

Ejercicio Productor – Consumidor [PC]

Ejercicio PC1

Crea un programa donde se instancian dos hilos (o procesos) de dos tipos distintos:

- **Hilo1-productor.** Crea **UN** número aleatorio del 1 al 10 y lo inserta en una variable [podéis usar una cola, Queue]. Luego avisa al **Hilo2-consumidor** para poder consumirlo (imprimirlo).
- **Hilo2-consumidor.** Espera a que el **Hilo1-productor** le avise para poder consumir el elemento. Cuando lo consume (imprime) libera la Condición para que el **Hilo1-productor** pueda comenzar su proceso de producir un nuevo elemento.

Ejercicio PC2

Modificar el programa **PC1** para que el **Hilo1-productor**, produzca un número aleatorio N [1..10] de números aleatorios y los inserte en la cola. Una vez insertados se bloquea y comunica al **Hilo2-consumidor** que ya puede consumir.

A continuación, el **Hilo2-consumidor** intentará consumir otro número aleatorio N [1..10] de elementos, siempre y cuando tenga los suficientes elementos en la cola para poder consumirlos. Si no hay suficientes elementos se bloqueará y comunicará al **Hilo1-productor** que vuelva a producir. Por ejemplo, si la cola tiene 6 elementos y queremos consumir 8, como no hay suficientes, el **Hilo2-consumidor** deberá ponerse en espera (bloqueo) y permitir que el **Hilo1-productor** inserte más elementos en la cola.

Cuando nuevamente el **Hilo1-productor** haya generado más elementos, el **Hilo2-consumidor** volverá a intentar consumirlos siempre que haya suficiente cantidad de elementos en la cola.

El **Hilo2-consumidor** nunca podrá consumir más elementos de los que existen en la cola.

Ejercicio PC3

Modificamos el **PC2**, y ahora queremos incluir una modificación en el **Hilo2-consumidor**. Este será capaz de consumir mientras haya elementos, independientemente de que la cantidad de elementos que quiera consumir sea mayor de los que haya.

Eso sí, si la cola se vacía, deberá liberar la condición para que el **Hilo1-productor**, pueda añadir más de nuevo. Por ejemplo, si la cola tiene 6 elementos y queremos consumir 8, el **Hilo2-consumidor** consumirá hasta que la cola esté vacía (6 elementos) y se pondrá a la espera (bloqueo) de que el **Hilo1-productor** inserte más elementos en la cola.

En resumen, el **Hilo2-consumidor** podrá consumir siempre, pero tiene que comprobar que la cola no esté vacía, de esta forma evitaremos el error de leer de una cola vacía y evitar interbloqueos.

Ejercicio PC4

Modificamos el **PC3**, y ahora queremos incluir una modificación en el **Hilo1-productor**. Tenga un límite de tamaño de la cola (pondremos tamaño 30). El **Hilo1-productor** será capaz de producir mientras no se llene el almacén. Si la cola se llena debe ponerse a la espera de que el **Hilo2-consumidor** consuma elementos suficientes para seguir produciendo.

Ejercicio PC5 (opcional)

Finalmente, vamos a crear un programa que permita lanzar varios hilos productores y consumidores gestionando una (o varias colas de producción) y poder gestionar e intercalar entre sí los procesos Productores y Consumidores.