

Productor-consumidor: Sincronización con colas en Python

[crayon-6a9d57a8d0b94639951558/]

Este código hace lo siguiente:

1. **Importaciones:** Importa los módulos necesarios (threading, queue, time, y random).
2. **Cola:** Crea una cola q con una capacidad máxima de 10 elementos.
3. **Productor:** Define la función del productor que produce 20 elementos aleatorios y los coloca en la cola. Después de producir todos los elementos, el productor coloca None en la cola para señalar a los consumidores que ha terminado.
4. **Consumidor:** Define la función del consumidor que consume los elementos de la cola. Si el consumidor encuentra None, sale del bucle y termina.
5. **Hilos:** Crea e inicia los hilos para el productor y el consumidor.
6. **Sincronización:** Espera a que ambos hilos terminen utilizando join.

¿Por qué funciona correctamente?

El uso de la cola (queue.Queue) maneja la sincronización entre el productor y el consumidor de manera automática:

- **Bloqueo y Desbloqueo Automático:** La cola se encarga de bloquear el productor si la cola está llena y el consumidor si la cola está vacía, gestionando así el flujo de manera segura y evitando que se ejecuten

incorrectamente múltiples hilos a la vez.

- **Sin Necesidad de Locks Adicionales:** No es necesario utilizar locks adicionales (`threading.Lock`) porque la cola maneja la sincronización interna.