

1 Problema del productor-consumidor

Un problema típico que se suele dar al usar la concurrencia, es el del productor-consumidor. En el **problema del productor-consumidor** se dispone uno o varios hilos que producen información y uno o varios hilos que deben consumir dicha información.

Esta situación se suele dar por ejemplo en la cola de impresión de la impresora.

Para compartir información entre los hilos se dispone de un buffer en el que los productores van escribiendo sus datos en una posición que se llamará cabeza y los consumidores van leyendo la información desde una posición llamada cola:

cola					cabeza				
vacío	vacío	lleno	lleno	lleno	lleno	lleno	lleno	vacío	vacío

La solución al problema del productor-consumidor usando semáforos es sencilla y necesita de:

- binario: Es un semáforo binario que va a controlar la lectura y escritura en el buffer.
- lleno: Es un semáforo, inicialmente con el contador a cero, que indicará el número de posiciones llenas en el buffer.
- vacío: Es un semáforo, inicialmente con el contador marcando el número de posiciones del buffer, que representa el número de posiciones vacías en el buffer.
- cabeza y cola: Son dos enteros que representan las posiciones del buffer cabeza y cola. Inicialmente tendrán el valor 0.
- N: Es un entero que representa el tamaño del buffer.

El pseudocódigo de la solución sería la siguiente:

```
class Productor {
...
void run() {
    Se produce el dato
    vacío.acquire();
    binario.acquire();
    buffer[cabeza] = dato;
    cabeza = (cabeza + 1) % N;
    binario.release();
    lleno.release();
}
...
}
```

```
class Consumidor {
...
void run() {
    lleno.acquire();
    binario.acquire();
    dato = buffer[cola];
    cola = (cola + 1) % N;
    binario.release();
    vacío.release();
    // Se consume el dato
}
}
```

```
...  
}
```

Ejercicios:

1. Escriba un programa en Java que consista en un productor y un consumidor. El productor debe generar los números del 1 al 100 y el consumidor debe mostrarlos en pantalla.
2. Modifique el ejercicio anterior para tener 10 productores y 10 consumidores, produciendo y leyendo datos del mismo buffer. Cada productor debe generar los números del 1 al 100. Al final, se habrán generado en total 1000 números, 10 series del 1 al 100.
3. Escriba un programa en Java con dos hilos, uno debe ir obteniendo los 100 primeros términos de la sucesión Fibonacci y mostrarlos en pantalla. El otro debe ir sumándolos y mostrando la suma.
Nota: La sucesión se obtiene sumando los dos términos anteriores comenzando por el 0 y el 1: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
Curiosidad: La sucesión Fibonacci, por azares de la genética, aparece en muchos seres vivos. Por ejemplo, en un girasol las pipas forman espirales unas en sentido horario y otras en sentido antihorario. El número de unas y de otras son términos de la sucesión Fibonacci. Las escamas de una piña también crecen siguiendo la sucesión de Fibonacci. Se recomienda ver el vídeo <https://www.youtube.com/watch?app=desktop&v=ctRIK56jV3k>
4. En el ejercicio anterior los números de la sucesión Fibonacci, ¿se vuelven negativos a partir de un determinado valor! ¿A qué se debe esto? Solución: <https://stackoverflow.com/questions/15004944/max-value-of-integer>
5. Escriba un programa en Java con 5 hilos que produzca los 10 primeros términos de la sucesión Fibonacci cada uno. Debe haber un hilo adicional que toma estos valores y los va mostrando por pantalla.
6. Escriba un programa en Java con un hilo que produzca los 10 primeros términos de la sucesión Fibonacci. Debe haber un 5 hilos adicional que toman estos valores y los muestran en pantalla multiplicados por 2.