**CIFP Rodolfo Ucha Piñeiro**

**Puesta en Producción Segura**

**PRÁCTICA 2: SECDEVOPS**

**Nombre: Miguel Angel Jaimes Linares**

**Curso: CE Ciberseguridad en entornos TI**

# 

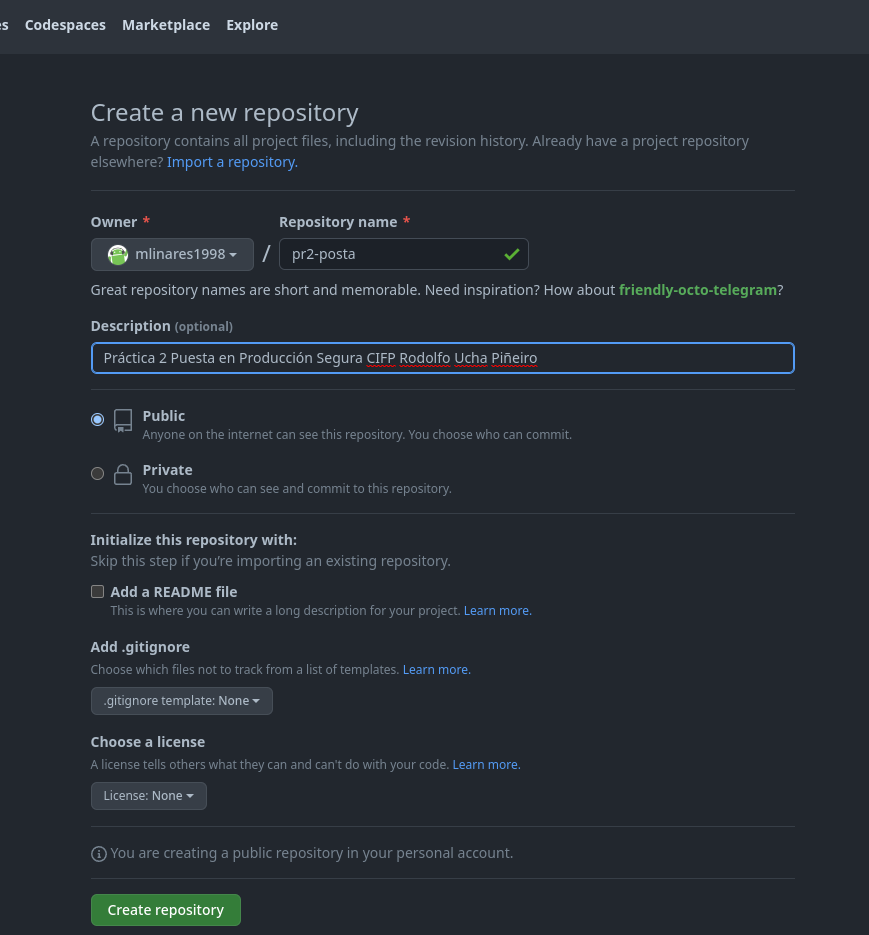
Contenido

[Crear nuestro repositorio en GitHub 3](#_Toc125737597)

[Crear Dockerfile copiando la aplicación directamente desde el anfitrión 8](#_Toc125737598)

# Crear nuestro repositorio en GitHub

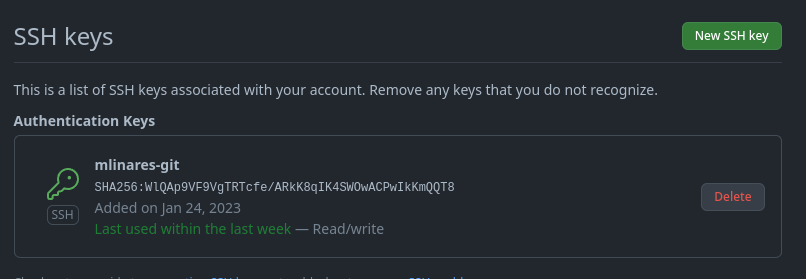
* Creamos un nuevo repositorio en GitHub para la realización de la práctica.

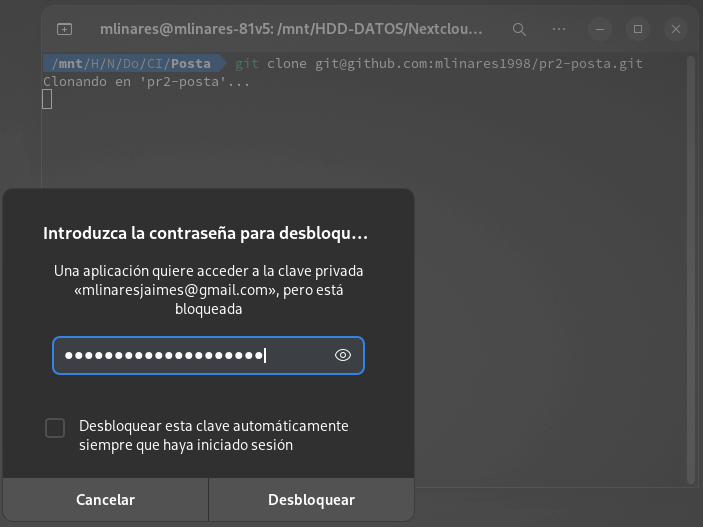


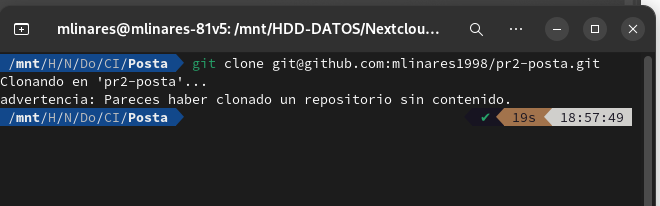
* Clonamos el repositorio en local usando git clone

**git clone** [**git@github.com:mlinares1998/pr2-posta.git**](mailto:git@github.com:mlinares1998/pr2-posta.git)

**NOTA: Estoy usando una clave SSH asociada a mi cuenta GitHub como mecanismo de autenticación con GitHub.**

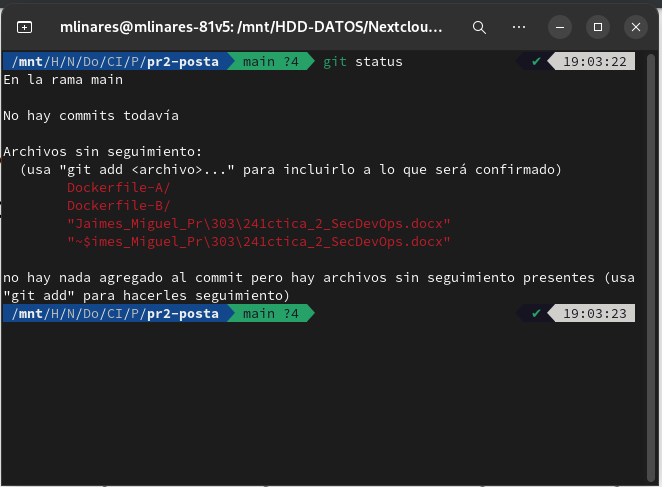
****

****

****

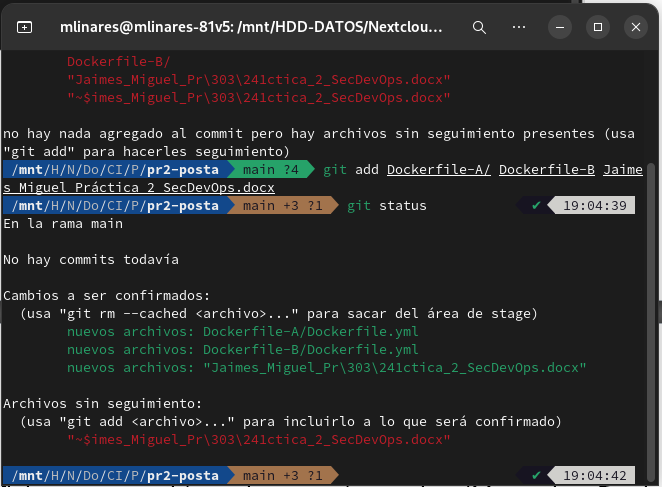
Nos ha clonado el repositorio vacío, listo para poder agregar nuestros archivos.

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

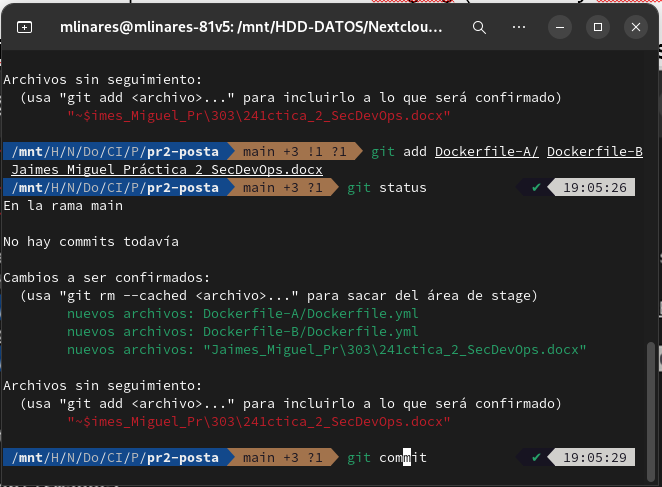


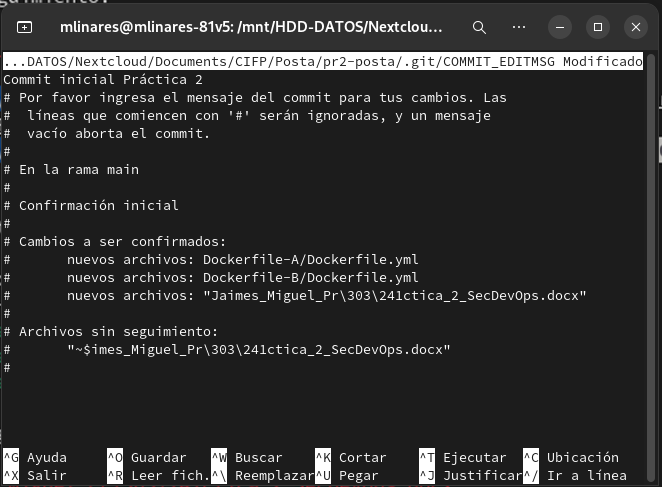
* Agregamos los ficheros requeridos a la zona de staging (Memoria y Dockerfile vacios)

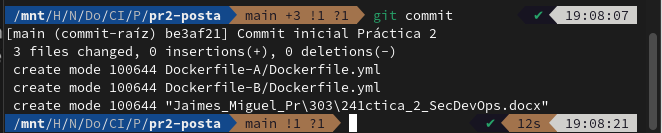
**git add Dockerfile-A/ Dockerfile-B Jaimes\_Miguel\_Práctica\_2\_SecDevOps.docx**



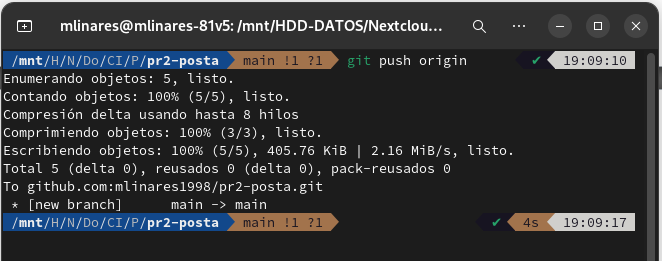
* Hacemos commit de nuestros cambios con **git commit**

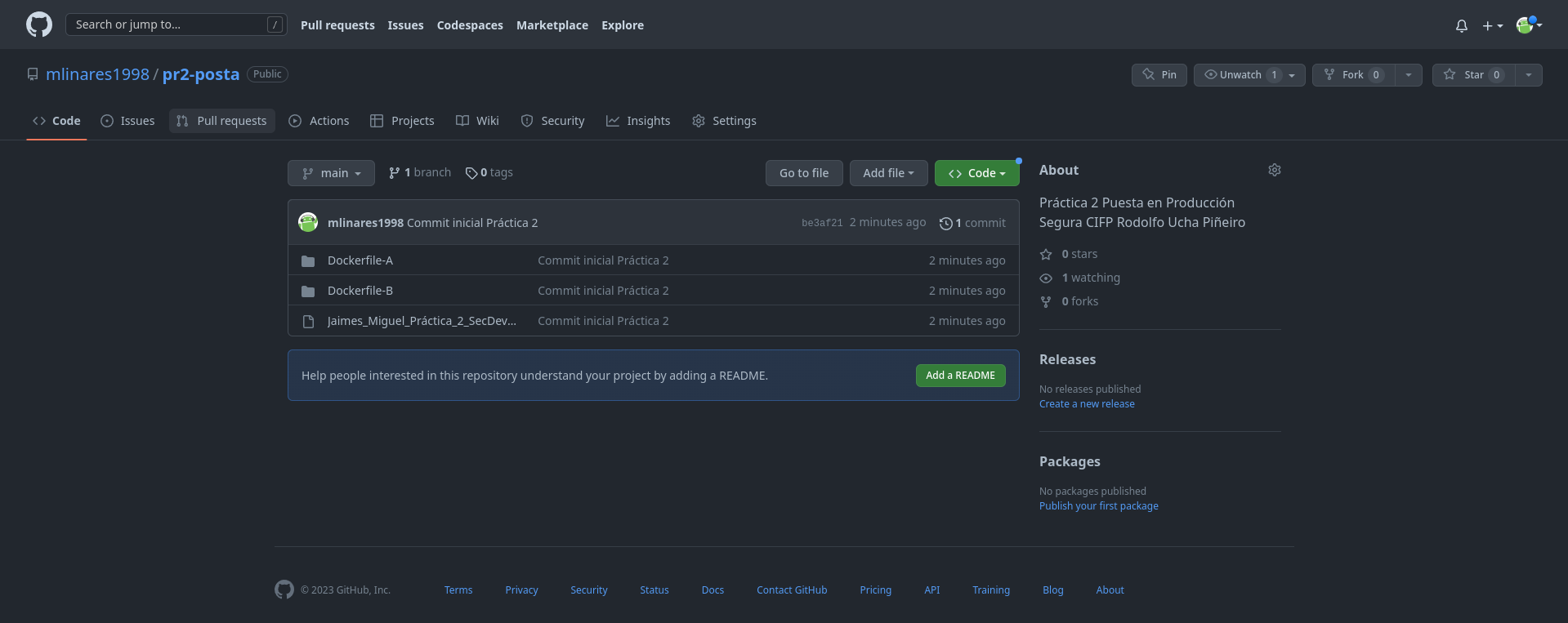






* Usamos **git push** para subir nuestros cambios al repo.





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

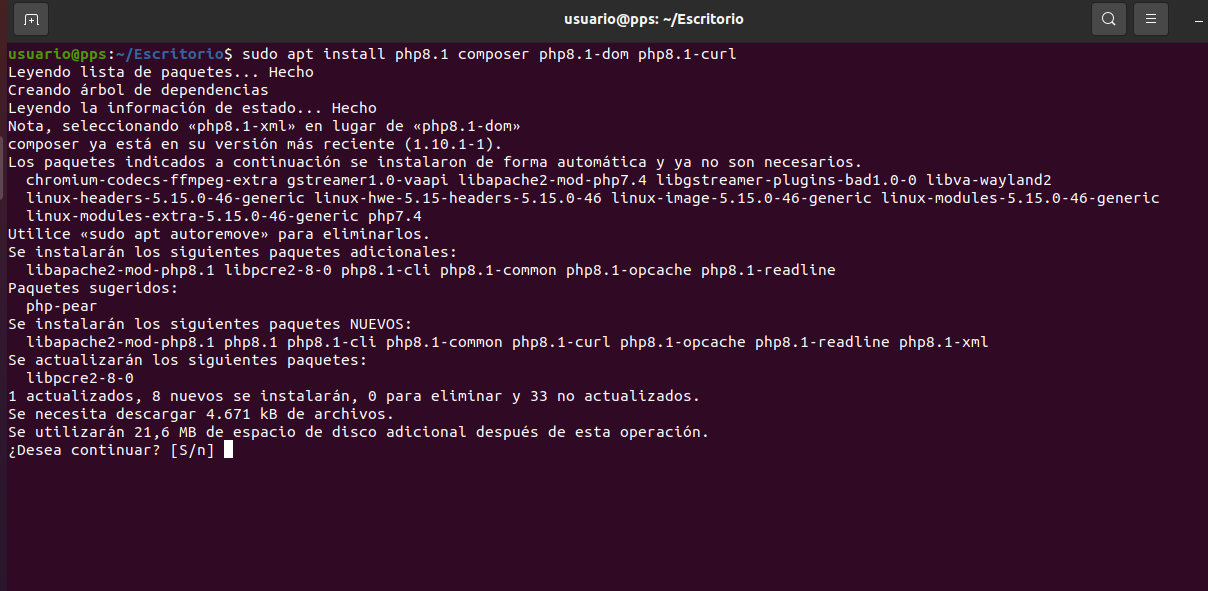
# Crear Dockerfile copiando la aplicación directamente desde el anfitrión

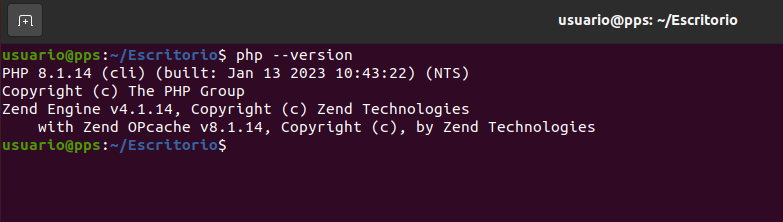
* Crea un Dockerfile que partiendo de una imagen PHP genera una imagen que copia una aplicación en PHP a un directorio del contenedor. Esta aplicación se debe copiar directamente desde un directorio del anfitrión.

Usaremos la siguiente aplicación demo:

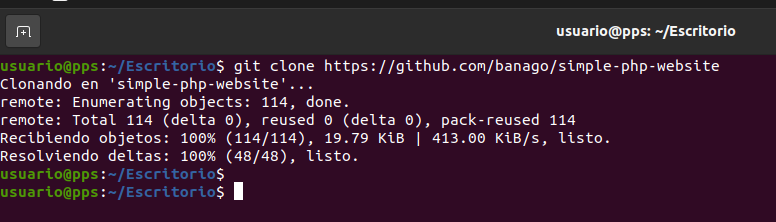
<https://github.com/banago/simple-php-website>

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



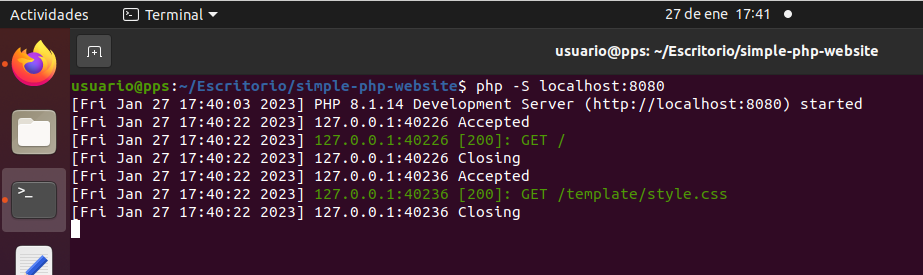


* Clonamos el repositorio



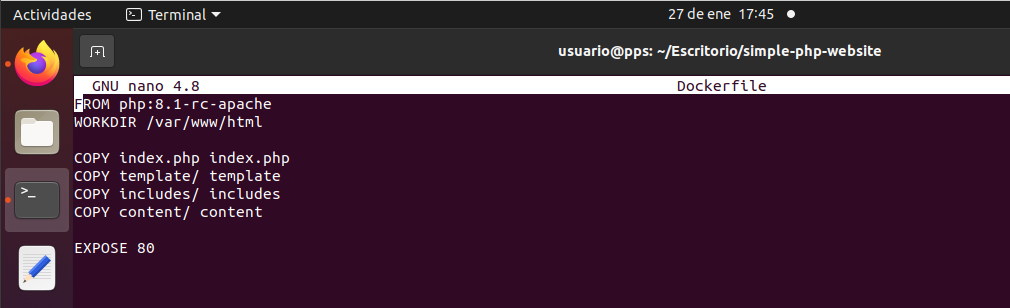
* Lanzamos la aplicación con el servidor de desarrollo de PHP para comprobar que funciona.

**php –S localhost:8080**





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

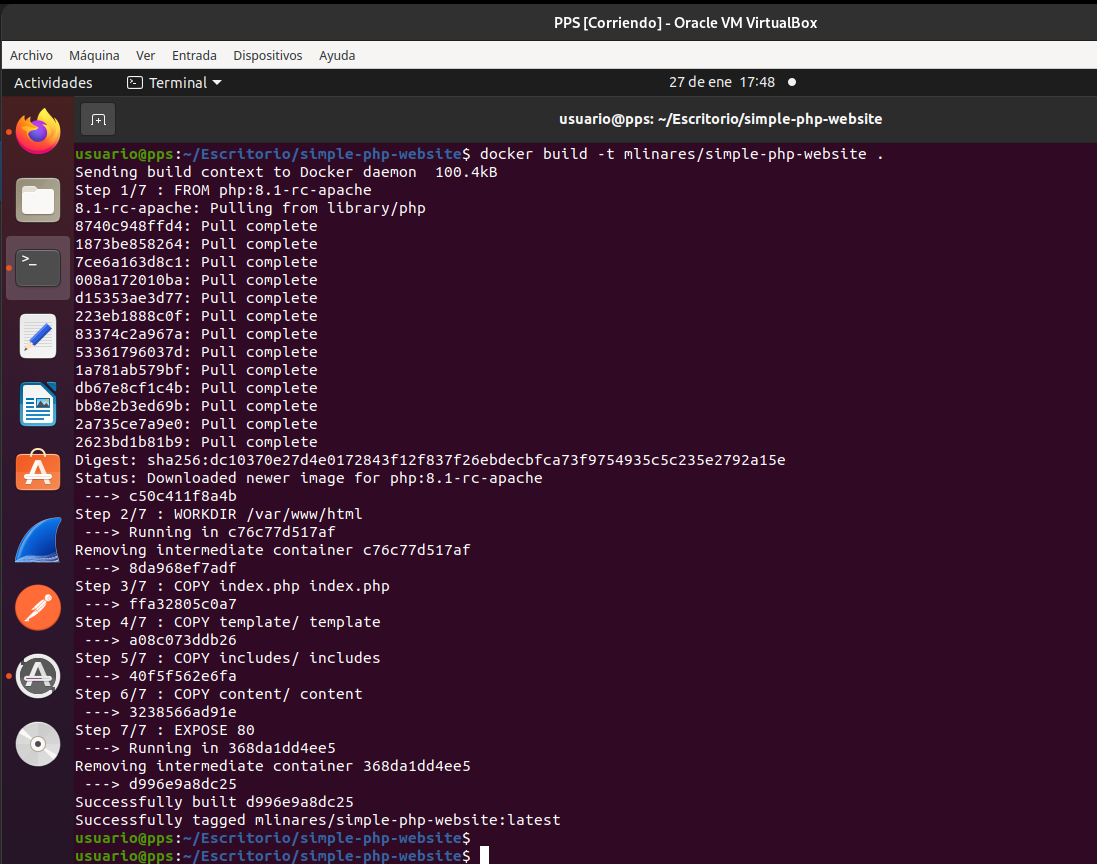


Este Dockerfile usara la imagen de PHP 8.1 con el servidor Apache pre-configurado, copiara el fichero index.php y carpetas requeridas a la carpeta /var/www/html del contenedor. Finalmente expone el servicio desde el puerto 80.

* Construimos la imagen.

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

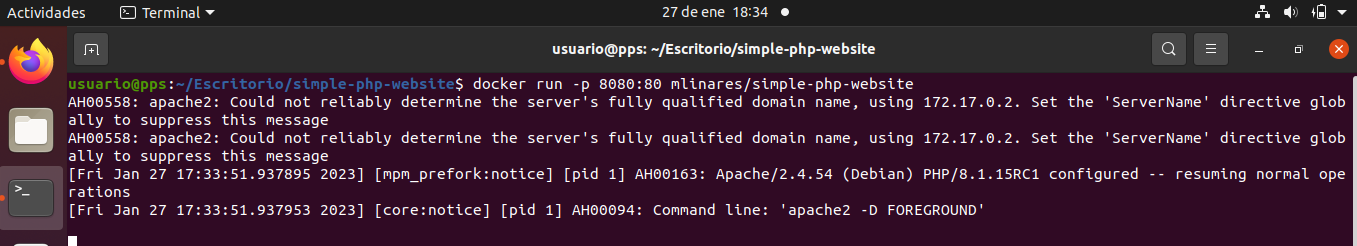
Nos construirá una imagen con el nombre especificado usando el Dockerfile de la carpeta en la que estamos posicionados.

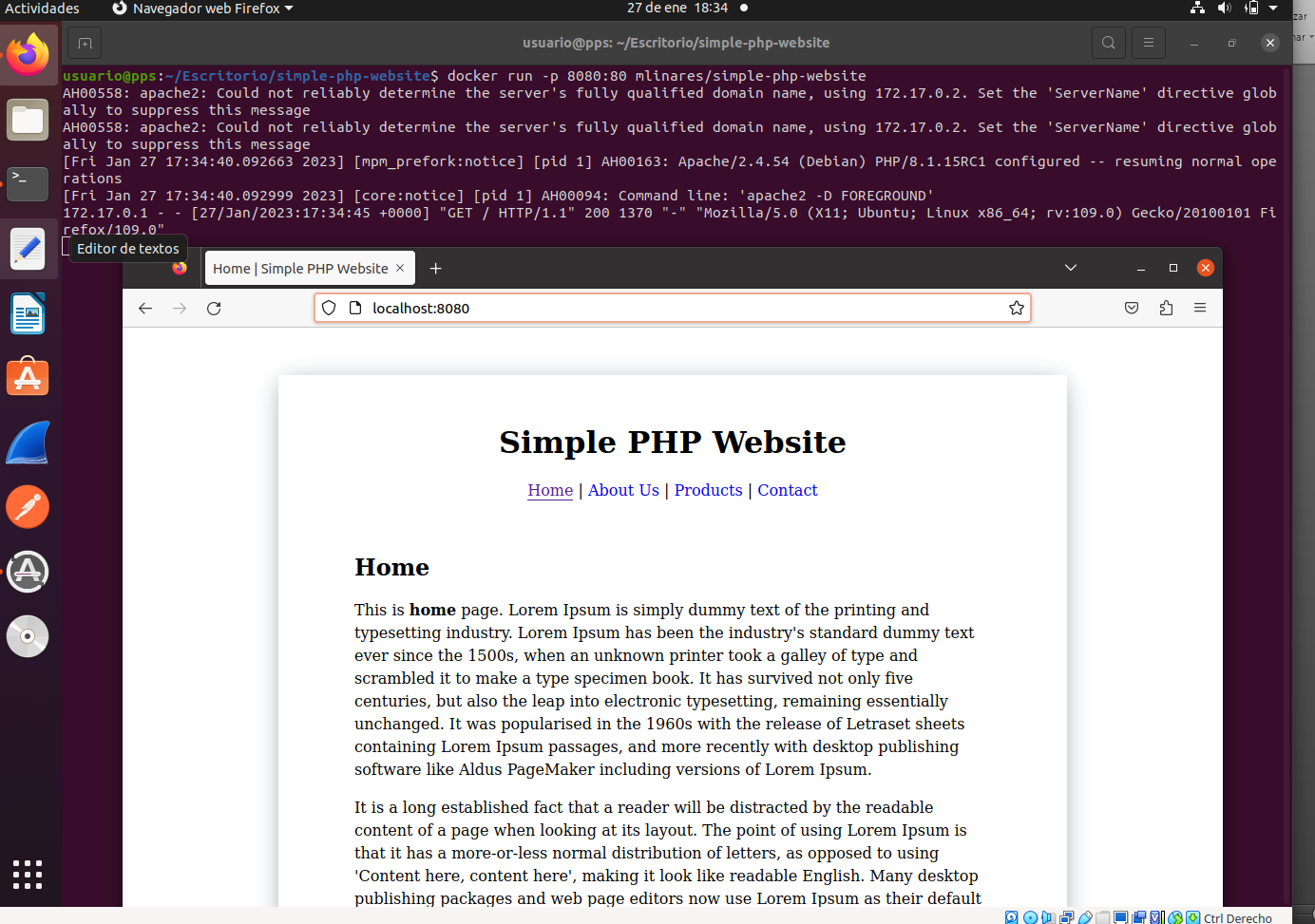


* Lanzamos la imagen y comprobamos que funciona correctamente.

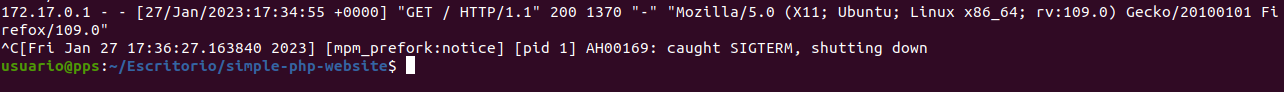
**docker run –p 8080:80 mlinares/simple-php-website**

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**

