Índice

[**Git donde se gardará o progreso**](#_8zjtsd88iztk) **3**

[Repositorio en GitHub e carpeta local](#_wcqybcgefltg) 3

[Primeiro PUSH ao GitHub](#_vuqt4sbcrguk) 4

[**Docker**](#_eq38eyek7020) **6**

[Instalar Docker](#_eevxxpm2eem) 6

[Exercicio 1](#_wuftbgk4qg4u) 8

[Exercicio 2](#_blrgf043f98z) 9

[Exercicio 3](#_on8t8ydh2j0v) 10

[Contenedor 1](#_4qogiqvtisro) 10

[Contenedor 2](#_hbkq2sd95gen) 11

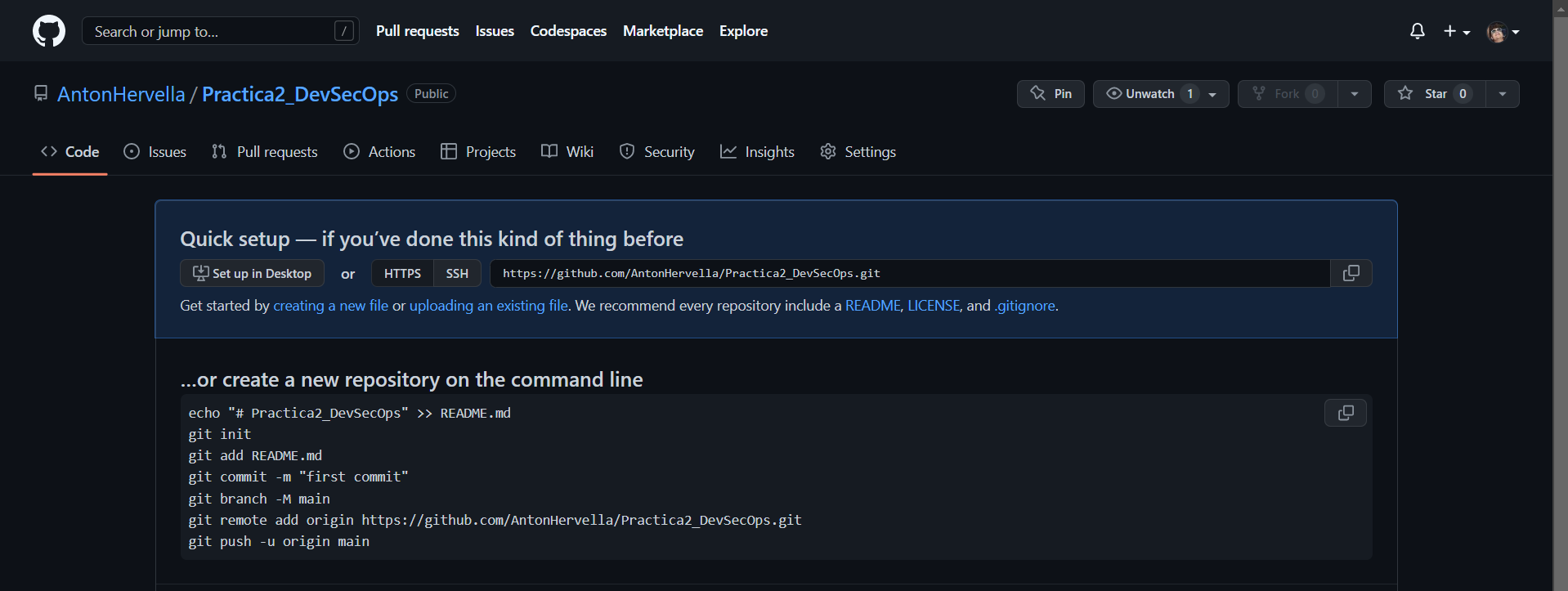
# Git donde se gardará o progreso

## Repositorio en GitHub e carpeta local

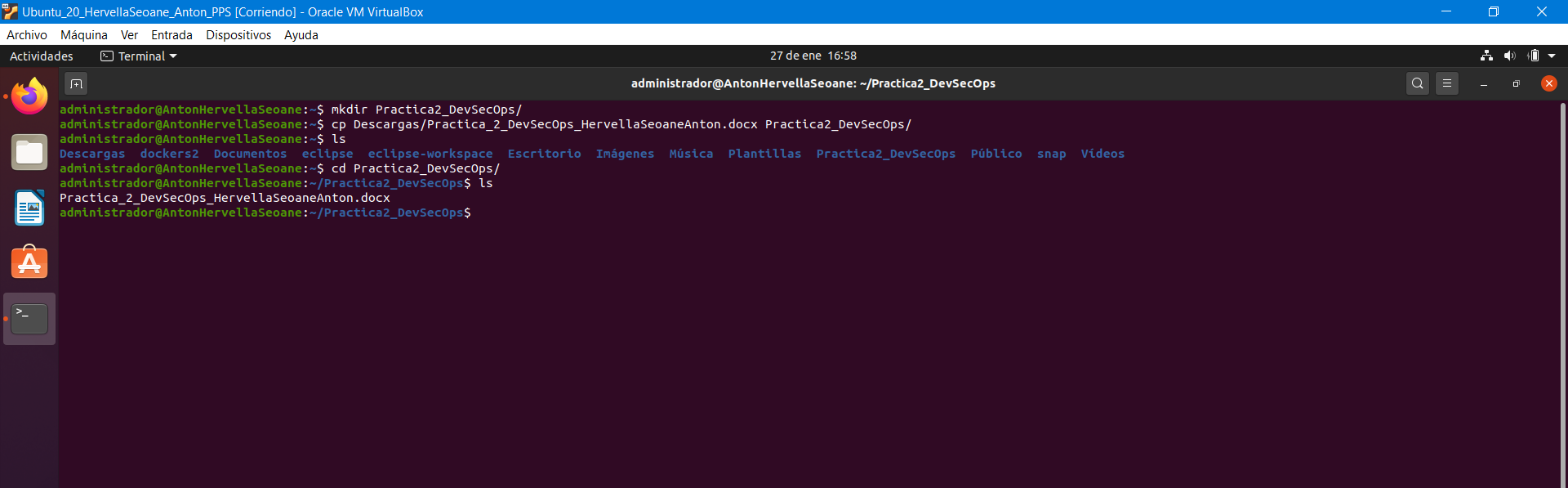
O repositorio Git donde gardaremos todos os avances que faremos nesta práctica será:

<https://github.com/AntonHervella/Practica2_DevSecOps.git>

Que acabamos de crear vacío en GitHub:



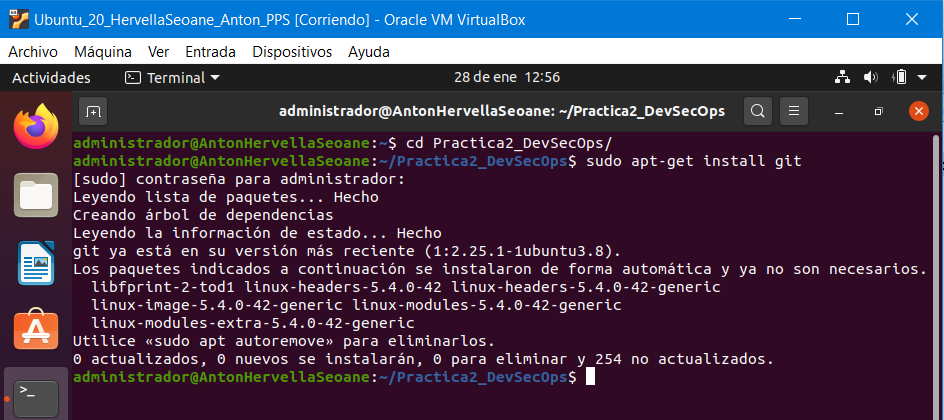
Aquí deixaremos todos os procesos que terá a practica actual. Tamen crearemos unha carpeta no noso equipo para gardar todos os avances da práctia, donde tamén se incluirá esta propio documento.



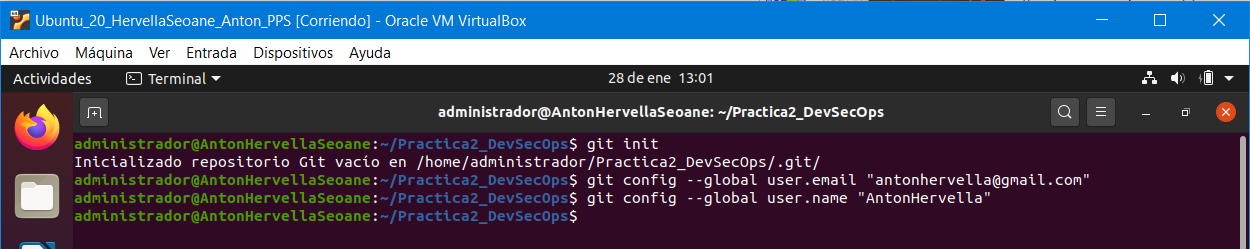
Cada documento que se añada a esta carpeta será subido ao repositorio de GitHub anteriormente creado. Tamen iremos subindo cambios que se fagan nos mesmos arquivos.

## Primeiro PUSH ao GitHub

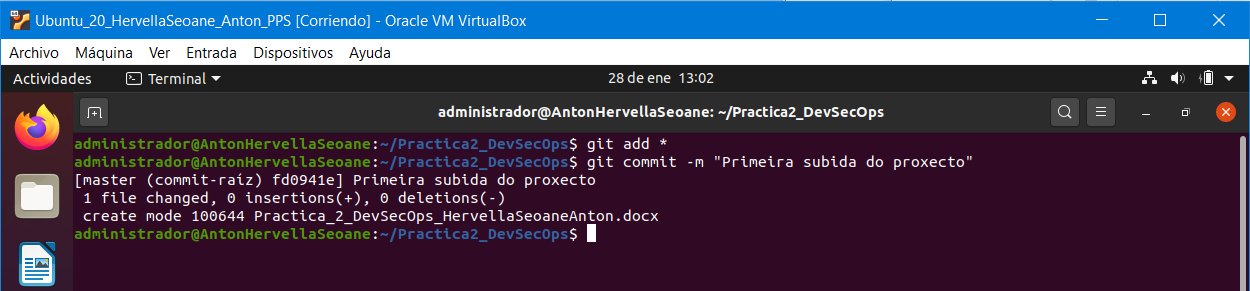
Para subir o arquivo que estamos creando actualmente dberemos aprender o básico de git. Para descargar git usaremos:



Neste caso xao temos instalado, o seguinte será inicializar a carpeta da práctica e inserir os nosos datos como o noso email e o noso nome:

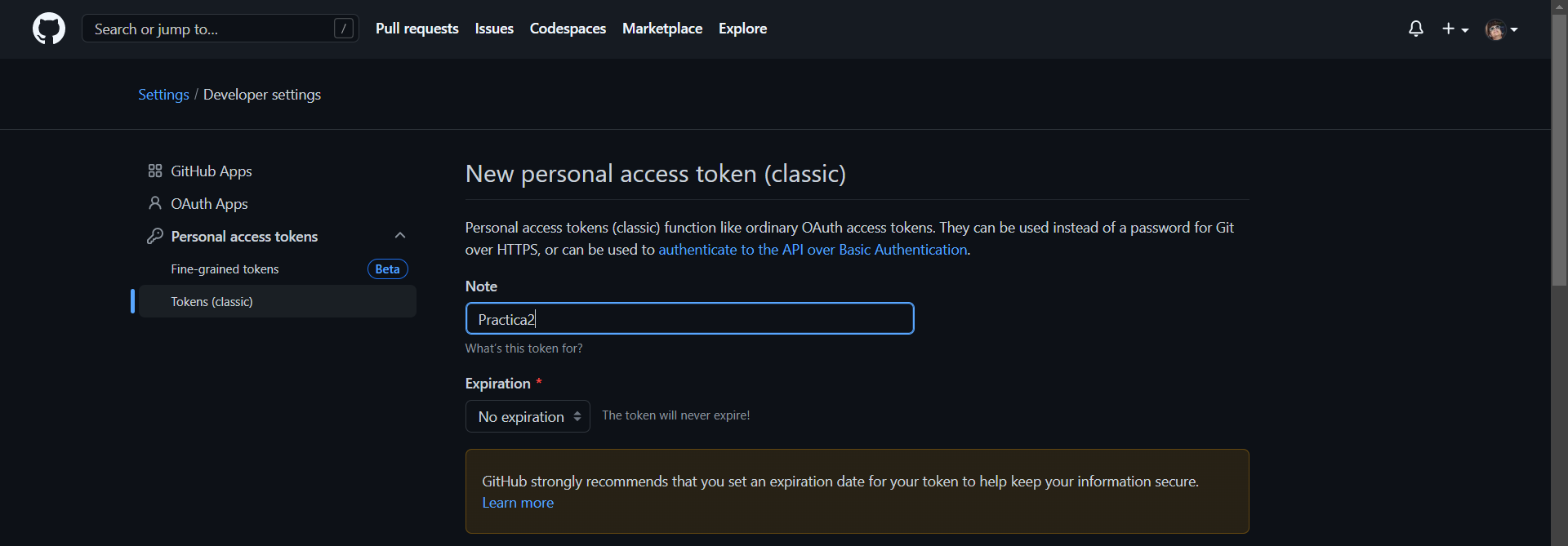


Agora tocará añadir o arquivo e facer un commit

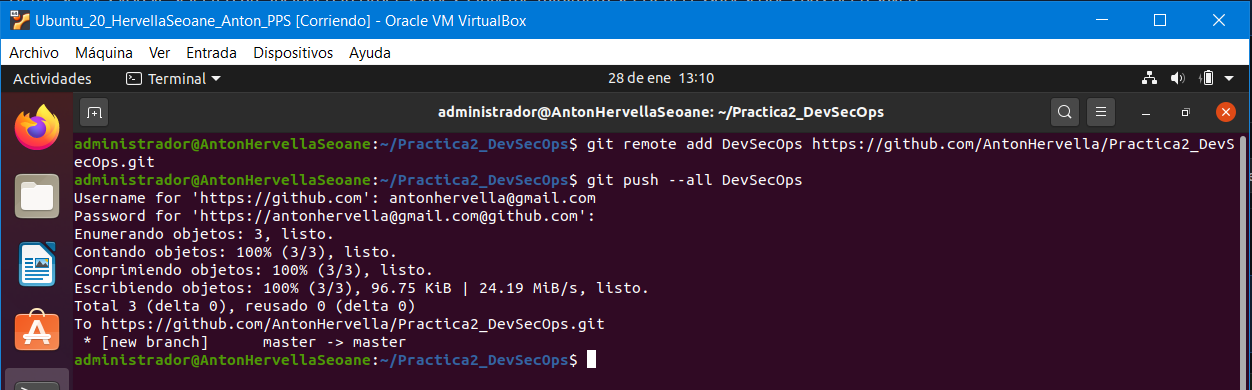


Con esto feito, tocará subir o proxecto gon un git push, donde deberemos inserir o noso usuario e o noso token privado de github.

Se aínda non tes token privado debes obtelo nas opcións de desarrollador no teu perfil de GitHub:

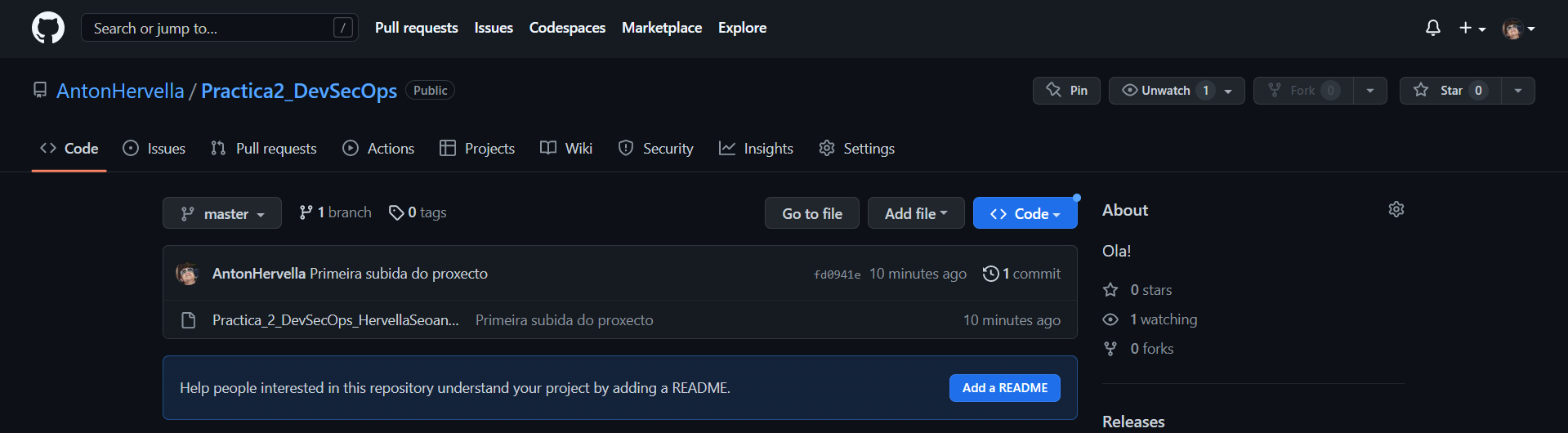


Co token creado, deberemos indicar a que repositorio o queremos subir e despois poderemos gardar os avances da nosa práctica en GitHub:



Donde nos dí “password” é donde debemos inserir ou pegar o token que creamos anteriormente.

Agora comprobamos no repositorio se o noso arquivo está subido:



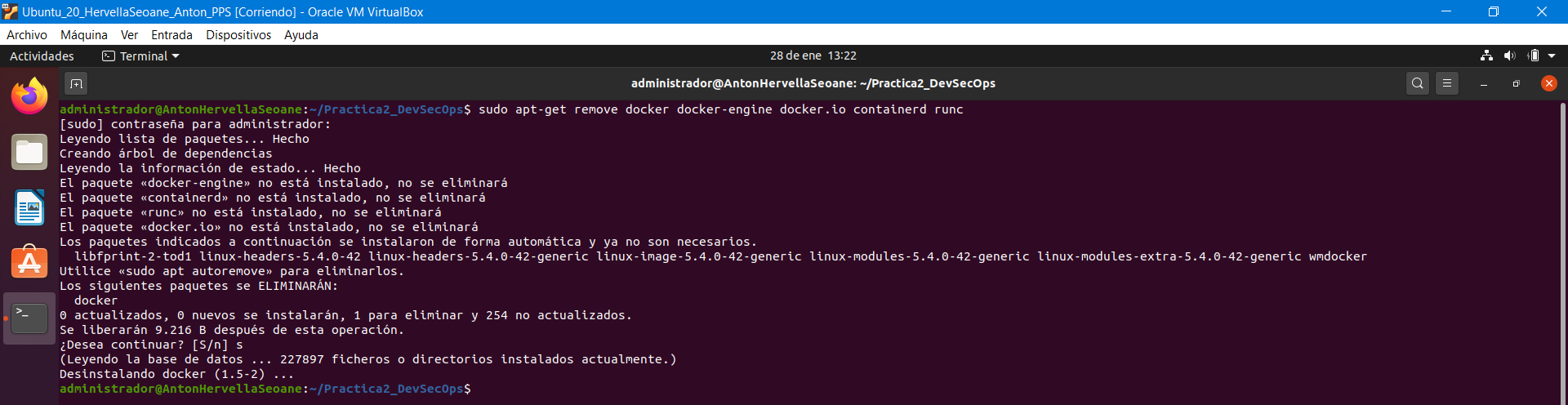
Agora podemos continuar co proxecto, subindo os arquivos, carpetas e resto de elementos.

# Docker

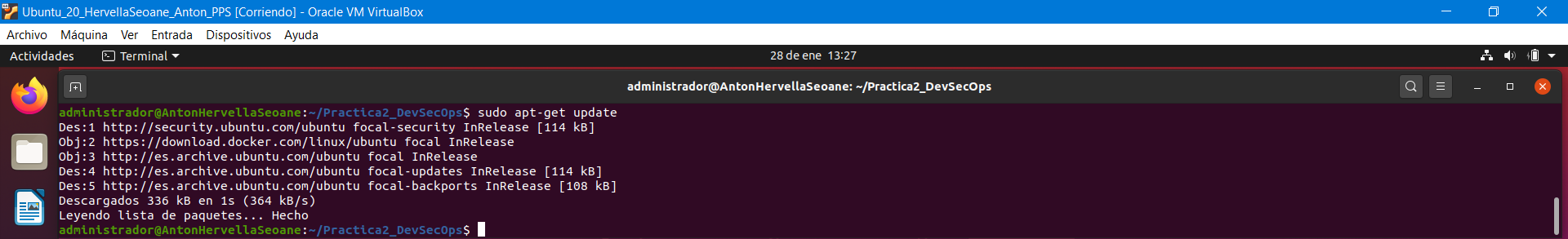
## Instalar Docker

O primeiro que debemos facer cando queremos comezar a usar docker é instalar o programa pertencente. Neste caso faremos unha instalación en un ubuntu.

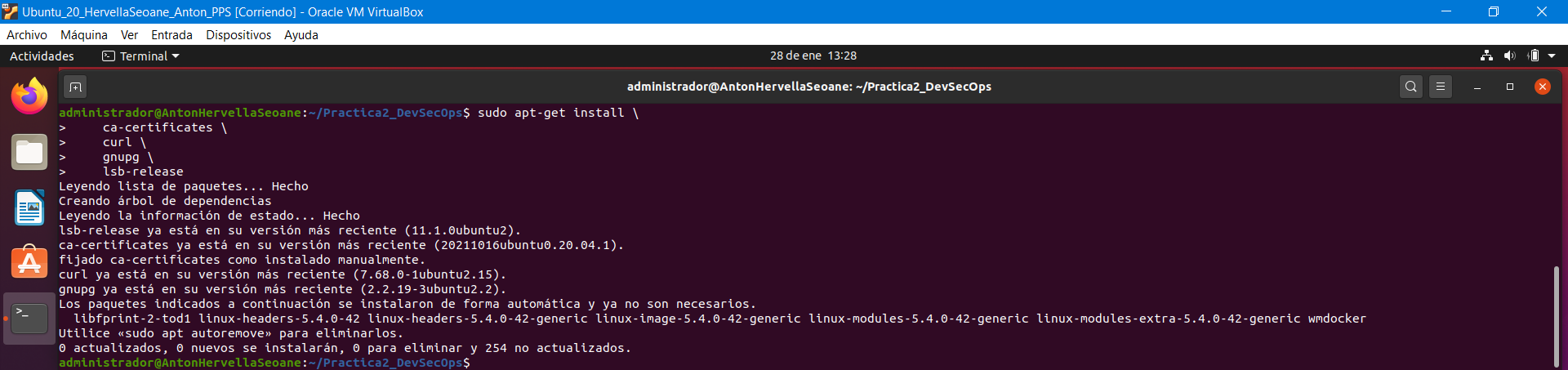
O primeiro a facer é eliminar posibles instalacións antigas que teña o noso equipo:



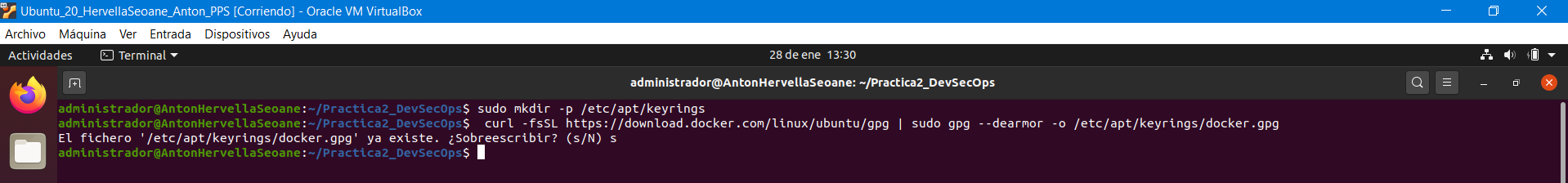
Actualizaremos para que non teñamos erros na instalación:



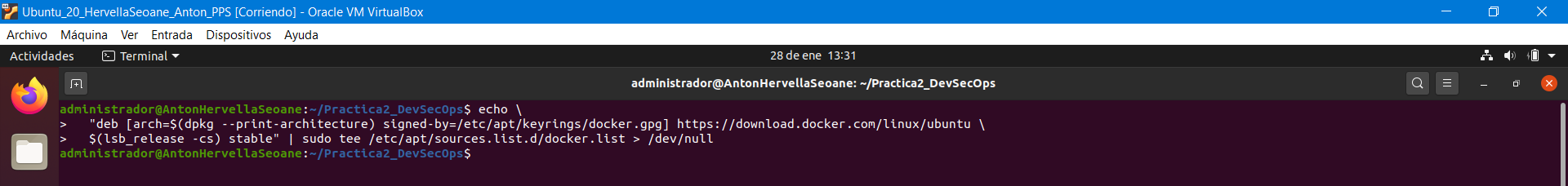
Instalaremos o seguinte tamén:



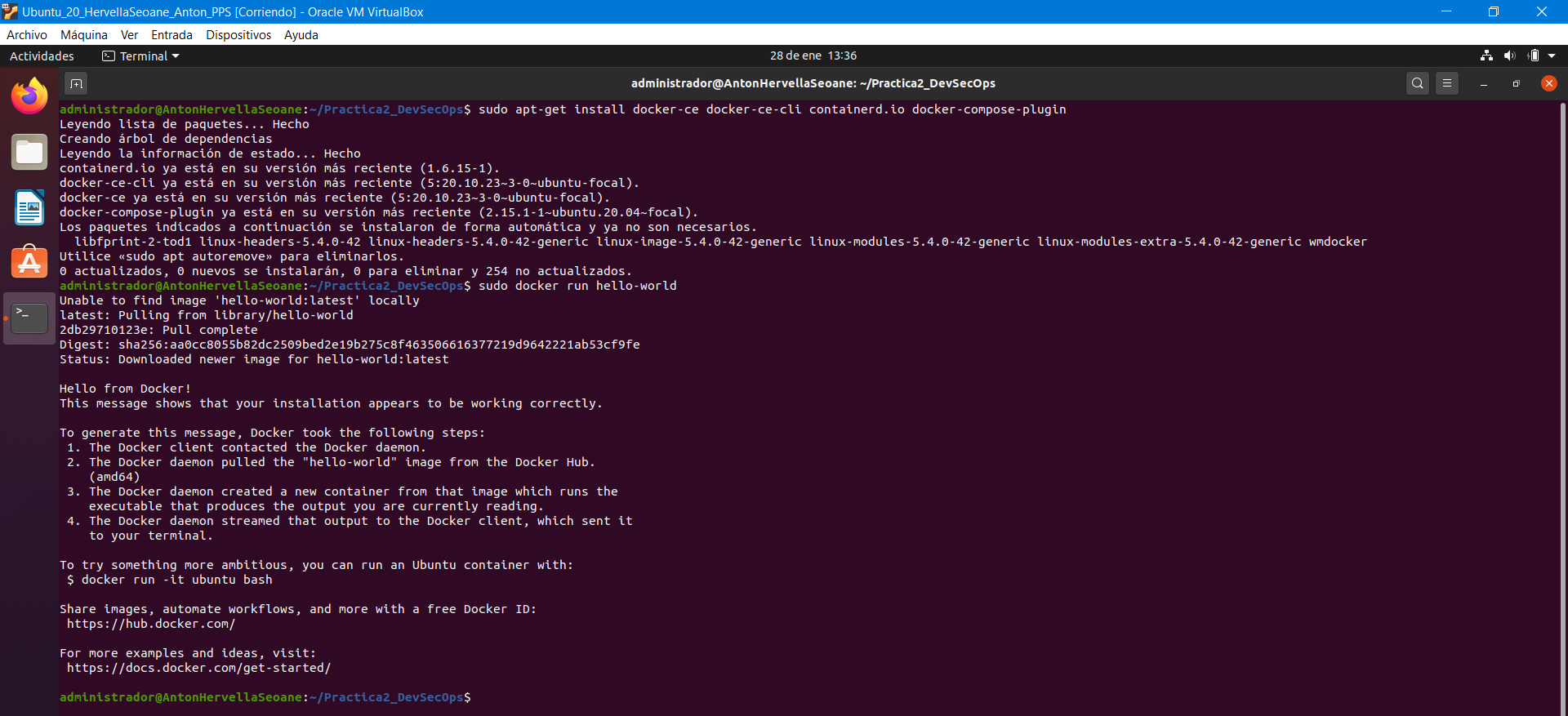
E crearemos :



Agora facemos un set-up do repositorio:



O seguinte que faremos será instalar Docker:

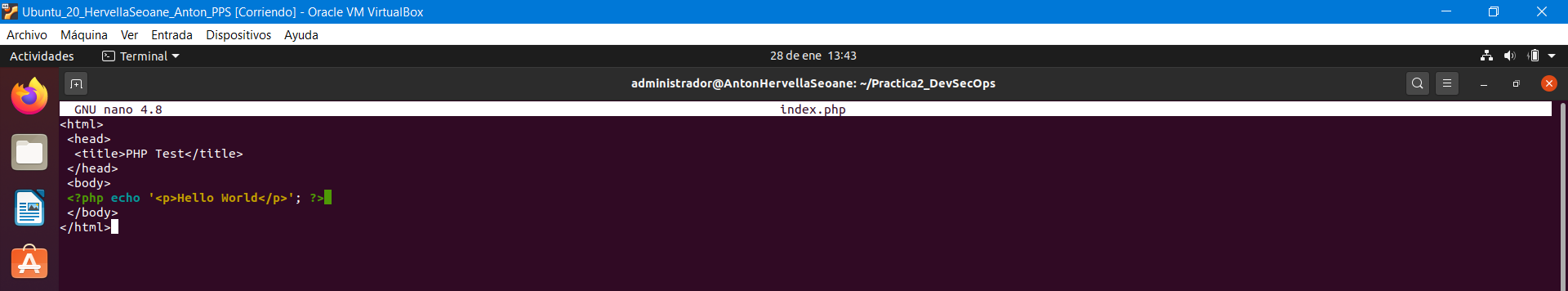


O noso equipo xa tiña o docker na sua versión mais recente, ainda así para comprobar, no segundo comando executamos o “hello world” de Docker, que se usa para comprobar se a instalación do Docker está correcta.

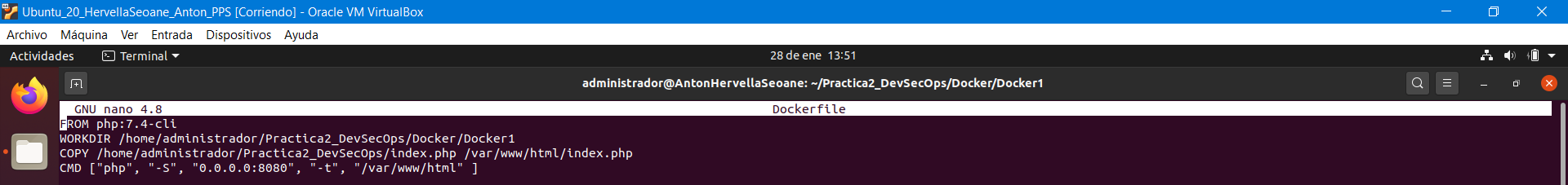
## Exercicio 1

Deberemos crear un dockerfile que vai partir de unha imaxe PHP, copiando unha aplicación PHP do noso equípo local. Neste caso non empregaremos un PHP con base de datos xa que deberíamos instalar o MySQL.

O primeiro que debemos facer é ou ben crear ou ben descargar unha aplicación PHP, neste caso crearei un arquivo que se vai chamar index.php.



Agora, crearemos o dockerfile que faga o anterior mencionado:

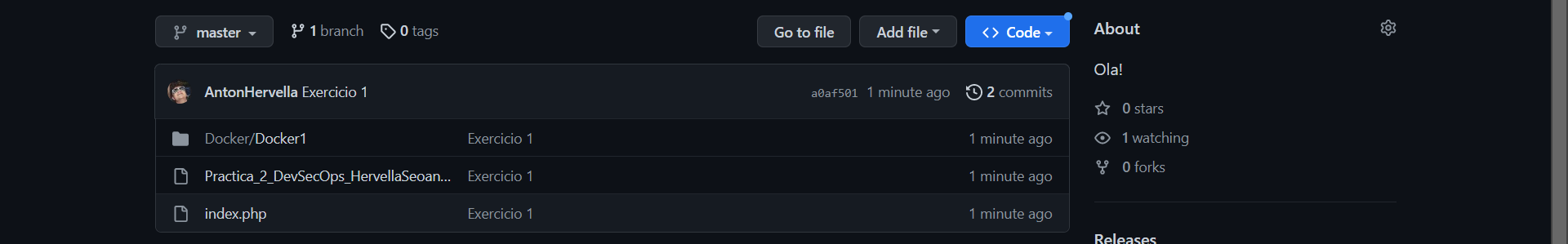
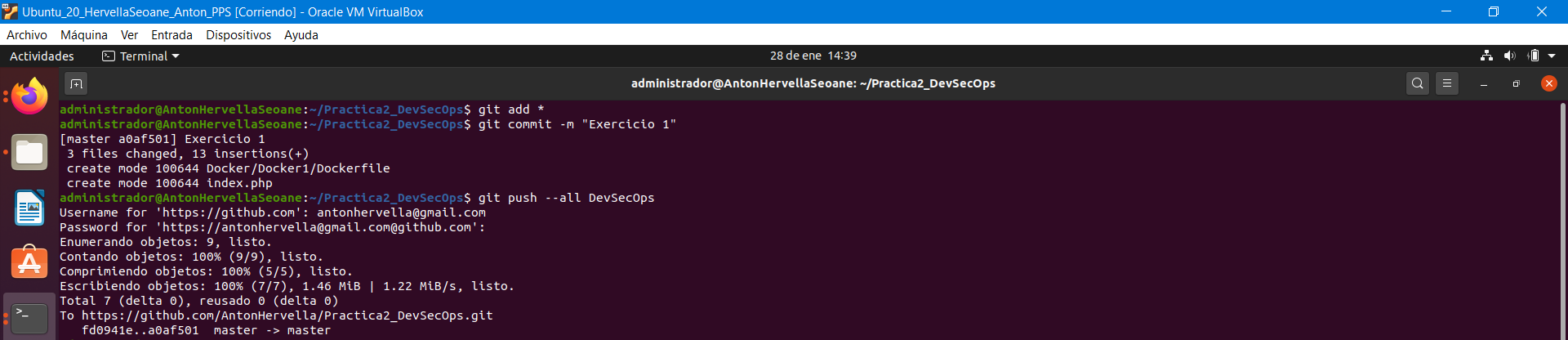


En “FROM” indicamos que imaxe temos que usar, neste caso suamos unha imaxe de PHP na versión 7.4

En “WORKDIR” seleccionamos cal é a carpeta desde donde traballaremos

“COPY” será a finalidade deste Dockerfile, que é copiar index.php da carpeta “/Practica2\_DevSecOps” ata a carpeta “/var/www/html/”.

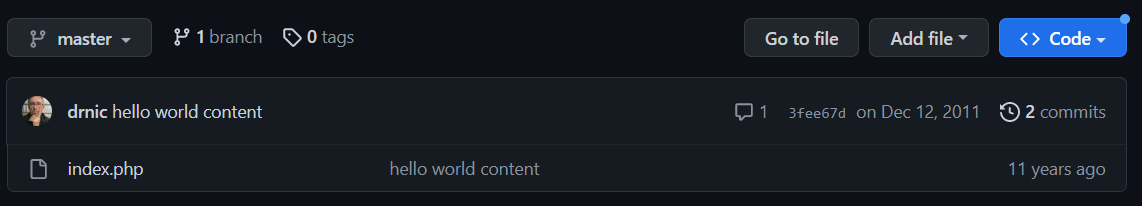
E subimolo a GitHub



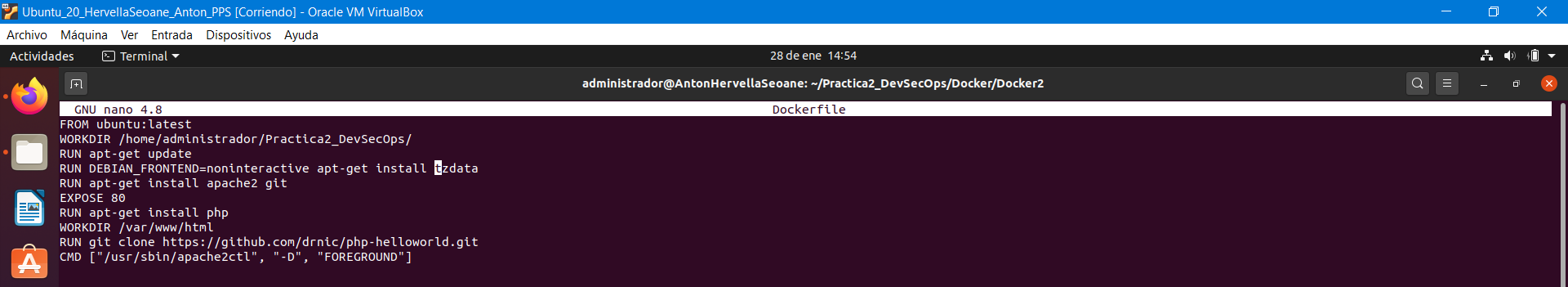
## Exercicio 2

Nesta tarefa pidennos instlar tanto apache(expondo o porto 80) como php. Tamen teremos que descargar de algun repositorio unha aplicación php. Seguiremos usando algunha que non teña ningunha base de datos para evitar instalar MySQL.

A aplicación php a descargar será:



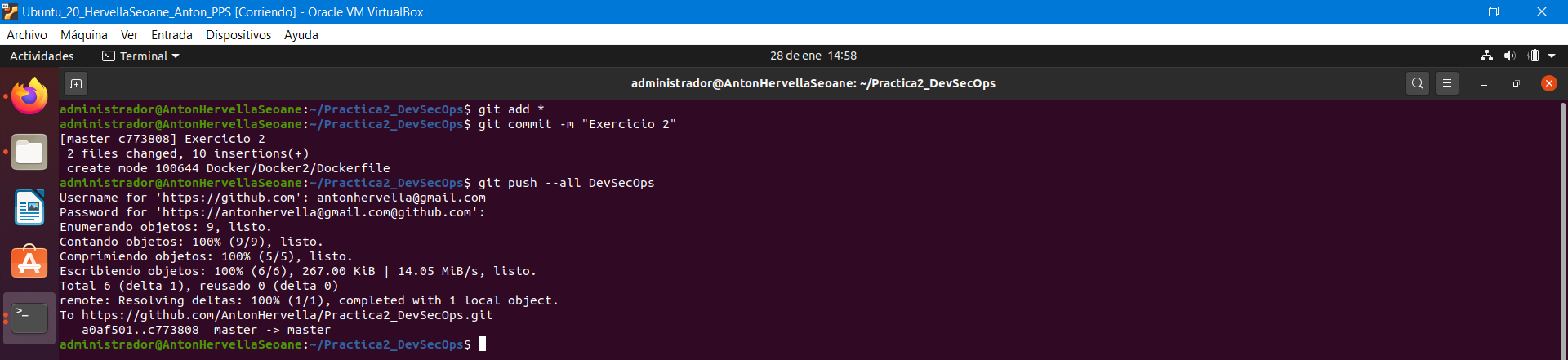
O dockerfile que creamos é:



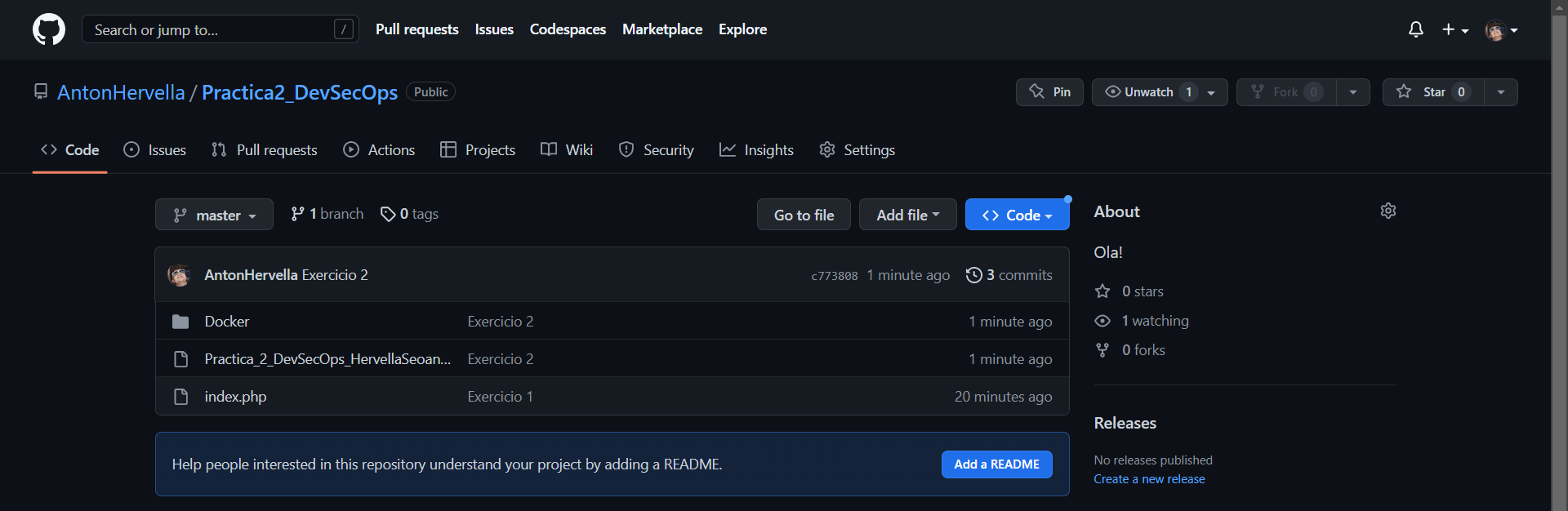
“RUN” usase para executar comandos, como a instalación de apache2 e php ou o “update”

“EXPOSE” para expor un porto, neste caso o 80

Agora subimos os cambios a GitHub:



E miramos en github se está actualizado:

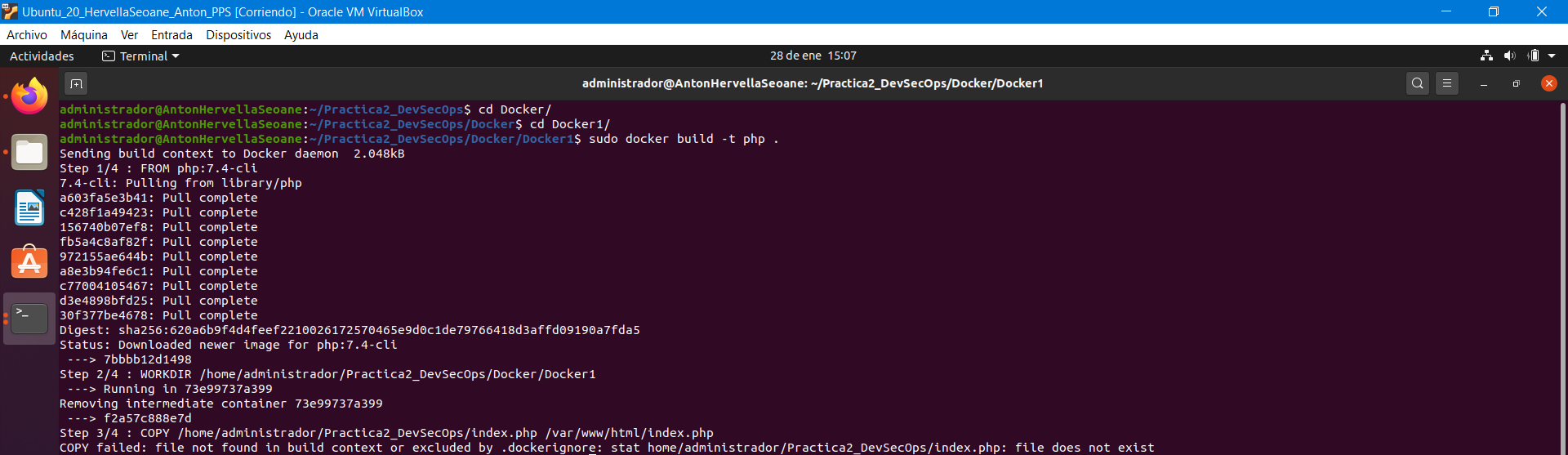


Agora poderemos seguir coa creación de contenedores.

## Exercicio 3

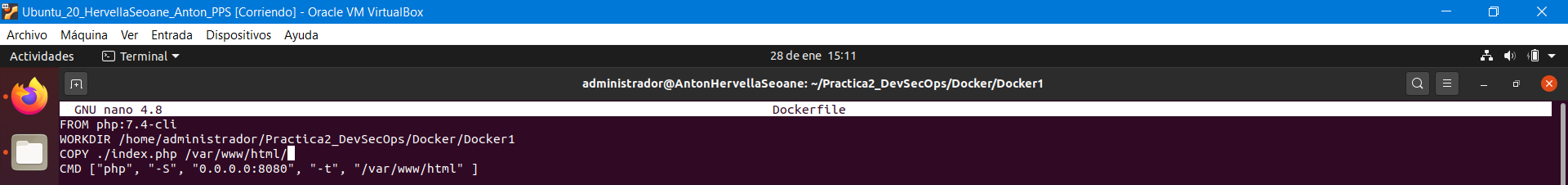
### Contenedor 1

Debemos dirixirnos á carpeta donde gardamos o Dockerfile e crear unha imaxe a partir da mesma:

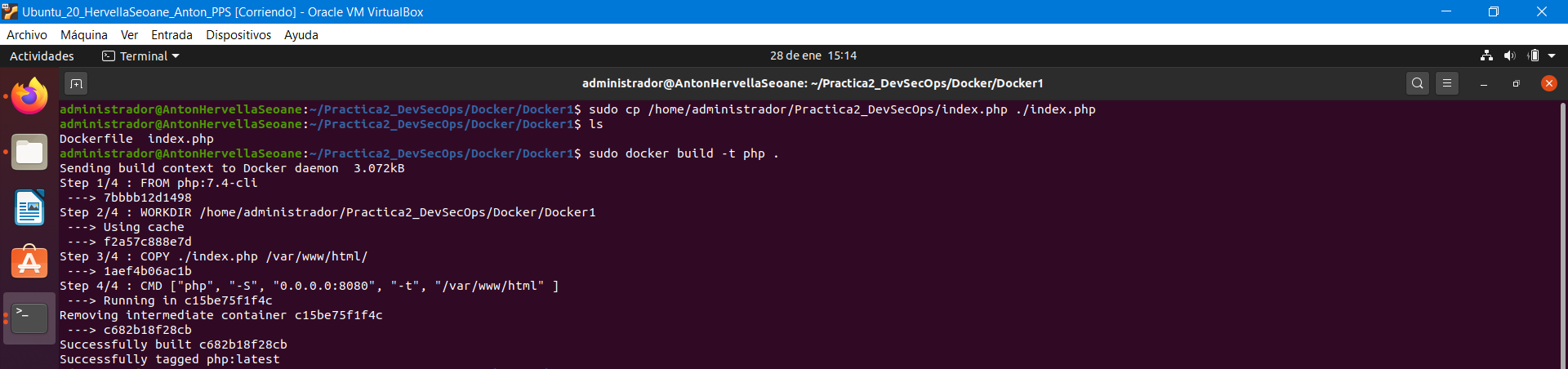


Algo sucedeu!

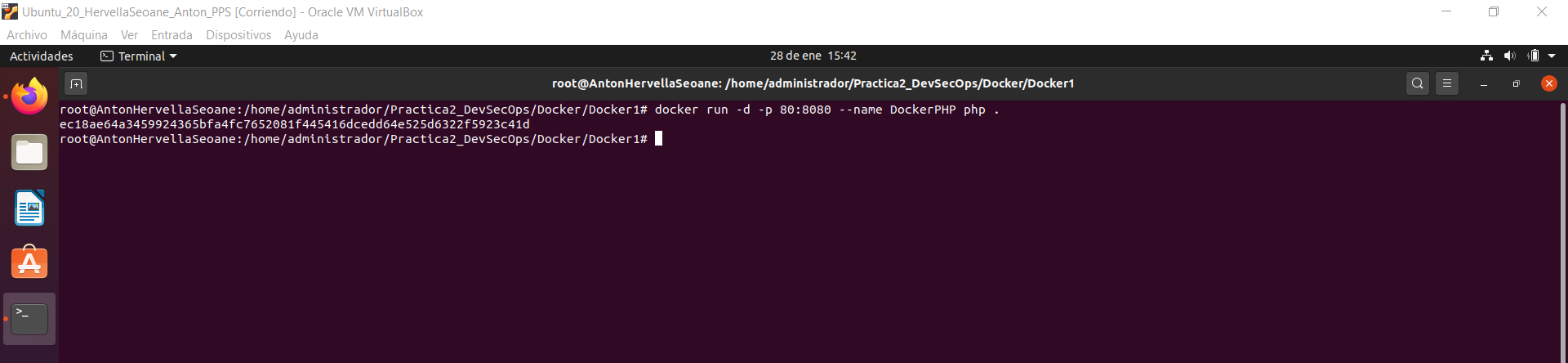
Imos cambiar o Dockerfile polo seguinte:



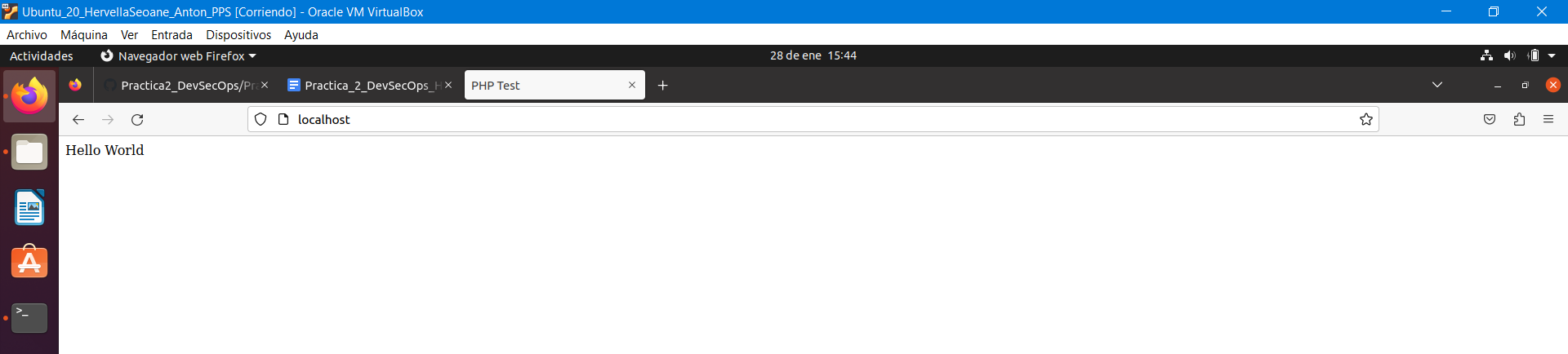
Reubicamos index.php na carpeta Docker1 e intentamos lanzar de novo a imaxe:



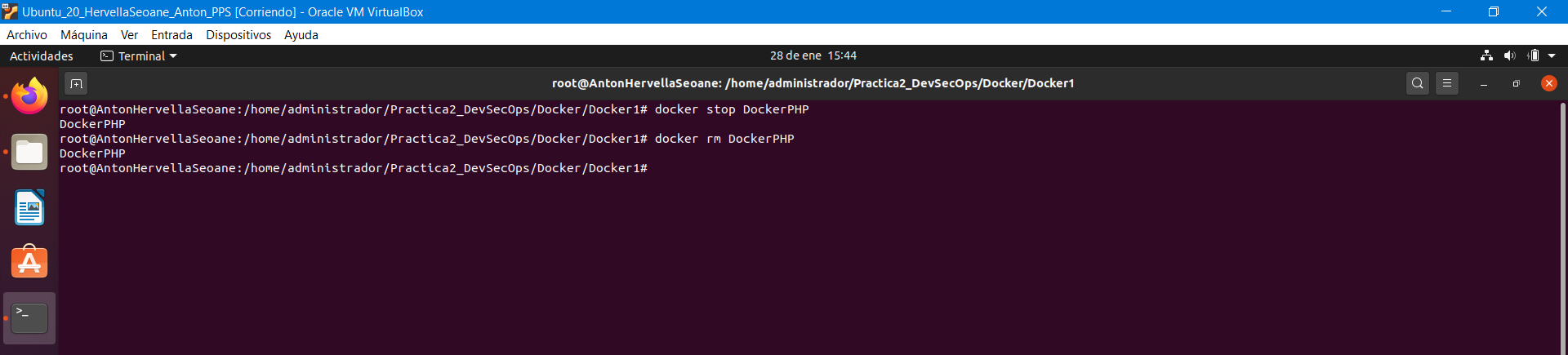
Unha vez aparezca que a imaxe foi creada sen erros, agora lanzaremos o contenedor:



(Tiven un erro, enton para crear, eliminar e parar os contenedores debía dacelo como super usuario, asi que antes de este comando empreguei o #sudo su para que todo puidese correr ben)

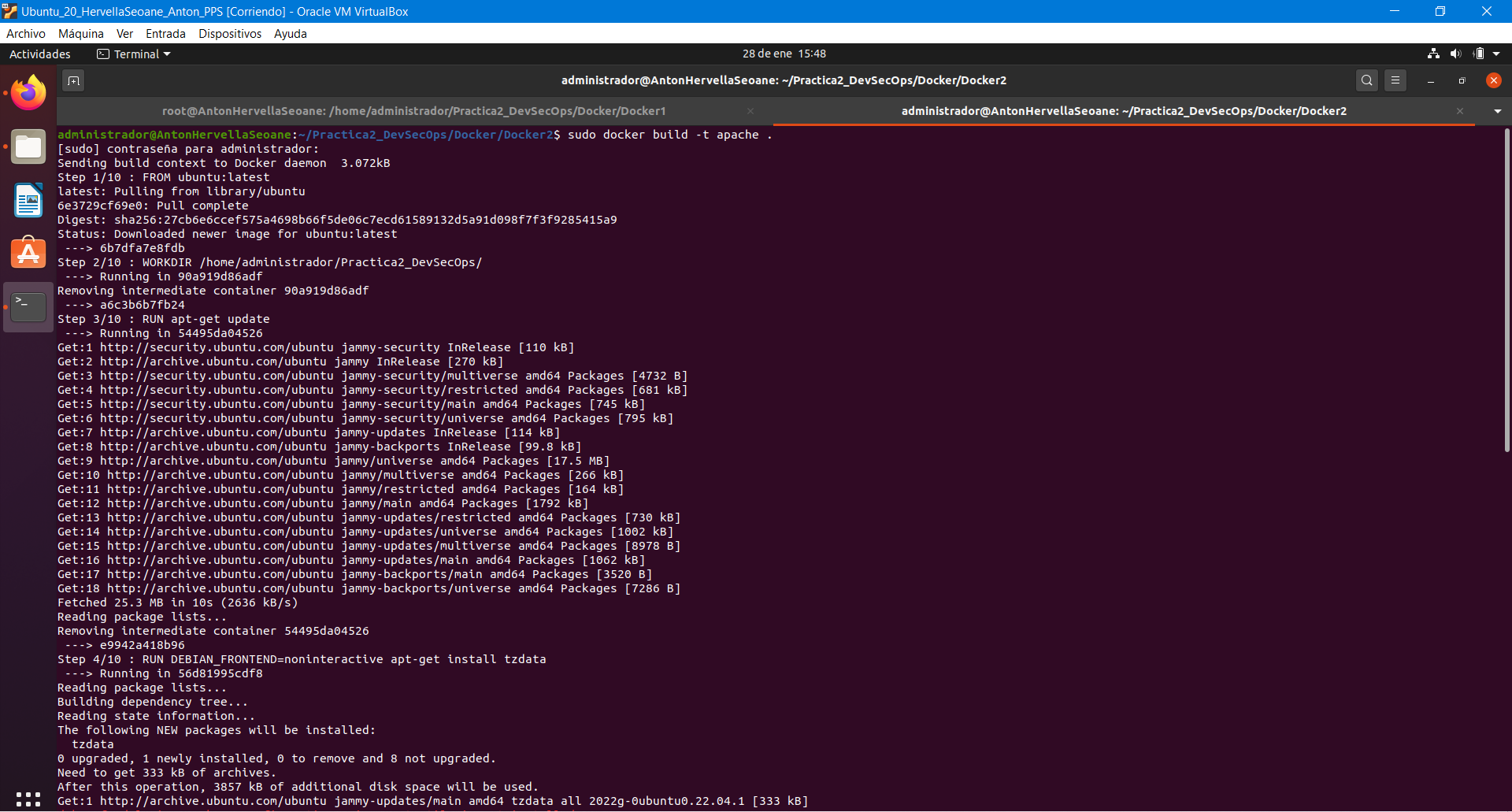


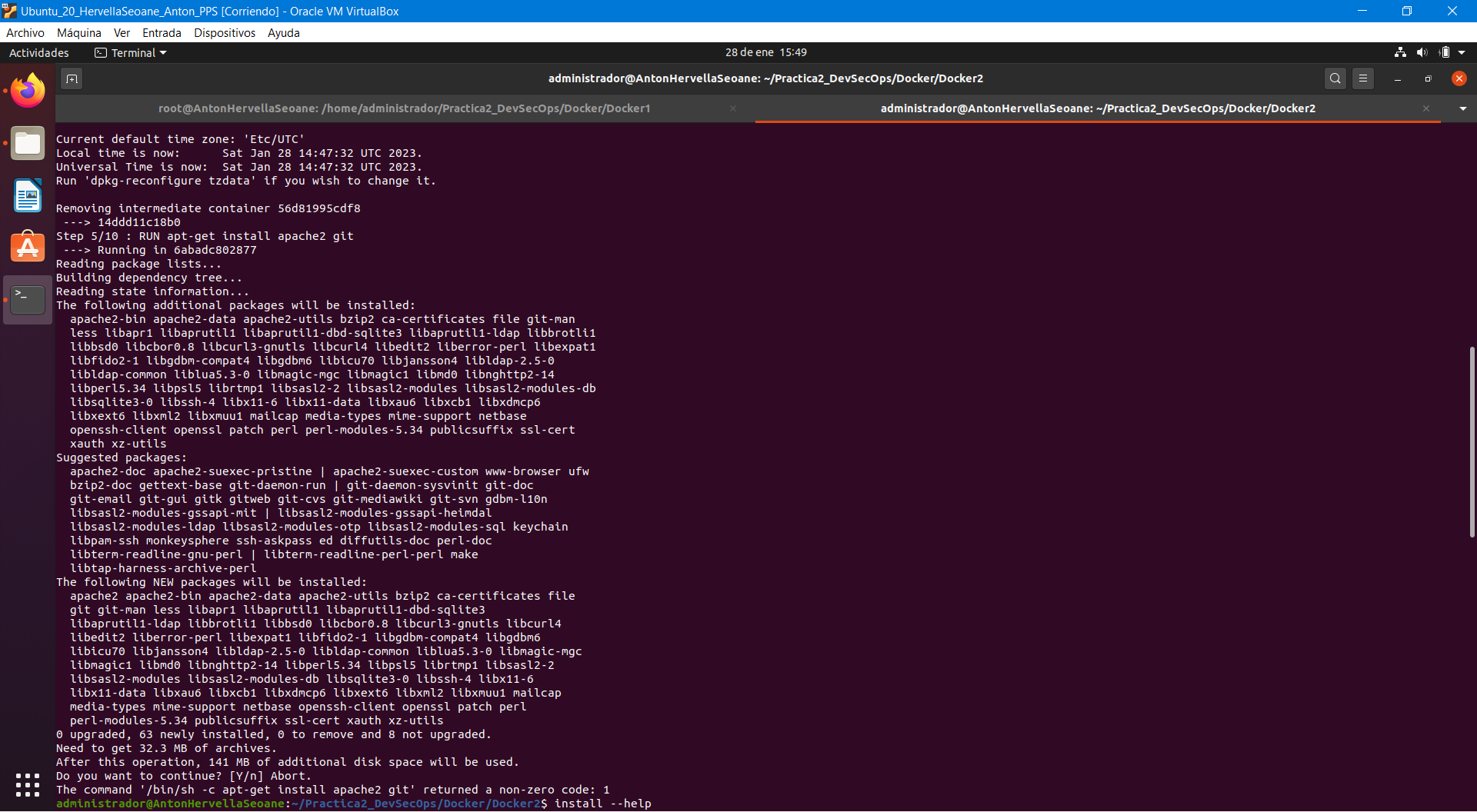
Agora pararemos e eliminaremos o contenedor:



### Contenedor 2

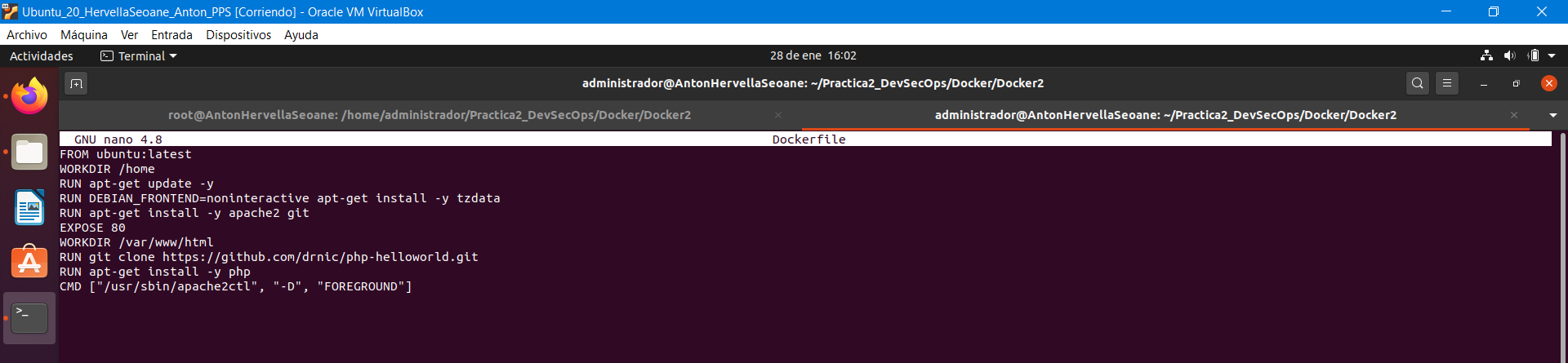
O mesmo faremos co contenedor 2:





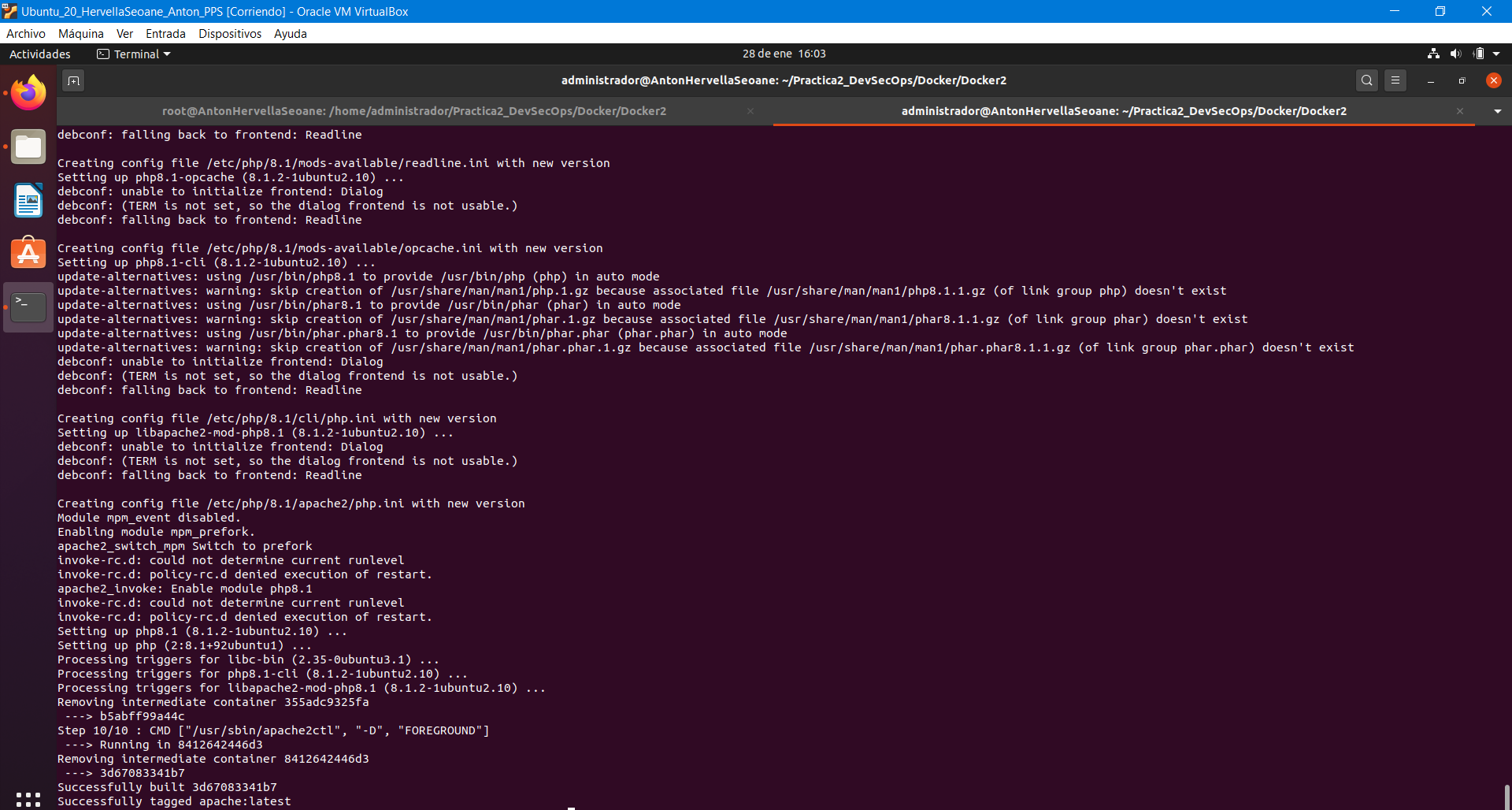
Tamen abortou?

Arreglase realizando un cambio no dockerfile:

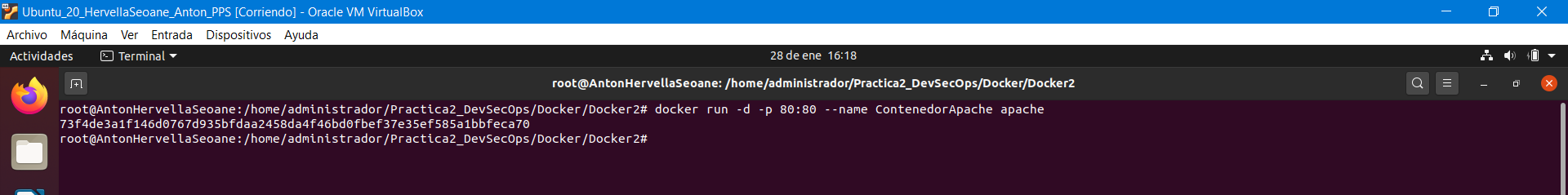


(Neste caso agregolle -y para que ao instalar sempre conteste “Yes” nas preguntas das instalacións)

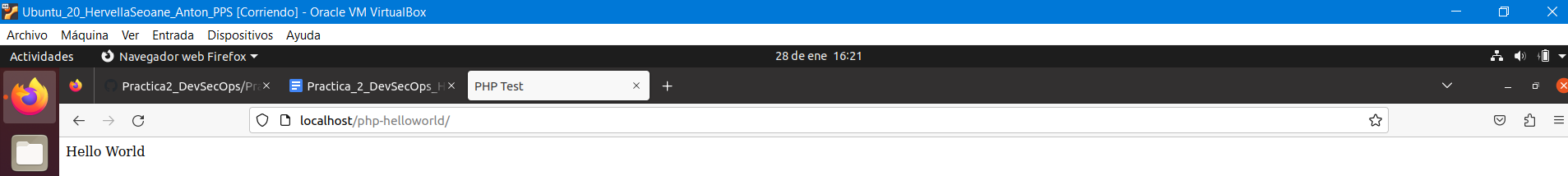
Probamos de novo:



Agora si, asi que procedemos co lanzamento do contenedor:

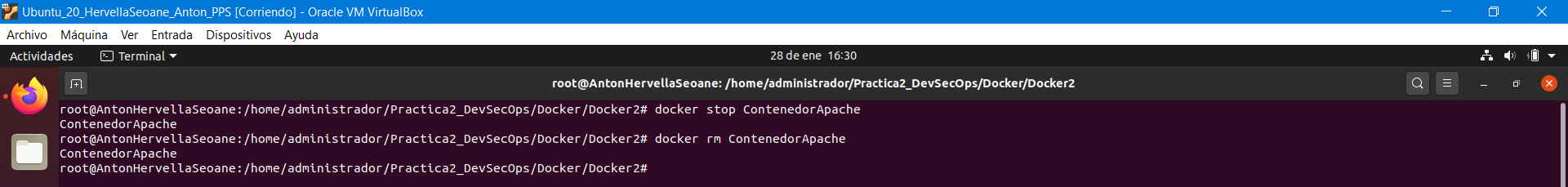


E miramos en localhost/php-helloworld que é donde descargamos a aplicación php



Aparece o hello world!

Agora toca parar e eliminar este:



E subir os cambios e avances a github