**Resultado Daniel Muñoz Palma**

**Preguntas de Python**

1. **¿Qué devuelve esta función en llamadas consecutivas?**

def append\_to\_list(value, my\_list=[]):

my\_list.append(value)

return my\_list

print(append\_to\_list(1))

print(append\_to\_list(2))

print(append\_to\_list(3))

**Respuesta:**

[1]

[1, 2]

[1, 2, 3]

1. **Considera el siguiente código:**

import threading

counter = 0

def increment():

global counter

for \_ in range(1000000):

counter += 1

t1 = threading.Thread(target=increment)

t2 = threading.Thread(target=increment)

t1.start()

t2.start()

t1.join()

t2.join()

print(counter)

**¿Qué puedes decir sobre la salida?**

**Respuesta:** El código está usando dos hilos (ej: como si tuviera 2 trabajadores) para aumentar una variable counter 1,000,000 de veces cada uno. Como resultado da 2000000

1. **¿Cuál será la salida de este código?**

class A:

x = 10

class B(A):

pass

class C(A):

x = 20

print(B.x, C.x)

B.x = 30

print(A.x, B.x, C.x)

**Respuesta:**

10 20

10 30 20

**Preguntas de Docker**

1. **Ejecutas un contenedor PostgreSQL así:**

docker run -d --name db -e POSTGRES\_USER=user -e POSTGRES\_PASSWORD=pass

postgres

Luego, dentro de otro contenedor, intentas conectarte usando:

import psycopg2

conn = psycopg2.connect(host="localhost", database="postgres",

user="user", password="pass")

**¿Por qué falla la conexión y cómo puedes solucionarlo?**

**respuesta:** El error ocurre porque se intentando conectarte a localhost desde otro contenedor, pero localhost dentro de un contenedor se refiere a sí mismo, no al contenedor de PostgreSQL (db). Los contenedores son entornos aislados y no comparten localhost por defecto.

docker run -d --name db -e POSTGRES\_USER=user -e POSTGRES\_PASSWORD=pass

postgres

import psycopg2

conn = psycopg2.connect(

host="db", # Nombre del contenedor PostgreSQL

database="postgres",

user="user",

password="pass"

)

1. **Construyes una imagen usando:**

docker build -t myapp .

Luego ejecutas un contenedor:

docker run -d --name app myapp

Más tarde, modificas tu Dockerfile y reconstruyes:

docker build -t myapp .

Pero cuando reinicias el contenedor:

docker restart app

Solución: docker restart **no esta actualizando la imagen**  
Al reiniciar el contenedor (app), este sigue usando la versión **antigua** de la imagen (myapp), ignorando los cambios del nuevo Dockerfile.