

CIFP Rodolfo Ucha Piñeiro
Puesta en Producción Segura

PRÁCTICA 2: SECDEVOPS

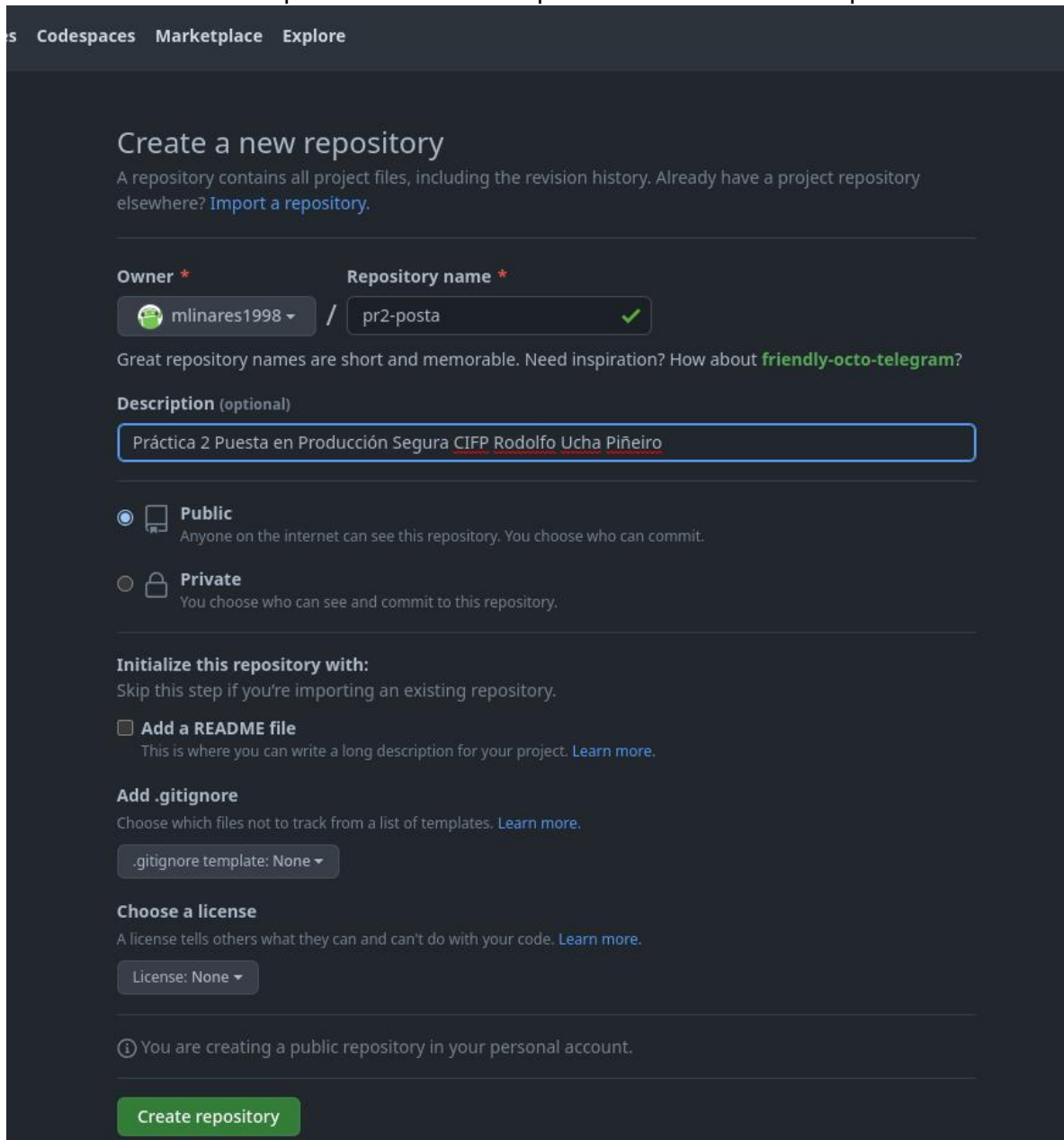
Nombre: Miguel Angel Jaimes Linares
Curso: CE Ciberseguridad en entornos TI

Contenido

Crear nuestro repositorio en GitHub.....	3
Crear Dockerfile copiando la aplicación directamente desde el anfitrión	8

Crear nuestro repositorio en GitHub

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





Codespaces Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner * **Repository name ***

 mlinares1998 / pr2-posta 

Great repository names are short and memorable. Need inspiration? How about [friendly-octo-telegram?](#)

Description (optional)

Práctica 2 Puesta en Producción Segura CIPF Rodolfo Ucha Piñeiro

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None

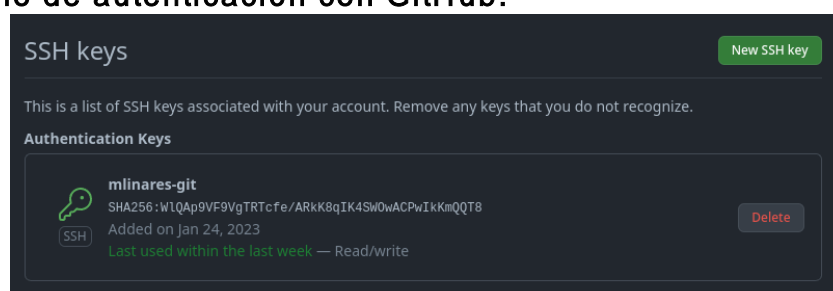
Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

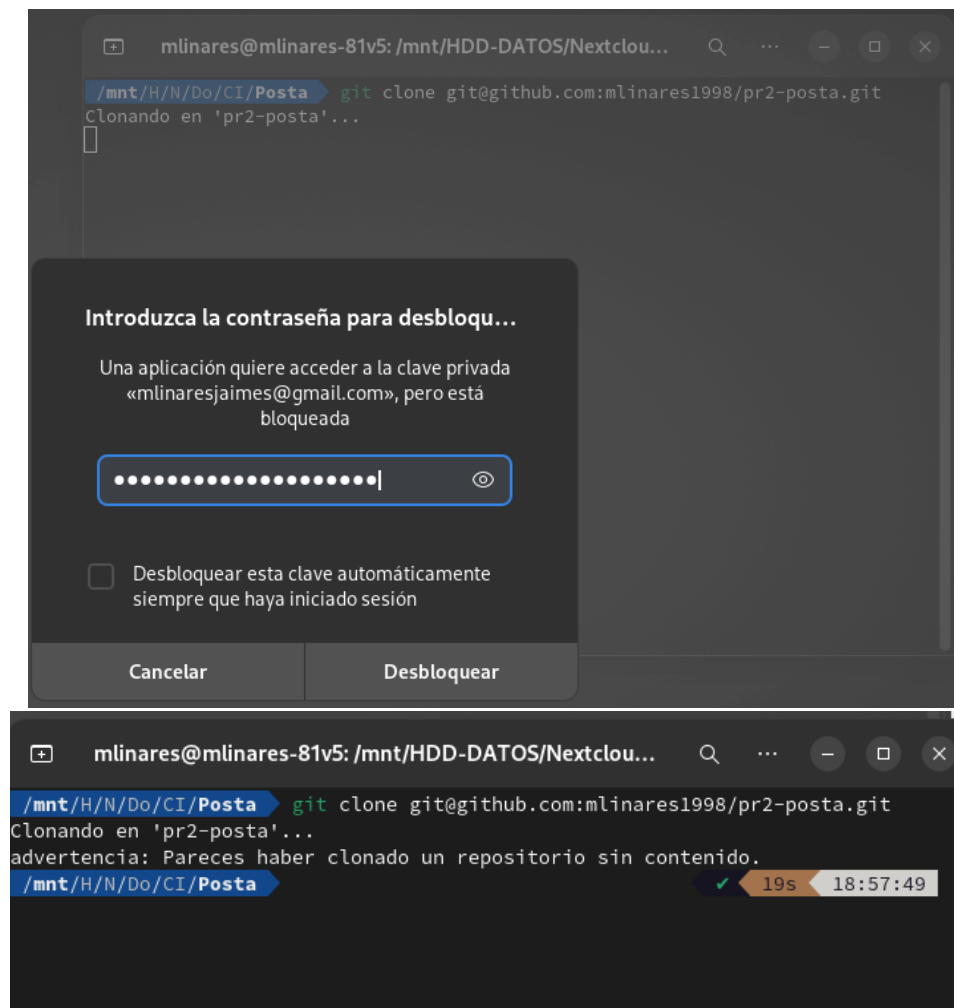
License: None

 You are creating a public repository in your personal account.

[Create repository](#)

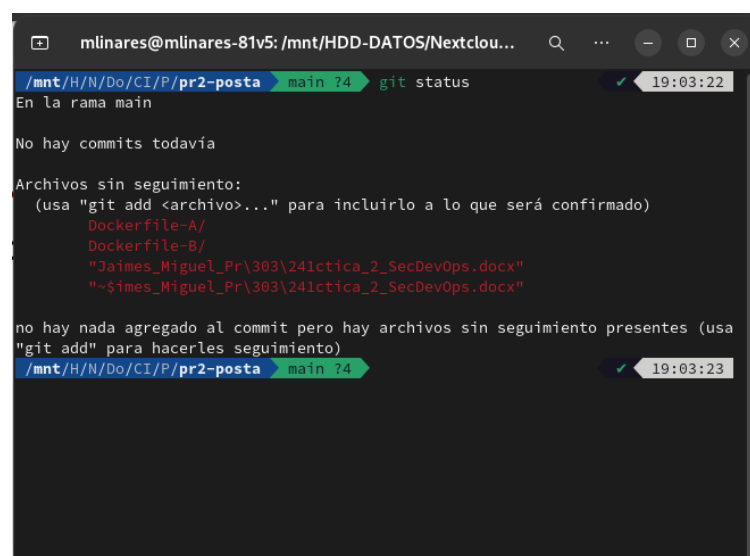
- Clonamos el repositorio en local usando git clone
git clone [git@github.com:mlinares1998/pr2-posta.git](https://github.com/mlinares1998/pr2-posta.git)
NOTA: Estoy usando una clave SSH asociada a mi cuenta GitHub como mecanismo de autenticación con GitHub.





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

```

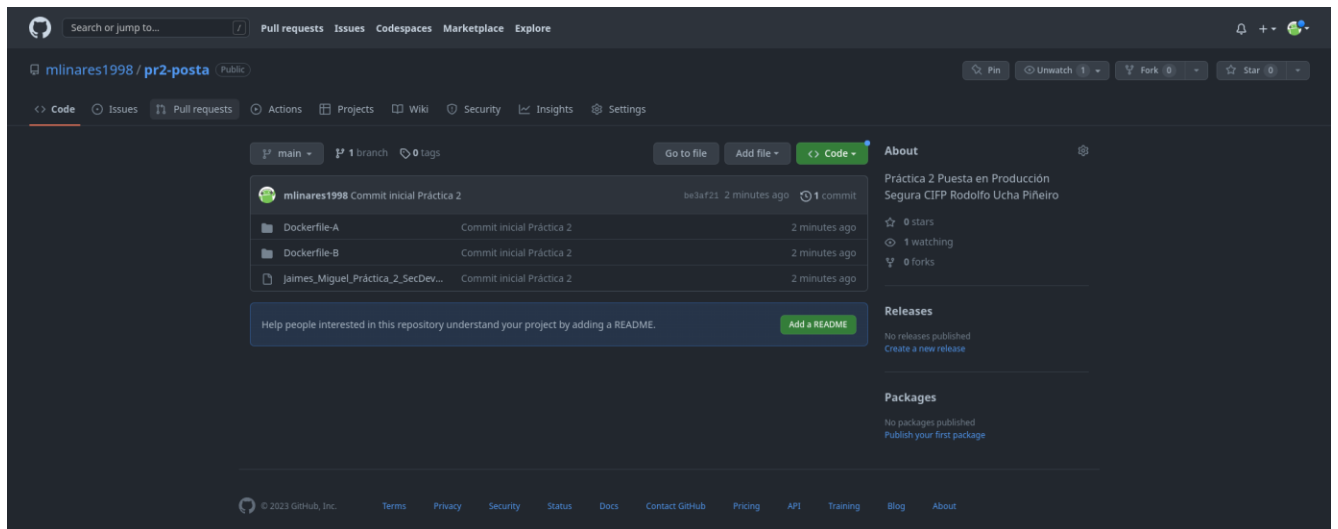
```
mлинаres@mлинаres-81v5: /mnt/HDD-DATOS/Nextclou...
...DATOS/Nextcloud/Documents/CIFP/Posta/pr2-posta/.git/COMMIT_EDITMSG Modificado
Commit inicial Práctica 2
# Por favor ingresa el mensaje del commit para tus cambios. Las
# líneas que comiencen con '#' serán ignoradas, y un mensaje
# vacío aborta 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 copiando 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

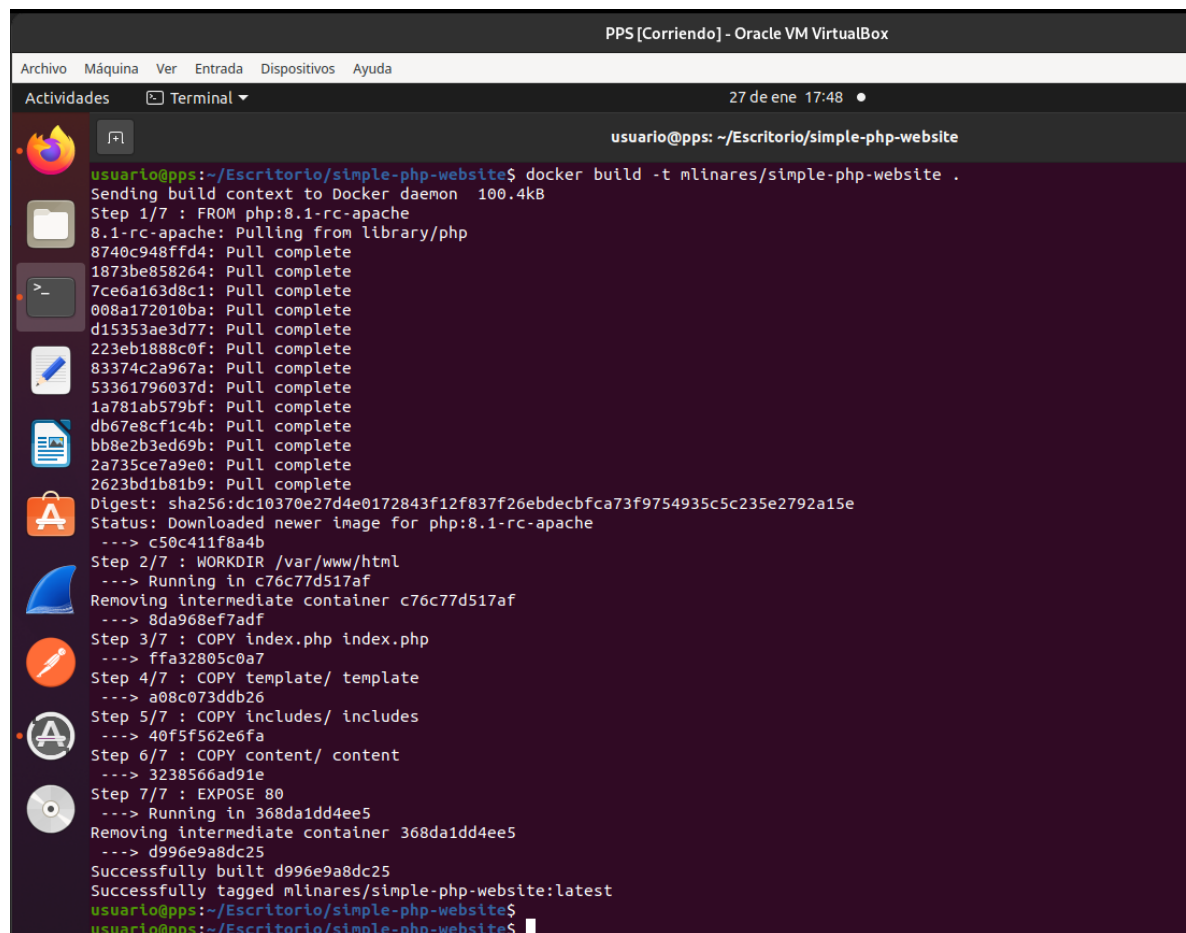
EXPOSE 80
  
```

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.

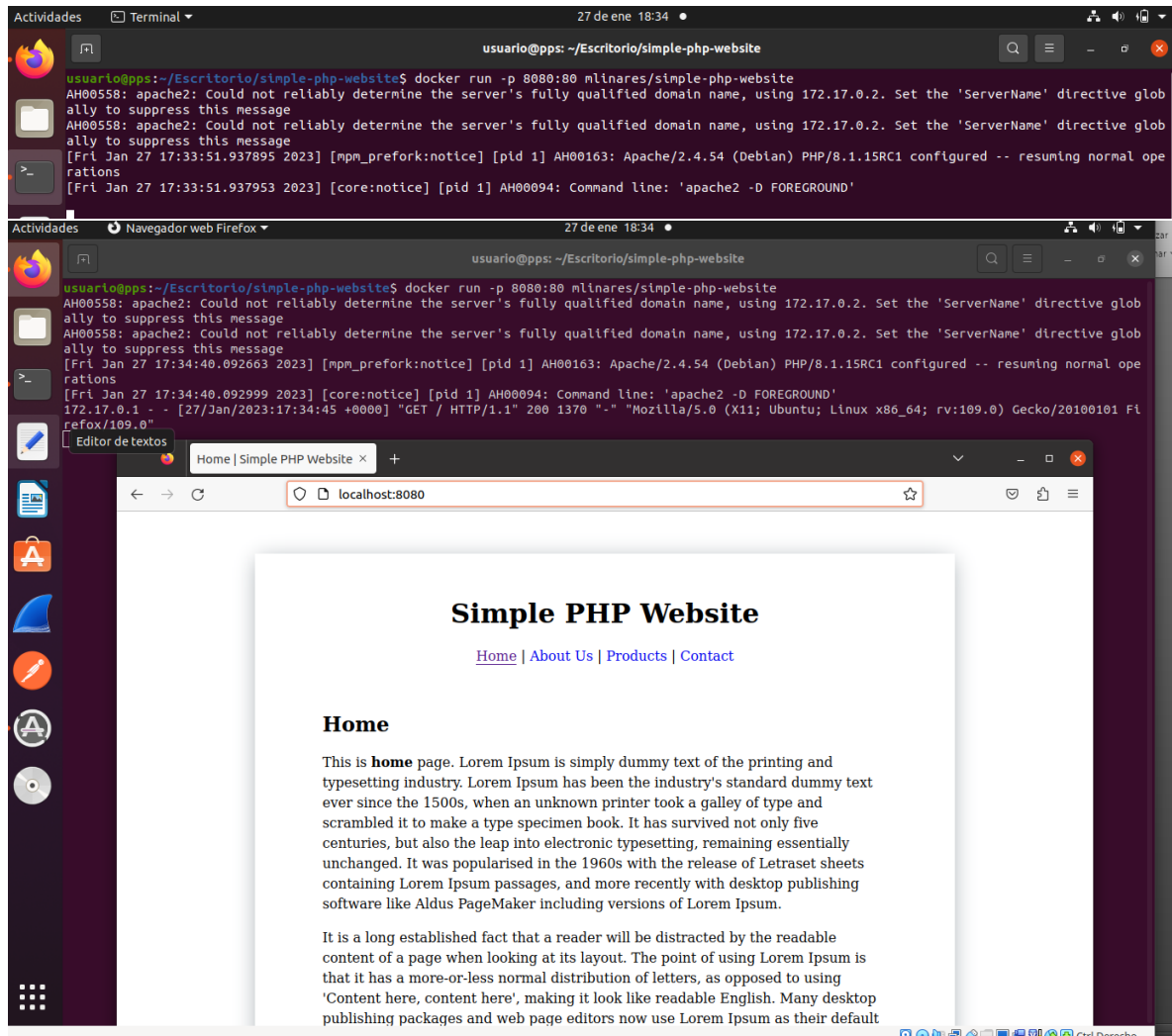
Usamos el comando **docker build -t mlinares/simple-php-website**

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

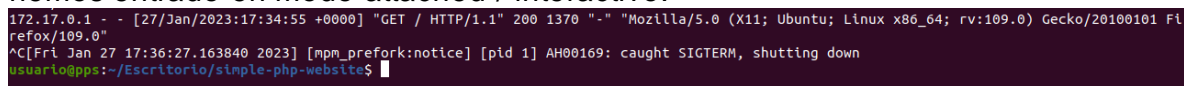


```
PPS [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades 27 de ene 17:48
usuario@pps: ~/Escritorio/simple-php-website
usuario@pps:~/Escritorio/simple-php-website$ docker build -t mlinares/simple-php-website .
Sending build context to Docker daemon 100.4kB
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.



- 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**

