

# Tarea para PSP02.

Detalles de la tarea de esta unidad.

## Enunciado.

### Ejercicio 1

De igual manera a lo visto en el tema, ahora te proponemos un ejercicio del tipo productor-consumidor que mediante un hilo productor almacene datos (15 caracteres) en un búfer compartido, de donde los debe recoger un hilo consumidor (consume 15 caracteres). La capacidad del búfer ahora es de 6 caracteres, de manera que el consumidor podrá estar cogiendo caracteres del búfer siempre que éste no esté vacío. El productor sólo podrá poner caracteres en el búfer, cuando esté vacío o haya espacio.

Te mostramos una posible salida del programa que debes realizar

```
run:
Depositado el carácter T en el buffer
Recogido el carácter T del buffer
Depositado el carácter R en el buffer
Recogido el carácter R del buffer
Depositado el carácter J en el buffer
Depositado el carácter W en el buffer
Depositado el carácter L en el buffer
Depositado el carácter Y en el buffer
Depositado el carácter M en el buffer
Depositado el carácter P en el buffer
Recogido el carácter P del buffer
Depositado el carácter J en el buffer
Recogido el carácter J del buffer
Depositado el carácter Q en el buffer
Recogido el carácter Q del buffer
Depositado el carácter X en el buffer
Depositado el carácter H en el buffer
Recogido el carácter X del buffer
Recogido el carácter H del buffer
Depositado el carácter G en el buffer
Recogido el carácter G del buffer
Depositado el carácter O en el buffer
Recogido el carácter O del buffer
Depositado el carácter U en el buffer
Recogido el carácter U del buffer
Recogido el carácter M del buffer
Recogido el carácter Y del buffer
Recogido el carácter T del buffer
```

Observa:

- \*Se comienza depositando.
- \*Se pueden depositar seguidos hasta 6 caracteres.
- \*Cuando el búfer está lleno, la única opción es consumir.

### Ejercicio 2

De igual manera a lo visto en el tema, ahora te proponemos que resuelvas el clásico problema denominado "La cena de los filósofos" utilizando la clase `Semaphore` del paquete `java.util.concurrent`.

El problema es el siguiente: Cinco filósofos se sientan alrededor de una mesa y pasan su vida comiendo y pensando. Cada filósofo tiene un plato de arroz chino y un palillo a la izquierda de su plato. Cuando un filósofo quiere comer arroz, cogerá los dos palillos de cada lado del plato y comerá. El problema es el siguiente: establecer un ritual (algoritmo) que permita comer a los filósofos. El algoritmo debe satisfacer la exclusión mutua (dos filósofos no pueden emplear el mismo palillo a la vez), además de evitar el interbloqueo y la inanición.



Te mostramos una posible salida del programa que debes realizar.

Elementos observados	Salida - TareaCenaFilosofos
run:	
Filósofo 1 Pensando	Observa que:
Filósofo 4 Pensando	* Dos filósofos contiguos no
Filósofo 3 Pensando	pueden estar comiendo a la vez.
Filósofo 2 Pensando	
Filósofo 5 Pensando	* El proceso no debeterminar hasta
Filósofo 2 Hambriento	que tú fuerces su finalización.
Filósofo 2 Comiendo	
Filósofo 1 Hambriento	
Filósofo 3 Hambriento	
Filósofo 2 Termina de comer, Libres palillos:2,1	
Filósofo 3 Comiendo	
Filósofo 2 Pensando	
Filósofo 1 Comiendo	
Filósofo 4 Hambriento	
Filósofo 3 Termina de comer, Libres palillos:3,2	
Filósofo 3 Pensando	
Filósofo 4 Comiendo	
Filósofo 5 Hambriento	
Filósofo 1 Termina de comer, Libres palillos:1,5	
Filósofo 1 Pensando	
Filósofo 1 Hambriento	
Filósofo 3 Hambriento	

**Criterios de puntuación. Total 10 puntos.**

Total 10 puntos.

- **Ejercicio 1.** 5 puntos.
- **Ejercicio 2.** 5 puntos.

Se tendrá en cuenta que:

- La ejecución de los programas produce el resultado esperado.
- No se produce interbloqueo ni inanición.

### Recursos necesarios para realizar la Tarea.

**Ejercicio 1.-** Los contenidos y ejemplos realizados en la Unidad.

**Ejercicio 2.-** Además de los contenidos y ejemplos de la unidad, se proporciona el archivo **PSP02\_Recurso\_Tarea.zip** con indicaciones sobre las clases y métodos que puedes crear para resolver el problema. Para ver esa información, generada mediante JavaDoc, descomprime el archivo y ejecuta **index.html**.

[PSP02\\_Recurso\\_Tarea.zip](#)

### Consejos y recomendaciones.

Para el **Ejercicio1.-** Ten en cuenta que ahora el problema del productor-consumidor utiliza un búfer de tamaño 6, que lo puedes implementar mediante un array. La forma de ir introduciendo caracteres será de

izquierda a derecha y se consumirán de derecha a izquierda (el último que se produzca será el primero en consumirse).

Para el **Ejercicio2**.- Lo primero que debes tener claro es si se trata de proteger una sección crítica o bien de comunicar hilos, para así utilizar la clase `Semaphore` de la forma apropiada. Consulta el "problema de la cena de los filósofos" en la wikipedia, en él se comentan diversos algoritmos para resolver el problema y además hay enlaces a la solución visual del problema.

["El problema de la cena de los filósofos" en Wikipedia](#)