Actividad: “Simulación de descarga de archivos con hilos”

Objetivo

Aprender a crear y gestionar varios hilos (threads) en Python usando el módulo threading para acelerar tareas de entrada/salida, como la descarga de archivos.

Paso 1: Preparar la función de descarga

1. Abre tu editor y crea el archivo descarga_multihilo.py.
2. Escribe una función que simule la descarga de un archivo usando time.sleep():

Paso 2: Crear la lista de archivos a descargar

1. Añade una lista de nombres de archivos ficticios:

```python
archivos = [
    "documento1.pdf",
    "foto2.jpg",
    "musica3.mp3",
    "video4.mp4",
    "datos5.csv"
]
```

Paso 3: Crear y lanzar hilos

1. Importa threading y crea un hilo para cada descarga:

Paso 4: Esperar a que todos los hilos terminen

1. Usa .join() para asegurarte de que el programa no termina hasta que todos los archivos estén “descargados”:

Paso 5: Prueba y Reflexión

1. Ejecuta el programa:
2. Observa cómo las descargas se procesan en paralelo y terminan en diferentes tiempos.

3. Responde:
    - ¿En qué orden se terminan las descargas?
    - ¿Cómo cambia el tiempo total si descargas secuencialmente (uno por uno, sin hilos)?
    - ¿Para qué tipo de tareas resulta útil el uso de hilos?

Conclusiones

- Los hilos permiten ejecutar tareas de entrada/salida simultáneamente, lo que mejora la eficiencia y la experiencia de usuario cuando hay que esperar por recursos externos.
- No aumentan el rendimiento en cálculos intensivos de CPU debido al GIL de Python, pero sí para operaciones como descargas, acceso a bases de datos o lectura de archivos.
- Este patrón se usa en gestores de descargas, aplicaciones web, y programas que monitorean dispositivos al mismo tiempo.

Realiza variantes creando más archivos o usando una función que lea datos remotos reales si lo deseas para profundizar la práctica.