**Puesta en Producción Segura**

Práctica 2: DevSecOps

**Objetivo**

El objetivo de esta práctica es que el alumno comprenda los fundamentos de las DevOps seguras,también conocidas como DevSecOps.

**Corrección de la práctica**

El alumno deberá entregar una memoria con las soluciones a cada apartado y los pasos detallados que ha seguido en el proceso. Se valorará:

• La calidad de la memoria. Se deberán explicar y justificar los procedimientos y resultados, así como de dar soporte a lo descrito mediante capturas de pantalla.

• La corrección del proceso descrito.

• La corrección de los comandos empleados y de las opciones.

• La correcta justificación de los pasos seguidos.

• La propuesta de soluciones alternativas, documentando incluso aquellas que siendo

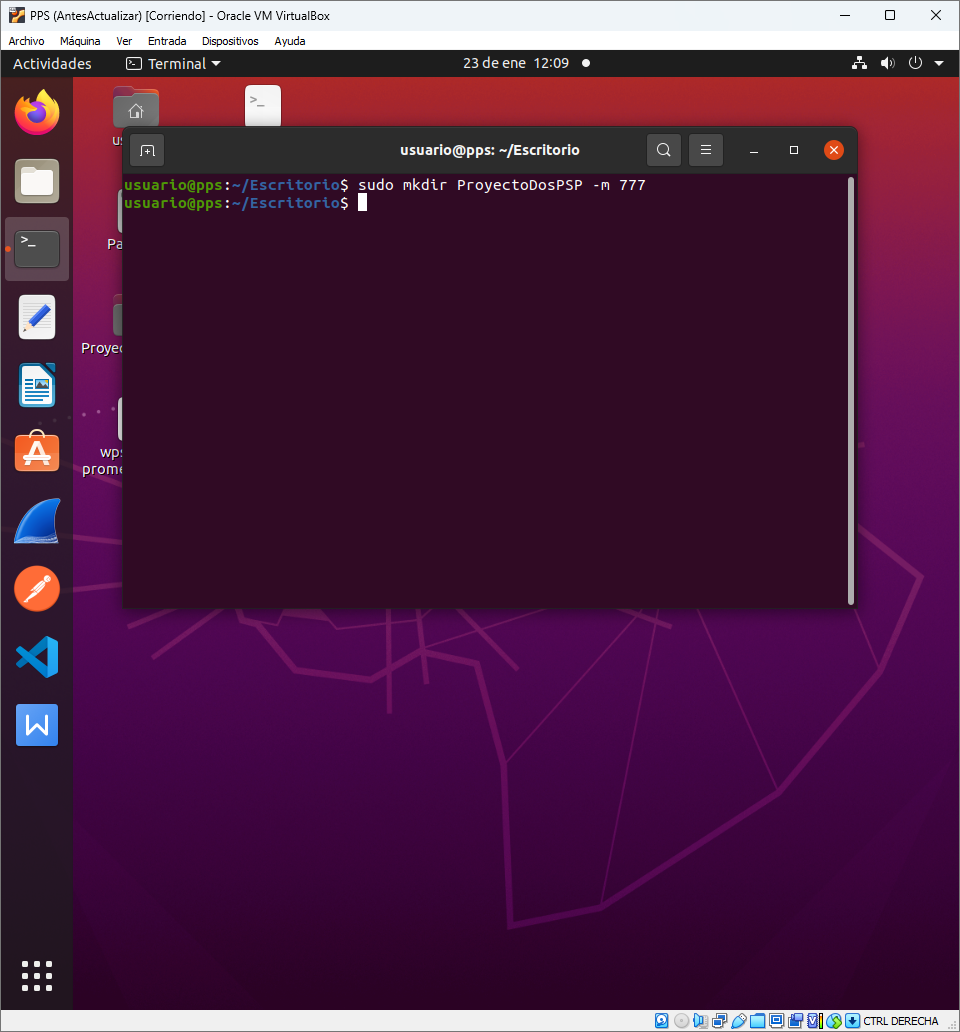
razonables no han logrado su propósito.

Forma de entrega: Vía Moodle. Se habilitará una tarea a tal efecto.

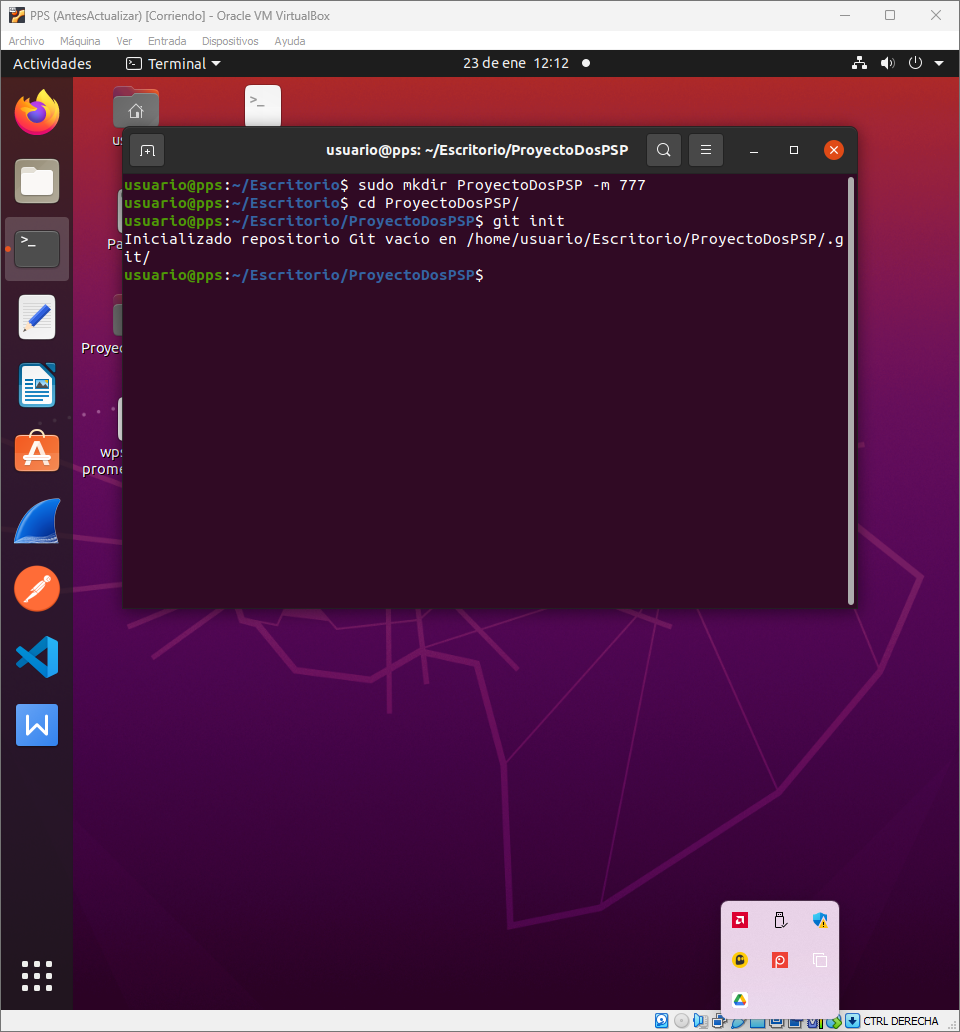
**Docker**

# Crea un Dockerfile que partiendo de una imagen PHP genera una imagen que:

Primero voy crear la carpeta en donde voy iniciar el repositorio de GitHub local, esto la haré con el comando *“sudo mkdir ProyectoDosPSP -m 777”*, la opción -m sirve para cambiar los permisos del directorio creado.



Ahora iniciaré el repositorio de GitHub localmente, esto la haré ejecutando el comando *“git init”*,el comando inicializa el repositorio el local el la carpeta donde estoy.



Ahora asociaré el repositorio local con el remoto, esto se hará con el comando *“git remote add ./ y la url del repositorio”*.

## 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).

## Crea un Dockerfile que partiendo de una imagen Ubuntu genera una imagen que:

### Instala Apache, de forma que se exponga el puerto 80.

### Instala PHP.

### 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 empleebases de datos (ya que si no también habría que instalar un MySQL).

## Crea un contenedor para cada una de esas imágenes y verifica que funciona. Para y borra dicho contenedor.

## 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.

**Git**

# Mantén un repositorio de Git público con:

## • Memoria de la práctica.

## • El fichero Dockerfile.

## Haz un *commit* y un *push* del proyecto cada vez que superes uno de los ítems de la sección anterior.