

**CIFP Rodolfo Ucha Piñeiro**  
**Puesta en Producción Segura**

**PRÁCTICA 2: SECDEVOPS**

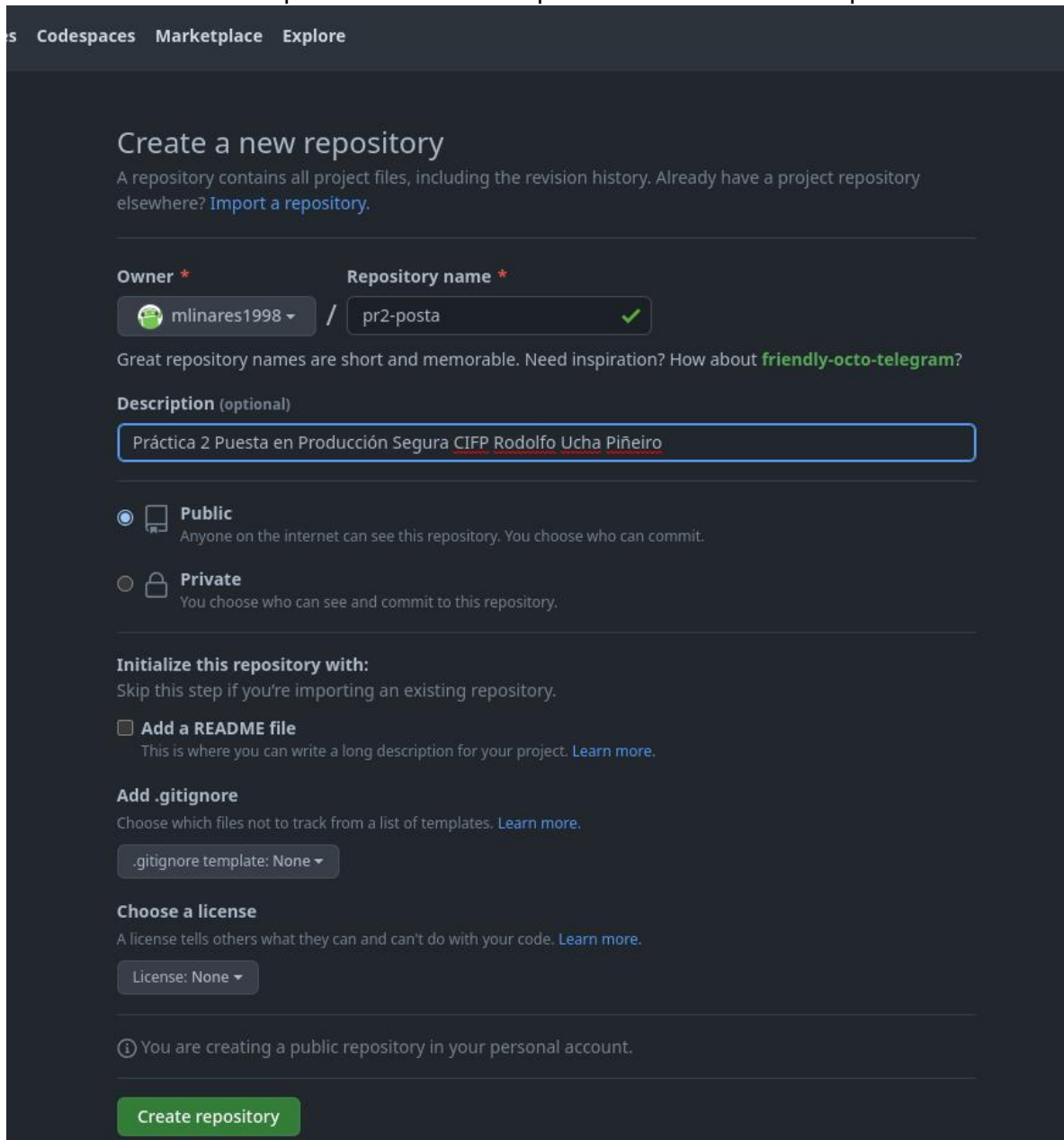
**Nombre: Miguel Angel Jaimes Linares**  
**Curso: CE Ciberseguridad en entornos TI**

## Contenido

Crear nuestro repositorio en GitHub.....	3
Crear Dockerfile que copia la aplicación directamente desde el anfitrión .....	8
Crear Dockerfile que prepara la aplicación desde una imagen de Ubuntu.....	12

## Crear nuestro repositorio en GitHub

- Creamos un nuevo repositorio en GitHub para la realización de la práctica.



The screenshot shows the GitHub 'Create a new repository' page. At the top, there are links for 'Codespaces', 'Marketplace', and 'Explore'. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

The form includes two main sections: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'mlinares1998'. The 'Repository name' field contains 'pr2-posta' with a green checkmark. Below this, a message says: 'Great repository names are short and memorable. Need inspiration? How about [friendly-octo-telegram?](#)'

The 'Description (optional)' field contains the text: 'Práctica 2 Puesta en Producción Segura CIPF Rodolfo Ucha Piñeiro'.

There are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.'

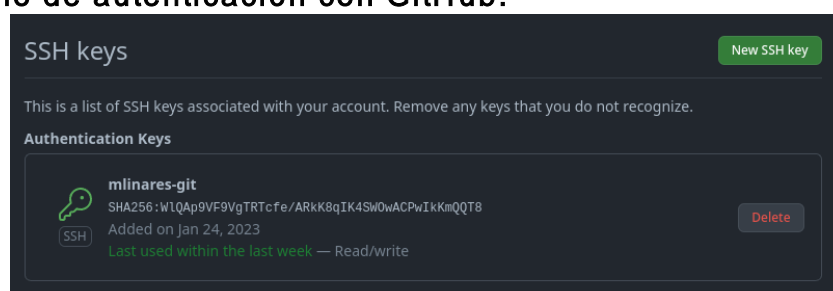
Under 'Initialize this repository with:', there is a checkbox for 'Add a README file' (checked) with a subtext: 'This is where you can write a long description for your project. [Learn more.](#)'

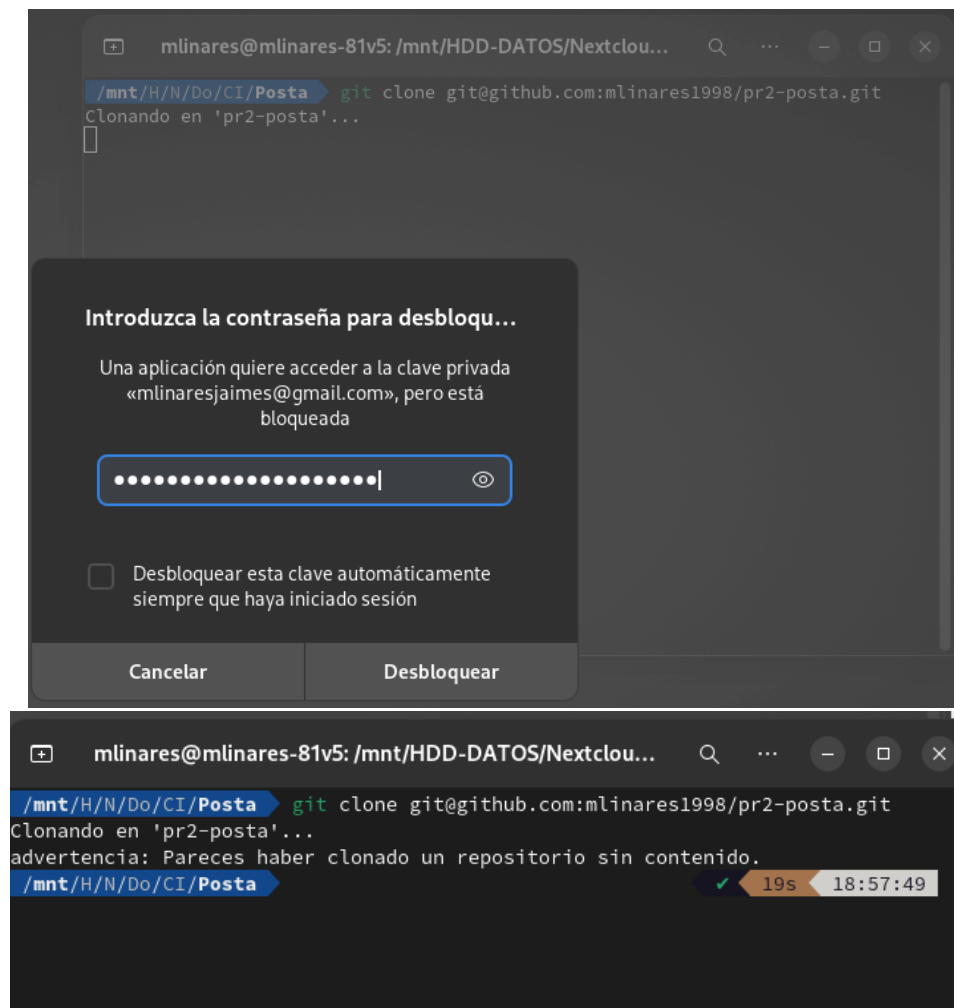
Below that is the 'Add .gitignore' section with a subtext: 'Choose which files not to track from a list of templates. [Learn more.](#)' and a dropdown menu set to '.gitignore template: None'.

The 'Choose a license' section has a subtext: 'A license tells others what they can and can't do with your code. [Learn more.](#)' and a dropdown menu set to 'License: None'.

At the bottom, there is an information icon and text: 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

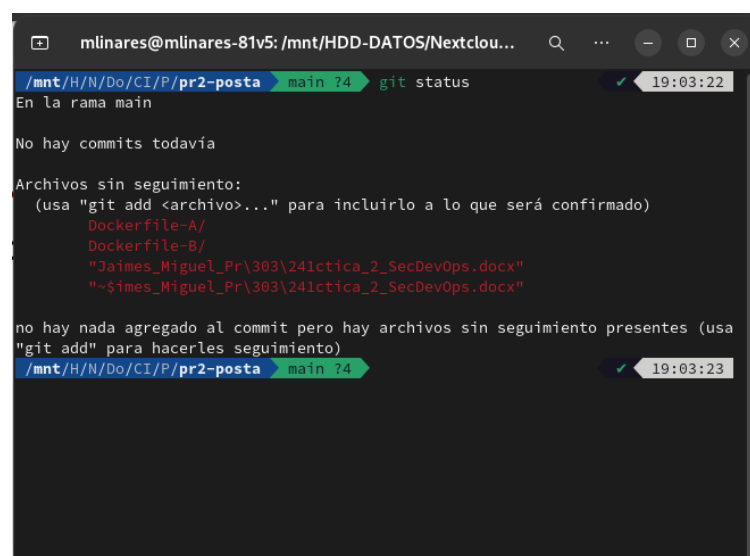
- Clonamos el repositorio en local usando git clone  
`git clone git@github.com:mlinares1998/pr2-posta.git`  
**NOTA:** Estoy usando una clave SSH asociada a mi cuenta GitHub como mecanismo de autenticación con GitHub.





Nos ha clonado el repositorio vacío, listo para poder agregar nuestros archivos.

- Usamos **git status** para ver el estado de nuestro repo local.



- Agregamos los ficheros requeridos a la zona de staging (Memoria y Dockerfile vacíos)  
**git add Dockerfile-A/ Dockerfile-B Jaimes\_Miguel\_Práctica\_2\_SecDevOps.docx**

```

mлинаres@mлинаres-81v5: /mnt/HDD-DATOS/Nextclou...
Dockerfile-B/
"Jaimes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"
"~$imes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa
"git add" para hacerles seguimiento)
/mnt/H/N/Do/CI/P/pr2-posta main ?4 git add Dockerfile-A/ Dockerfile-B Jaime
s Miguel Práctica 2 SecDevOps.docx
/mnt/H/N/Do/CI/P/pr2-posta main +3 ?1 git status 19:04:39
En la rama main

No hay commits todavía

Cambios a ser confirmados:
(usa "git rm --cached <archivo>..." para sacar del área de stage)
nuevos archivos: Dockerfile-A/Dockerfile.yml
nuevos archivos: Dockerfile-B/Dockerfile.yml
nuevos archivos: "Jaimes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
"~$imes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

/mnt/H/N/Do/CI/P/pr2-posta main +3 ?1 19:04:42

```

- Hacemos commit de nuestros cambios con **git commit**

```

mлинаres@mлинаres-81v5: /mnt/HDD-DATOS/Nextclou...

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
"~$imes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

/mnt/H/N/Do/CI/P/pr2-posta main +3 !1 ?1 git add Dockerfile-A/ Dockerfile-B
Jaimes Miguel Práctica 2 SecDevOps.docx
/mnt/H/N/Do/CI/P/pr2-posta main +3 ?1 git status 19:05:26
En la rama main

No hay commits todavía

Cambios a ser confirmados:
(usa "git rm --cached <archivo>..." para sacar del área de stage)
nuevos archivos: Dockerfile-A/Dockerfile.yml
nuevos archivos: Dockerfile-B/Dockerfile.yml
nuevos archivos: "Jaimes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que será confirmado)
"~$imes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"

/mnt/H/N/Do/CI/P/pr2-posta main +3 ?1 git commit 19:05:29

```

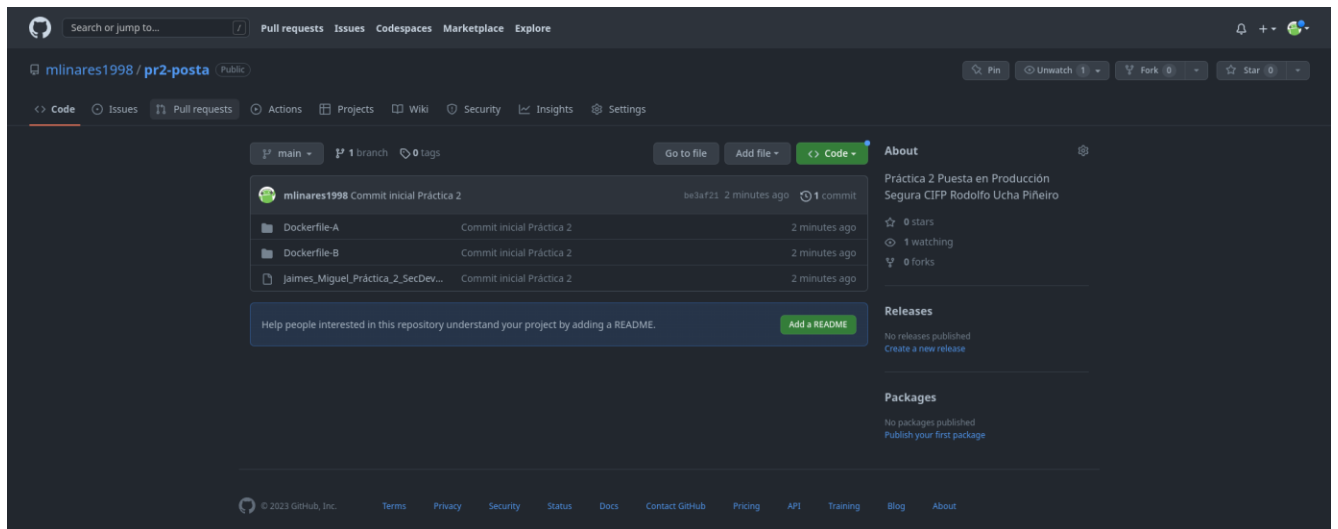
```
mлинаres@mлинаres-81v5: /mnt/HDD-DATOS/Nextclou...
...DATOS/Nextcloud/Documents/CIFP/Posta/pr2-posta/.git/COMMIT_EDITMSG Modificado
Commit inicial Práctica 2
# Por favor ingresa el mensaje del commit para tus cambios. Las
# líneas que comiencen con '#' serán ignoradas, y un mensaje
# vacío aborta el commit.
#
# En la rama main
#
# Confirmación inicial
#
# Cambios a ser confirmados:
#   nuevos archivos: Dockerfile-A/Dockerfile.yml
#   nuevos archivos: Dockerfile-B/Dockerfile.yml
#   nuevos archivos: "Jaimes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"
#
# Archivos sin seguimiento:
#   "~$imes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"
#

^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
```

```
/mnt/H/N/Do/CI/P/pr2-posta main +3 !1 ?1 git commit ✓ 19:08:07
[main (commit-raíz) be3af21] Commit inicial Práctica 2
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Dockerfile-A/Dockerfile.yml
create mode 100644 Dockerfile-B/Dockerfile.yml
create mode 100644 "Jaimes_Miguel_Pr\303\241ctica_2_SecDevOps.docx"
/mnt/H/N/Do/CI/P/pr2-posta main !1 ?1 ✓ 12s 19:08:21
```

- Usamos **git push** para subir nuestros cambios al repo.

```
mлинаres@mлинаres-81v5: /mnt/HDD-DATOS/Nextclou...
/mnt/H/N/Do/CI/P/pr2-posta main !1 ?1 git push origin ✓ 19:09:10
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (5/5), 405.76 KiB | 2.16 MiB/s, listo.
Total 5 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:mлинаres1998/pr2-posta.git
* [new branch] main -> main
/mnt/H/N/Do/CI/P/pr2-posta main !1 ?1 ✓ 4s 19:09:17
```



URL del repo: <https://github.com/mlinares1998/pr2-posta>

## Crear Dockerfile que copia la aplicación directamente desde el anfitrión

- Crea un Dockerfile que partiendo de una imagen PHP genera una imagen que copia una aplicación en PHP a un directorio del contenedor. Esta aplicación se debe copiar directamente desde un directorio del anfitrión.

Usaremos la siguiente aplicación demo:

<https://github.com/banago/simple-php-website>

- Instalamos las dependencias necesarias con  
**apt install php8.1 composer php8.1-dom php8.1-curl**

```
usuario@pps: ~/Escritorio
usuario@pps:~/Escritorio$ sudo apt install php8.1 composer php8.1-dom php8.1-curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «php8.1-xml» en lugar de «php8.1-dom»
composer ya está en su versión más reciente (1.10.1-1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libapache2-mod-php7.4 libgstreamer-plugins-bad1.0-0 libva-wayland2
linux-headers-5.15.0-46-generic linux-hwe-5.15-headers-5.15.0-46 linux-image-5.15.0-46-generic linux-modules-5.15.0-46-generic
linux-modules-extra-5.15.0-46-generic php7.4
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
libapache2-mod-php8.1 libpcre2-8-0 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Paquetes sugeridos:
php-pear
Se instalarán los siguientes paquetes NUEVOS:
libapache2-mod-php8.1 php8.1 php8.1-cli php8.1-common php8.1-curl php8.1-opcache php8.1-readline php8.1-xml
Se actualizarán los siguientes paquetes:
libpcre2-8-0
1 actualizados, 8 nuevos se instalarán, 0 para eliminar y 33 no actualizados.
Se necesita descargar 4.671 kB de archivos.
Se utilizarán 21,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

```
usuario@pps: ~/Escritorio
usuario@pps:~/Escritorio$ php --version
PHP 8.1.14 (cli) (built: Jan 13 2023 10:43:22) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.14, Copyright (c) Zend Technologies
with Zend OPcache v8.1.14, Copyright (c), by Zend Technologies
usuario@pps:~/Escritorio$
```

- Clonamos el repositorio

```
usuario@pps: ~/Escritorio
usuario@pps:~/Escritorio$ git clone https://github.com/banago/simple-php-website
Clonando en 'simple-php-website'...
remote: Enumerating objects: 114, done.
remote: Total 114 (delta 0), reused 0 (delta 0), pack-reused 114
Recibiendo objetos: 100% (114/114), 19.79 KiB | 413.00 KiB/s, listo.
Resolviendo deltas: 100% (48/48), listo.
usuario@pps:~/Escritorio$
usuario@pps:~/Escritorio$
```



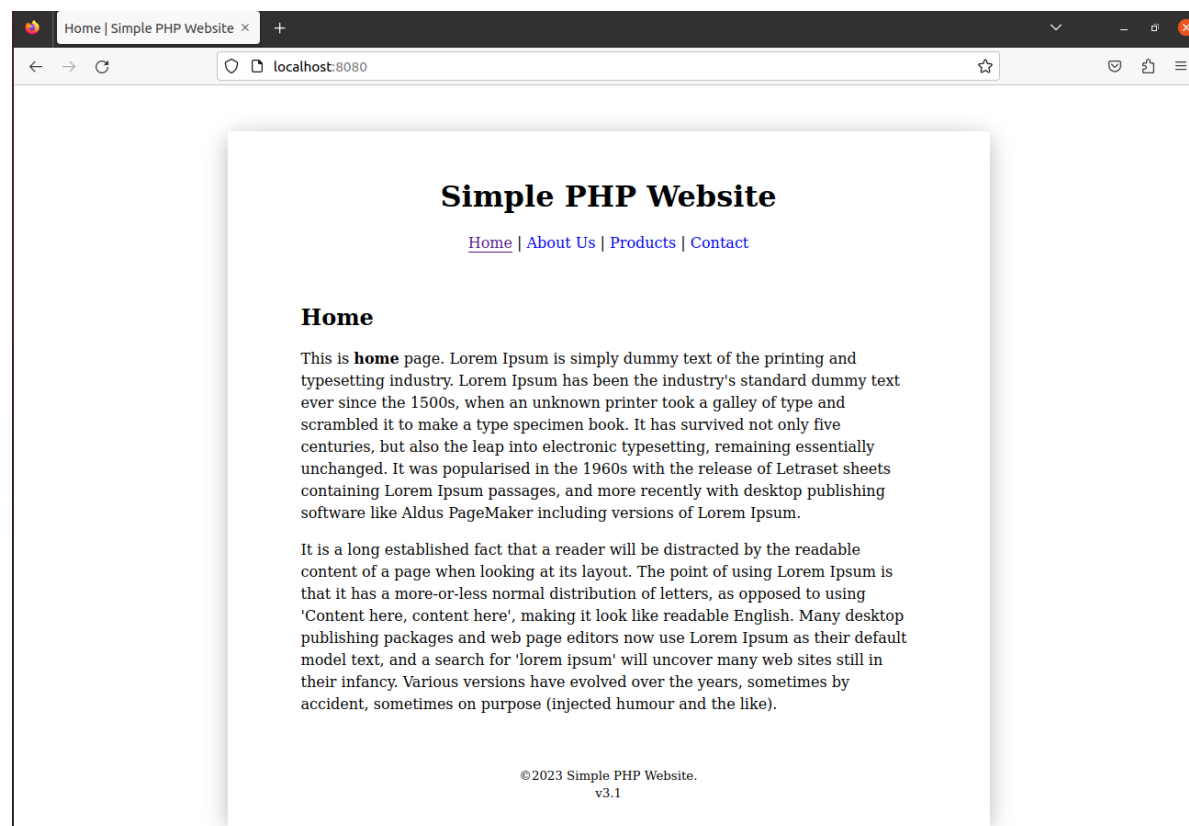
- Lanzamos la aplicación con el servidor de desarrollo de PHP para comprobar que funciona.

**php -S localhost:8080**

```

Actividades Terminal 27 de ene 17:41
usuario@pps: ~/Escritorio/simple-php-website

usuario@pps:~/Escritorio/simple-php-website$ php -S localhost:8080
[Fri Jan 27 17:40:03 2023] PHP 8.1.14 Development Server (http://localhost:8080) started
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40226 Accepted
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40226 [200]: GET /
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40226 Closing
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40236 Accepted
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40236 [200]: GET /template/style.css
[Fri Jan 27 17:40:22 2023] 127.0.0.1:40236 Closing
  
```



- Creamos nuestro fichero Dockerfile (En mi caso directamente usando nano).

```

Actividades Terminal 27 de ene 17:45
usuario@pps: ~/Escritorio/simple-php-website

GNU nano 4.8 Dockerfile
FROM php:8.1-rc-apache
WORKDIR /var/www/html

COPY index.php index.php
COPY template/ template
COPY includes/ includes
COPY content/ content

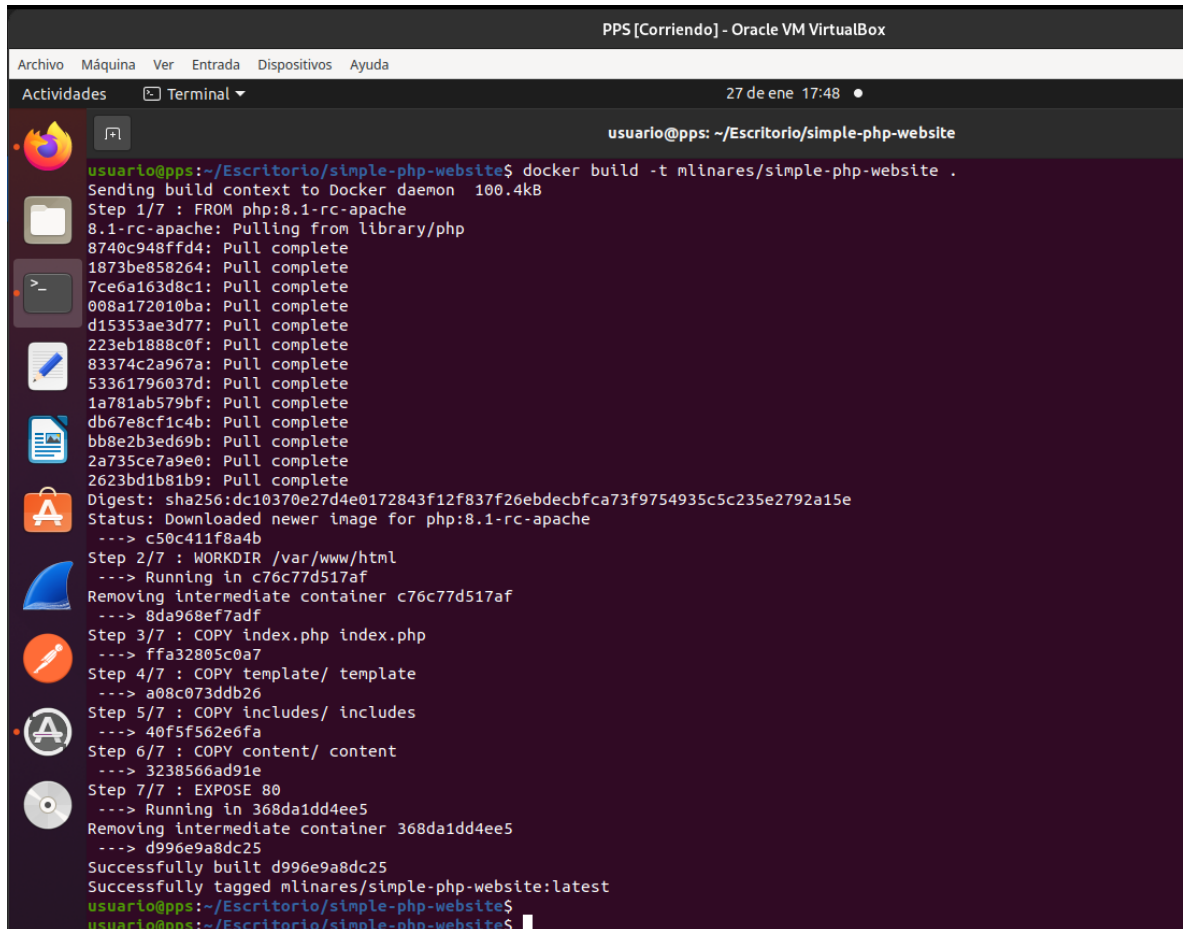
EXPOSE 80
  
```

Este Dockerfile usara la imagen de PHP 8.1 con el servidor Apache pre-configurado, copiara el fichero index.php y carpetas requeridas a la carpeta /var/www/html del contenedor. Finalmente expone el servicio desde el puerto 80.

- Construimos la imagen.

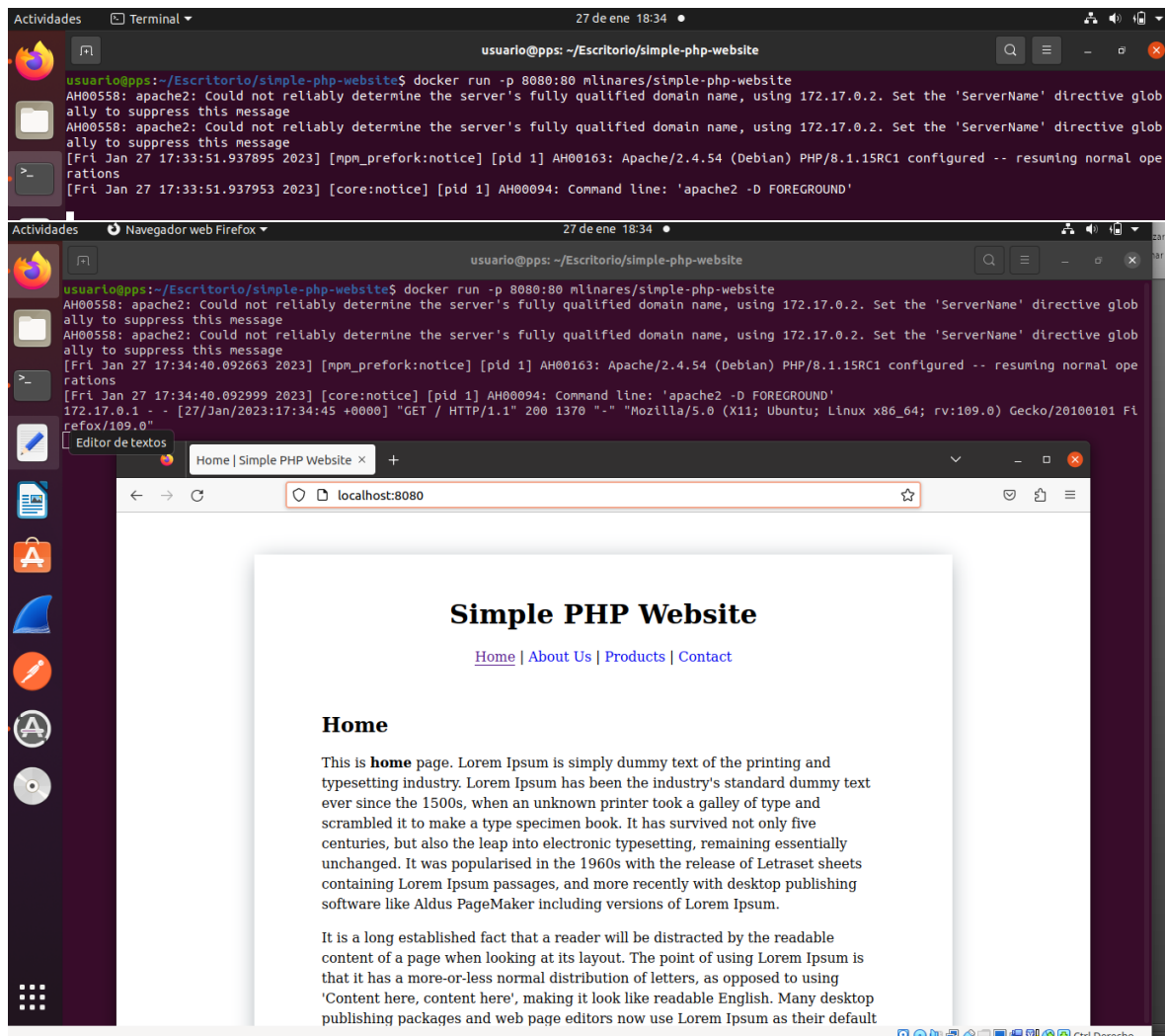
Usamos el comando **docker build -t mlinares/simple-php-website**

Nos construirá una imagen con el nombre especificado usando el Dockerfile de la carpeta en la que estamos posicionados.



```
PPS [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades 27 de ene 17:48
usuario@pps: ~/Escritorio/simple-php-website
usuario@pps:~/Escritorio/simple-php-website$ docker build -t mlinares/simple-php-website .
Sending build context to Docker daemon 100.4kB
Step 1/7 : FROM php:8.1-rc-apache
8.1-rc-apache: Pulling from library/php
8740c948ff4d: Pull complete
1873be858264: Pull complete
7ce6a163d8c1: Pull complete
008a172010ba: Pull complete
d15353ae3d77: Pull complete
223eb1888c0f: Pull complete
83374c2a967a: Pull complete
53361796037d: Pull complete
1a781ab579bf: Pull complete
db67e8cf1c4b: Pull complete
bb8e2b3ed69b: Pull complete
2a735ce7a9e0: Pull complete
2623bd1b81b9: Pull complete
Digest: sha256:dc10370e27d4e0172843f12f837f26ebdecbfca73f9754935c5c235e2792a15e
Status: Downloaded newer image for php:8.1-rc-apache
--> c50c411f8a4b
Step 2/7 : WORKDIR /var/www/html
--> Running in c76c77d517af
Removing intermediate container c76c77d517af
--> 8da968ef7adf
Step 3/7 : COPY index.php index.php
--> ffa32805c0a7
Step 4/7 : COPY template/ template
--> a08c073ddb26
Step 5/7 : COPY includes/ includes
--> 40f5f562e6fa
Step 6/7 : COPY content/ content
--> 3238566ad91e
Step 7/7 : EXPOSE 80
--> Running in 368da1dd4ee5
Removing intermediate container 368da1dd4ee5
--> d996e9a8dc25
Successfully built d996e9a8dc25
Successfully tagged mlinares/simple-php-website:latest
usuario@pps:~/Escritorio/simple-php-website$
usuario@pps:~/Escritorio/simple-php-website$
```

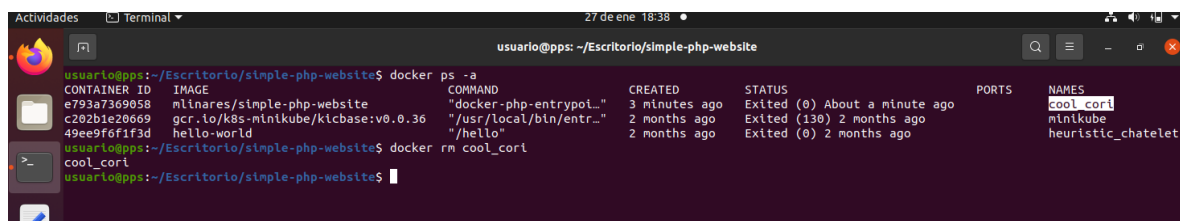
- Lanzamos la imagen y comprobamos que funciona correctamente.  
**docker run -p 8080:80 mlinares/simple-php-website**  
 Con el atributo **-p** enlazamos el puerto 8080 de nuestra máquina real con el puerto 80 interno del contenedor.



- Paramos el contenedor pulsando **CTRL+C** debido a que automáticamente hemos entrado en modo attached / interactivo.

```
172.17.0.1 - - [27/Jan/2023:17:34:55 +0000] "GET / HTTP/1.1" 200 1370 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0"
^C[Fri Jan 27 17:36:27.163840 2023] [mpm_prefork:notice] [pid 1] AH00169: caught SIGTERM, shutting down
usuario@pps: ~/Escritorio/simple-php-website$
```

- Eliminamos el contenedor  
 Usamos **docker ps -a** para obtener el slug del contenedor y su id (`cool_cori`)  
 Luego eliminamos el contenedor usando **docker rm cool\_cori**

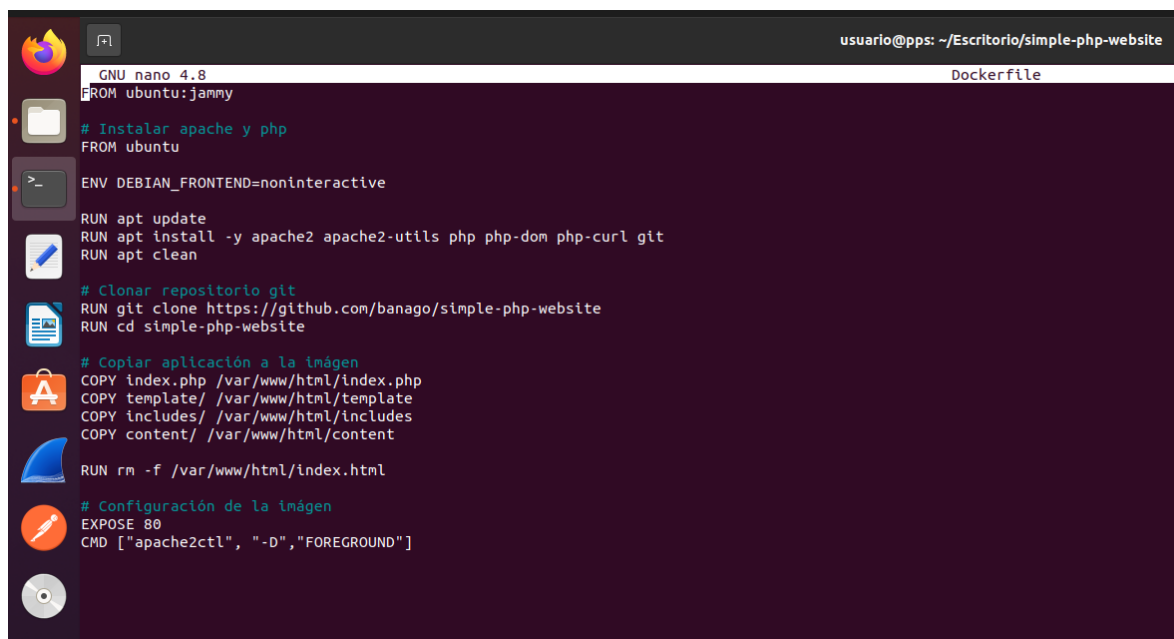


## Crear Dockerfile que prepara la aplicación desde una imagen de Ubuntu

- 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 en PHP al directorio de Apache que expone las páginas web (Usamos la misma anterior). 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).

Este Dockerfile parte de una imagen de Ubuntu 22.04, la variable DEBIAN\_FRONTEND previene que al construir la imagen la shell pueda solicitar un input al usuario como ocurre al actualizar TZDATA.

Posteriormente se instala Apache, PHP y git, se clona el repositorio, se copian los archivos en /var/www/html, se elimina el html por defecto incluido en apache y se empaqueta la aplicación, exponiendo el puerto 80 y lanzando Apache con CMD.



```
GNU nano 4.8
FROM ubuntu:jammy

# Instalar apache y php
FROM ubuntu

ENV DEBIAN_FRONTEND=noninteractive

RUN apt update
RUN apt install -y apache2 apache2-utils php php-dom php-curl git
RUN apt clean

# Clonar repositorio git
RUN git clone https://github.com/banago/simple-php-website
RUN cd simple-php-website

# Copiar aplicación a la imagen
COPY index.php /var/www/html/index.php
COPY template/ /var/www/html/template
COPY includes/ /var/www/html/includes
COPY content/ /var/www/html/content

RUN rm -f /var/www/html/index.html

# Configuración de la imagen
EXPOSE 80
CMD ["apache2ctl", "-D","FOREGROUND"]
```

- Construimos la imagen con el comando `docker build -t mlinares/simple-php-website-ubuntu .`

```

Actividades Terminal 27 de ene 20:14
usuario@pps: ~/Escritorio/simple-php-website

usuario@pps:~/Escritorio/simple-php-website$ docker build -t mlinares/simple-php-website-ubuntu .
Sending build context to Docker daemon 100.9kB
Step 1/15 : FROM ubuntu:jammy
----> 6b7dfa7e8fdb
Step 2/15 : FROM ubuntu
----> 6b7dfa7e8fdb
Step 3/15 : ENV DEBIAN_FRONTEND=noninteractive
----> Using cache
----> 32b94bbe7039
Step 4/15 : RUN apt update
----> Using cache
----> e3a5a53509a2
Step 5/15 : RUN apt install -y apache2 apache2-utils php php-dom php-curl git
----> Running in 72664ff7ac4c

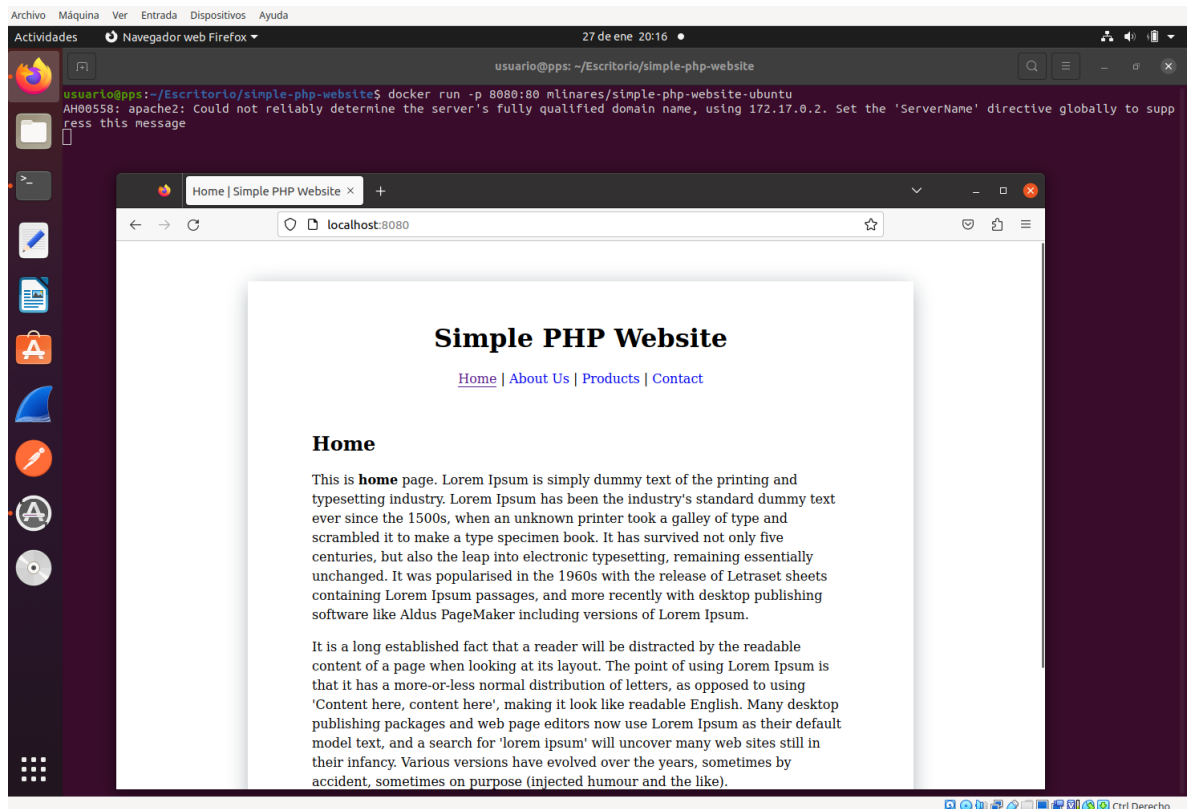
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  apache2-bin apache2-data bzip2 ca-certificates file git-man less
  libapache2-mod-php8.1 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libargon2-1 libbrotli1 libbsd0 libcbor0.8 libcurl3-gnutls
  libcurl4 libedit2 liberror-perl libexpat1 libfido2-1 libgdbm-compat4
  libgdbm6 libicu70 libjansson4 libldap-2.5-0 libldap-common liblua5.3-0
  libmagic-mgc libmagic1 libmd0 libnghttp2-14 libperl5.34 libpsl5 librtmp1
  libsasl2-2 libsasl2-modules libsasl2-modules-db libsodium23 libsqlite3-0
  libssh-4 libx11-6 libx11-data libxau6 libxcb1 libxdmcp6 libxext6 libxml2
  libxmuu1 libxslt1.1 mailcap media-types mime-support netbase openssh-client
  openssl patch perl perl-modules-5.34 php-common php8.1 php8.1-cli
  php8.1-common php8.1-curl php8.1-opcache php8.1-readline psmisc publicsuffix
  ssl-cert tzdata ucf xauth xz-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser ufw
  bzip2-doc gettext-base git-daemon-run | git-daemon-sysvinit git-doc
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn php-pear
  gdbm-l10n libsasl2-modules-gssapi-mit | libsasl2-modules-gssapi-heimdal
  libsasl2-modules-ldap libsasl2-modules-otp libsasl2-modules-sql keychain
  libpam-ssh monkeysphere ssh-askpass ed diffutils-doc perl-doc
  libterm-readline-gnu-perl | libterm-readline-perl-perl make
  libtap-harness-archive-perl

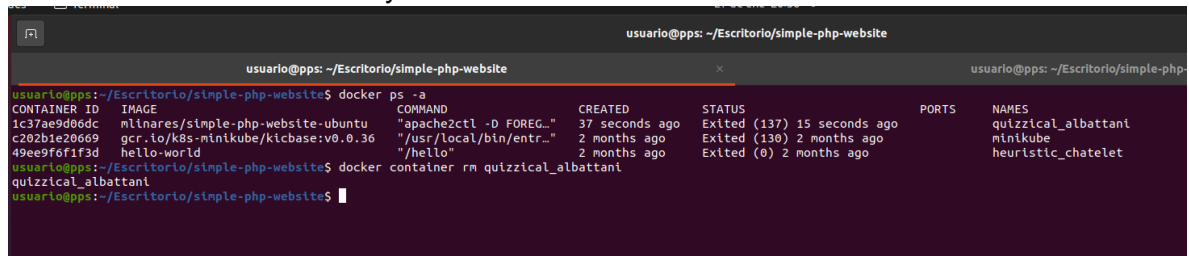
Removing intermediate container cb64268dd651
----> 5fff41932144
Step 8/15 : RUN cd simple-php-website
----> Running in ba51f4e6ac24
Removing intermediate container ba51f4e6ac24
----> 24f5a8409c59
Step 9/15 : COPY index.php /var/www/html/index.php
----> 03351049d0ec
Step 10/15 : COPY template/ /var/www/html/template
----> 42b45b08ed39
Step 11/15 : COPY includes/ /var/www/html/includes
----> ee5b8e0e0a9f
Step 12/15 : COPY content/ /var/www/html/content
----> 832f5baa4eac
Step 13/15 : RUN rm -f /var/www/html/index.html
----> Running in d72ca35aa71d
Removing intermediate container d72ca35aa71d
----> 73feda63c130
Step 14/15 : EXPOSE 8080
----> Running in c63f85099fad
Removing intermediate container c63f85099fad
----> 6e22f9384639
Step 15/15 : ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]
----> Running in 10051563d3a7
Removing intermediate container 10051563d3a7
----> 680b67ba3a7e
Successfully built 680b67ba3a7e
Successfully tagged mlinares/simple-php-website-ubuntu:latest
usuario@pps:~/Escritorio/simple-php-website$

```

- Levantamos la imagen con el comando  
`docker run -p 8080:80 mlinares/simple-php-website-ubuntu`



- Detenemos el contenedor y lo eliminamos



- 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.

Usamos el siguiente script de bash





- Observamos en la terminal que se han lanzado los 20 contenedores

```

des Terminal ▾

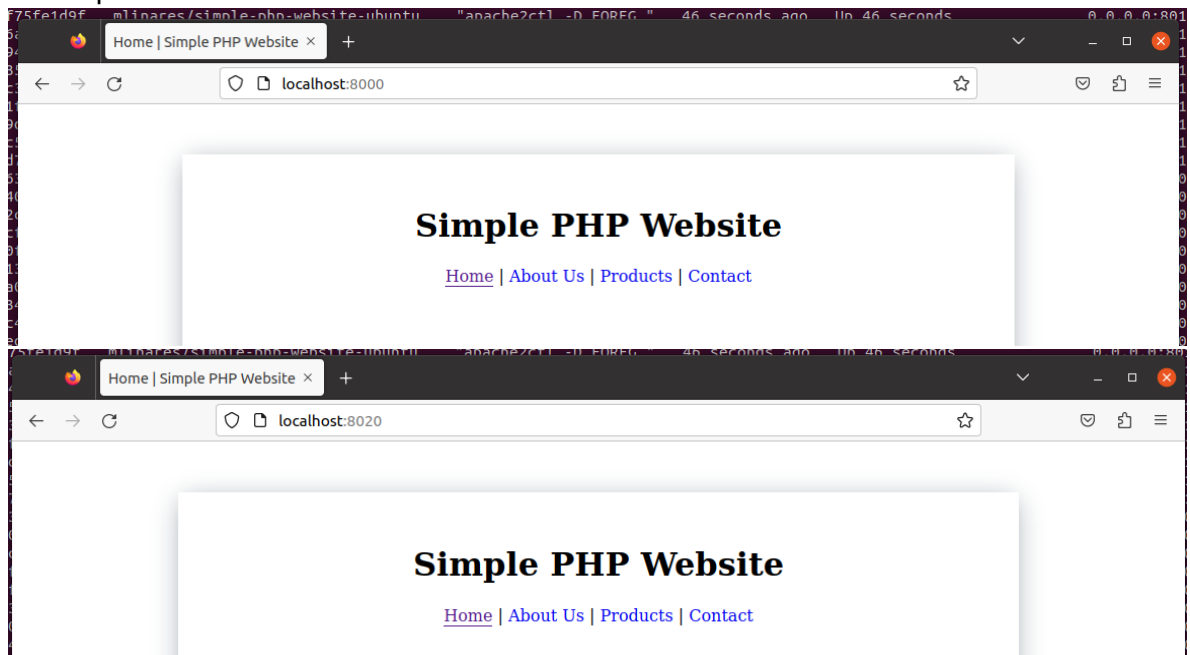
usuario@pps:~/Escritorio/simple-php-website$ bash launch-20.sh
8e1ed072f75a9a7076143f6b565daa550534d9b180ab6465a08d74a26d751970
e42c47705acf3810206d119deaba0b9d6098972d96780a11554fd5a9196e0a87
b9a343362c3e0b3039360d2c2e274cfae2bbd1a2715bef3d480f53101a1fb593
c7ea0f5a35156a8daf327792c45f6ca00485c1a361f0467152104cbf012d242
23413f92d0a3dc8fe969ead21febae4a26a7eb1d13b534d8c28f3f618aa068c8
6540fdd5e69d5f794f89fd76fd947c68444a229c629dc7abba1e5931bd59cf78
40ccf94c4a9479039b7d2e075ff3f4dd175ad218d142e283f02146fee070c558
1f72cf9e98ee2d2b8fcc7ed9df4160d94f2fe93b0541a9cc0625de084b4c2c59
57e4025bfa00df723b1d5c9d5bb1bc46b57944c02568970ca45308f388cc3b62
5cb6377adc785074dbd387839f68cb72c47021c1af5186218de9b9b1ea55652e
e0bd7b3d843af5f21c06e686ed4e075c0178ceac61ce78178fb44a67612ef4da
d90c52fd6ce34b1f4dddc4324dd5d54fa8641675ecf8edc7a24042af5049c0e6
2a69d6b596d32d4c3085b9918d7d5397d78fbdcb43dbde5b765c00be4f3b7ebd
ae81fcd0edb43360f0500bd5d64247a01176f1ebbaa15d7e2be054313fa443e3
013c37bd91915d7c041efb94de97202054284e7496169c09e3201aa4c0e7d011
7f435eb2f25bc72b67f8e38cd2b48a2472963b67b0f85da4a625e27a5d6dea84
e47946d75bf18b8ec2f492d99534b673bc01461ccdc4cc132a66a6f28546a9d8
f8b6a28f8c91a83c3c58097a2e9ec7c0d89473a5f3e00bc276ed65952c689719
1fdf75fe1d9fcaeb72b989c2d16cd6ce3638c0def321b70050372ac56e8b41f5
c00f277d806265d43e04c2a8dc30ddc243da1bbbed1c64c0ebbf1b2ae1c5d8df
072dec440b1861e739fff361441b7354750221e34812300aba8547085a1c8ba2
usuario@pps:~/Escritorio/simple-php-website$

usuario@pps:~/Escritorio/simple-php-website$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
072dec440b18	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	45 seconds ago	Up 45 seconds	0.0.0.0:8020->80/tcp, :::8020->80/tcp	modest_mclaren
c00f277d8062	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	46 seconds ago	Up 45 seconds	0.0.0.0:8019->80/tcp, :::8019->80/tcp	infallible_tausstg
1fd775fe1d9f	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	46 seconds ago	Up 46 seconds	0.0.0.0:8018->80/tcp, :::8018->80/tcp	infallible_agnest
f8b6a28f8c91	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	47 seconds ago	Up 46 seconds	0.0.0.0:8017->80/tcp, :::8017->80/tcp	sweet_alameda
e47946d75bf1	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	47 seconds ago	Up 47 seconds	0.0.0.0:8016->80/tcp, :::8016->80/tcp	vibrant_bartik
7f435eb2f25b	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	48 seconds ago	Up 47 seconds	0.0.0.0:8015->80/tcp, :::8015->80/tcp	mystifying_banach
013c37bd9191	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	48 seconds ago	Up 48 seconds	0.0.0.0:8014->80/tcp, :::8014->80/tcp	unruffled_robinson
ae81fcd0edb4	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	49 seconds ago	Up 48 seconds	0.0.0.0:8013->80/tcp, :::8013->80/tcp	inspiring_booth
2a69d6b596d3	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	49 seconds ago	Up 49 seconds	0.0.0.0:8012->80/tcp, :::8012->80/tcp	friendly_satoshi
d90c52fd6ce3	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	50 seconds ago	Up 49 seconds	0.0.0.0:8011->80/tcp, :::8011->80/tcp	loving_johnson
e0bd7b3d843a	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	50 seconds ago	Up 50 seconds	0.0.0.0:8010->80/tcp, :::8010->80/tcp	cool_banzai
5cb6377adc78	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	51 seconds ago	Up 50 seconds	0.0.0.0:8009->80/tcp, :::8009->80/tcp	recurring_spence
57e4025bfa00	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	51 seconds ago	Up 51 seconds	0.0.0.0:8008->80/tcp, :::8008->80/tcp	dreamy_shirley
40ccf94c4a94	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	52 seconds ago	Up 51 seconds	0.0.0.0:8007->80/tcp, :::8007->80/tcp	silly_turing
6540fdd5e69d	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	53 seconds ago	Up 52 seconds	0.0.0.0:8006->80/tcp, :::8006->80/tcp	boring_hertz
23413f92d0a3	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	53 seconds ago	Up 53 seconds	0.0.0.0:8005->80/tcp, :::8005->80/tcp	wonderful_sinoussi
c7ea0f5a3515	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	54 seconds ago	Up 53 seconds	0.0.0.0:8004->80/tcp, :::8004->80/tcp	gallant_sammet
b9a343362c3e	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	54 seconds ago	Up 54 seconds	0.0.0.0:8003->80/tcp, :::8003->80/tcp	pensive_kilby
e42c47705acf	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	55 seconds ago	Up 54 seconds	0.0.0.0:8002->80/tcp, :::8002->80/tcp	practical_antonnelli
8e1ed072f75a	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	55 seconds ago	Up 54 seconds	0.0.0.0:8001->80/tcp, :::8001->80/tcp	zen_hertz
8e1ed072f75a	mlinares/simple-php-website-ubuntu	"apache2ctl -D FOREG..."	55 seconds ago	Up 54 seconds	0.0.0.0:8000->80/tcp, :::8000->80/tcp	clever_kapitsa

- Intentamos acceder a los servidores alojados en los puertos 8000 y 8020 como prueba. Funcionan correctamente.



- Usamos estos dos comandos para detener y eliminar todos los contenedores.  
`docker stop $(docker ps -a -q)`  
`docker rm $(docker ps -a -q)`

```

usuario@pps: ~/Escritorio/simple-php-website$ docker stop $(docker ps -a -q)
e479e1bf5c89
11b52be9e726
8e350b62f2dc
c65ee0700391
b076a592fe7c
92f545f839a1
2a1d063fcfec
0c5a5ff0abb4
5a35f4b16bed
0d5189458b6f
ea3096696467
f56fd23ce29b
dd58655d6b09
bb46461b48c3
bf17438c0988
735bc7f9e6e5
7e2dfe5ee50c
58ecde15c667
df263e01b83b
f427bb2bc51f
cdb230c78f6e
usuario@pps: ~/Escritorio/simple-php-website$ docker rm $(docker ps -a -q)
e479e1bf5c89
11b52be9e726
8e350b62f2dc
c65ee0700391
b076a592fe7c
92f545f839a1
2a1d063fcfec
0c5a5ff0abb4
5a35f4b16bed
0d5189458b6f
ea3096696467
f56fd23ce29b
dd58655d6b09
bb46461b48c3
bf17438c0988
735bc7f9e6e5
7e2dfe5ee50c
58ecde15c667
df263e01b83b
f427bb2bc51f
cdb230c78f6e
usuario@pps: ~/Escritorio/simple-php-website$

```

```

usuario@pps: ~/Escritorio/simple-php-website$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
usuario@pps: ~/Escritorio/simple-php-website$

```