Programación de Servicios y Procesos

Ejercicio opcional 11:
Productor - Consumidor:

Basándote en el ejercicio de la teoría amplía el programa de esta forma:

Pon 2 productores:

- Productor 1: Añade frutas (está especializado en fruta)
- Productor 2: Añade verduras

Tienes 2 consumidores que irás sacando frutas o verduras, no hace falta controlar esto.

Ve mostrando por pantalla que van haciendo:

- Productor 1 introdujo una fruta (si puedes poner el nombre de una fruta mejor)
- Productor 2 introdujo una verdura (si puedes poner el nombre de una verdura mejor)
- Consumidor 1: Ha sacado de la cesta (pon si es fruta, verdura o el nombre) producto
- Consumidor 2: Ha sacado de la cesta (pon si es fruta, verdura o el nombre) producto

Si te apetece pensar un extra (sin obligación de nada), podrías al final indicar cuántas frutas, verduras o el nombre de las mismas se ha llevado cada consumidor.

Ejercicio de la teoría:

```
import threading
import time
import random
# Variables globales
BUF_SIZE = 10
buffer = [None]*BUF_SIZE
# Semáforos
semaforo_productor = threading.Semaphore(BUF_SIZE) # Controla la cantidad máxima de productos en el buffer
semaforo_consumidor = threading.Semaphore(0) # Inicialmente, el buffer está vacío
# Hilo productor
class Productor(threading.Thread):
def run(self):
global BUF_SIZE, buffer
global semaforo_productor, semaforo_consumidor
indice_entrada = 0
for i in range(20):
time.sleep(random.uniform(0.1, 0.5)) # Simula la producción de un elemento
item = f"Item {i+1}" #Producido el elemento i
semaforo_productor.acquire()
buffer[indice_entrada] = item
indice_entrada = (indice_entrada +1)%BUF_SIZE
print(f"El producto ha añadido el elemeno {item}")
semaforo_consumidor.release()
# Hilo consumidos
class Consumidor(threading.Thread):
```

```
def run(self):
global BUF_SIZE, buffer
global semaforo_productor, semaforo_consumidor
indice_salida = 0
for i in range(20):
time.sleep(random.uniform(0.1, 0.5)) # Simula el tiempo de procesamiento
semaforo_consumidor.acquire()
producto = buffer[indice_salida]
indice_salida = (indice_salida +1)%BUF_SIZE
print(f"El consumidr ha sacado el producto {producto}")
semaforo_productor.release()
if __name__ == "__main__":
# Creamos hilos para el productor y el consumidor
hilo_productor = Productor()
hilo_consumidor = Consumidor()
# Iniciamos los hilos
hilo_productor.start()
hilo_consumidor.start()
# Esperamos a que ambos hilos terminen
hilo_productor.join()
hilo_consumidor.join()
print("Todas las operaciones han finalizado")
```