el dockerFile

```

usuario@pps:~/dockerfilephp$ sudo docker run -d --rm -p 80:8080 --name
contenedorPHP imagenphp1
[sudo] contraseña para usuario:
8ae792422825ef32841e06dffed5fb42bf390cbc5a2fa74ed18622e4bf45b473

```

6. lo iniciamos haciendo el run: -d modo demonio --rm q se pare y se elimine el run del contenedor al terminal.

```

usuario@pps:~/dockerfilephp$ sudo docker run --name contenedorPHP4 docker
<!DOCTYPE html>
<html>
<body>

Your favorite color is red!
</body>
</html>
usuario@pps:~/dockerfilephp$ cat index.php
cat: index.php: No existe el archivo o el directorio
usuario@pps:~/dockerfilephp$ cat documento.php
<!DOCTYPE html>
<html>
<body>

<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>

</body>
</html>

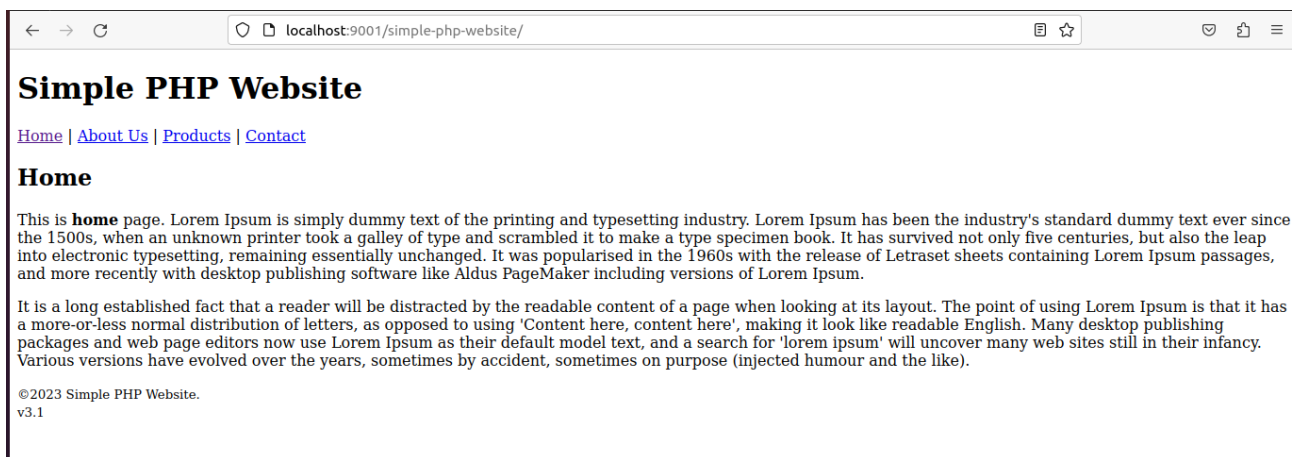
```

2. Crea un Dockerfile que partiendo de una imagen Ubuntu genera una imagen que:
 - A. Instala Apache, de forma que se exponga el puerto 80
 - B. Instala PHP.
 - C. Copia una aplicación web en PHP al directorio de Apache que expone las páginas web. Esta aplicación 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).
3. Crea un contenedor para cada una de esas imágenes y verifica que funciona. Para y borra dicho contenedor.

```
1 #instalacion del repositorio ubuntu
2 FROM ubuntu:latest
3 #no situamos en la raiz
4 WORKDIR /
5 #hacemos un update y intalaciones
6 RUN apt-get update
7 RUN DEBIAN_FRONTEND=noninteractive apt install -y tzdata
8 RUN apt-get install -y apache2 php libapache2-mod-php git apache2-utils
9 # creacion de una carpeta
10 RUN mkdir simple-php-website
11 #entrar a una carpeta
12 WORKDIR /simple-php-website
13 #descargar repository git
14 RUN git clone https://github.com/banago/simple-php-website.git
15 #copiar en la carpeta q da servicio
16 RUN cp -r /simple-php-website/simple-php-website/ /var/www/html/
17 #indicador que se utilizara el puerto 80
18 EXPOSE 80
19 #iniciar apache
20 CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

docker build -t apachephp .

```
usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache2$ docker run -d -p 9001:80 --name contenedorApache3 image
4e2681bb14dbf7eb32c40d78c62758b083491ee27680433088bed708900a9d23
usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache2$
```



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

```

1 #!/bin/bash
2 echo "Borrar-All: 0 | Crear: 1 "
3 read bul1
4 if [ $bul1 -eq 1 ];
5     then
6         echo -p "Numeros de copias: "
7         read int1
8         for (( count=$int1; count>0; count-- ))
9             do
10                 docker run -d -p 8$count:8080 --name apachephp$count apachephp
11             done
12     else
13         echo -p "Numeros para borrar: "
14         read int1
15         for (( count=$int1; count>0; count-- ))
16             do
17                 docker rm -f apachephp$count
18             done
19     fi
20

```

```

usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache$ bash lanzadorDockerfile.bash
Borrar-All: 0 | Crear: 1
5
-p Numeros de copias:
5
ff8c631b5bb788b896575ac3a41aa7235bc58254668abb05128acdcd38dc8cf0
a8bc07158167e1ce1869b390ab5f5155c24b990241fa4d84529707ecb9412876
2b7c973d90a855c192efee7f435b49bb1ba0df46041d1f555a2fd24d3247f5bf
a3e4aa5b9f06704c50280cf70909be94d430e922af26a3087d49ca09d5c7399e
0989be8b09ea7c761409001648abba4cf036fe6675011784cce7182a955b13f7
usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
0989be8b09ea   apachephp     "/usr/sbin/apache2ct..." 10 seconds ago Up 9 seconds  80/tcp, 0.0.0.0:81->8080/tcp, :::81->8080/tcp  apachephp1
a3e4aa5b9f06   apachephp     "/usr/sbin/apache2ct..." 10 seconds ago Up 10 seconds  80/tcp, 0.0.0.0:82->8080/tcp, :::82->8080/tcp  apachephp2
2b7c973d90a8   apachephp     "/usr/sbin/apache2ct..." 10 seconds ago Up 10 seconds  80/tcp, 0.0.0.0:83->8080/tcp, :::83->8080/tcp  apachephp3
a8bc07158167   apachephp     "/usr/sbin/apache2ct..." 11 seconds ago Up 10 seconds  80/tcp, 0.0.0.0:84->8080/tcp, :::84->8080/tcp  apachephp4
ff8c631b5bb7   apachephp     "/usr/sbin/apache2ct..." 11 seconds ago Up 11 seconds  80/tcp, 0.0.0.0:85->8080/tcp, :::85->8080/tcp  apachephp5

```

```

usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache$ bash lanzadorDockerfile.bash
Borrar-All: 0 | Crear: 1
0
-p Numeros para borrar:
5
apachephp5
apachephp4
apachephp3
apachephp2
apachephp1
usuario@pps:~/Escritorio/dockerejemplo/dockerfilephp/dockerApache$

```