

Pregunta 3

Nombre	
Apellidos	
Itinerario	No colaborativo

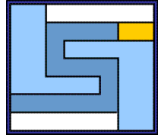
- Partiendo de una instalación limpia de Docker, indique los pasos que seguiría para:
 - Descargar la imagen de ubuntu en su versión 22.04.
 - Entrar dentro de un contenedor de dicha imagen para modificar algunos archivos.
 - Crear una nueva imagen a partir de dicho contenedor modificado. El nombre de la nueva imagen será ubuntucustom.
- Teniendo en cuenta la siguiente salida, ¿cómo podemos borrar todas las imágenes con una sola instrucción?

```
andres@andresPC:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
flaskexample         latest             8c486a784d7a       2 months ago       925MB
<none>              <none>            8691ef6be089       2 months ago       925MB
decide_web           latest             f8d6f837ac1b       2 months ago       782MB
decide_nginx         latest             3898a7f752e3       2 months ago       22.9MB
```

- Teniendo en cuenta la siguiente salida, ¿cómo podemos borrar el segundo contenedor?

```
andres@andresPC:~/decActions/docker$ docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS              PORTS
0f151e1f2b63   flaskexample     "python ./holamundo..." 2 months ago    Exited (0) 2 months ago
c9eba4f95437   flaskexample     "python ./holamundo..." 2 months ago    Up 8 seconds
5f4d9fa2fab3   8691ef6be089     "python ./holamundo..." 2 months ago    Exited (0) 2 months ago
44b6b3b864d3   8691ef6be089     "python ./holamundo..." 2 months ago    Exited (0) 2 months ago
```

- Escriba un archivo Dockerfile que parta de la imagen python:alpine y haga lo siguiente:
 - Copiar la carpeta local "code" a la ruta del contenedor "/app/www"
 - Cambiar el espacio de trabajo a "/app/www"
 - Instalar las dependencias de python que se encuentra en "/app/www/requirements.txt"
 - Ejecutar el script "/app/www/prepareinstall.sh"
 - Prepara la imagen para que, cuando sea lanzada en un contenedor, ejecute el script "/app/web/runserver.sh"
- Escriba un docker-compose.yml que tenga dos servicios.
 - El servicio 1:
 - Se llamará datos.
 - La imagen de partida será postgres:alpine.
 - El servicio 2:
 - Se llamará web.
 - Sólo se creará si el servicio datos se ha creado satisfactoriamente.
 - Partirá de la imagen myimage:latest. Esa imagen se construirá a partir del Dockerfile anterior (tenga en cuenta que tanto el Dockerfile como este docker-compose.yml se encuentran en la misma carpeta).
 - Redirigirá el tráfico recibido por el puerto 8080 al puerto 80 del contenedor.
- Indique en carpeta de su repositorio debería colocar los archivos yml correspondiente a los workflows de GitHub Actions. Y explique (no es necesario que escriba el código) qué pasos mínimos deberían hacerse para que dicho workflow pueda despliegue el código en heroku.
- Escriba un workflow que se lance ante cualquier push en las ramas master o develop y haga estos pasos en un mismo job:



- (a) Descargar su código del repositorio.
 - (b) Imprimir por pantalla el contenido descargado.
 - (c) Instalar los requisitos de python que se encuentran en requirements.txt dentro de la carpeta actual.
 - (d) Corra la instrucción `manage.py test` dentro de carpeta `app`
8. Partiendo del siguiente workflow escrito en el archivo `wf1.yml`. Haga los cambios necesarios para lanzarlo con las siguientes configuraciones (haga uso de las matrices)

```
name: WF1
on:
  push
jobs:
  build:
    runs-on: ubuntu-latest
    services:
      postgres:
        image: postgres:11
        ports:
          - 5432:5432
    steps:
      - name: Set up Python 3.8
        uses: actions/setup-python@v2
        with:
          python-version: 3.8
```

- (a) Versiones 2.8, 2.9 y 3.2 de python.
 - (b) Versiones 11 y 12 de postgres.
9. Modifique los workflows anteriores para hacer lo siguiente:
- (a) El WF1 sea invocable por otro workflow además de al hacer push.
 - (b) Que el primer workflow tenga un segundo job que invoque al workflow WF1.
 - (c) Este segundo job sólo se lanzará si el primer job ha terminado satisfactoriamente.