



## Programación de Sistemas y Concurrencia

## Control Bloque 2-Semáforos

Informática A + Software A

Curso 2022-2023

APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

DNI \_\_\_\_\_ ORDENADOR \_\_\_\_\_ GRUPO/TITULACIÓN \_\_\_\_\_

### Descripción del sistema

Un grupo de excursionistas *adultos y menores de edad* están pasando un día de fiesta cerca de un lago. Los excursionistas pueden dar un paseo por el lago utilizando una **Barca que tiene una capacidad para  $N$  pasajeros**. La *Barca* está manejada por el *Barquero* que la pone en marcha cuando está llena, y la detiene cuando se acaba el recorrido.

Se pide implementar dos versiones de este escenario **utilizando sólo semáforos binarios para sincronizar las hebras**. En la primera de ellas, hay exactamente  $N$  excursionistas y la barca sólo realiza un viaje. En la segunda hay  $Niter * N$  excursionistas por lo que la barca tiene que hacer  $Niter$  vueltas para que todos puedan disfrutar de un paseo. En el campus virtual, se encuentran los esqueletos de las dos versiones, conteniendo las siguientes clases:

- Principal: contiene el método main que crea el recurso Barca, varias instancias de las entidades activas (Adulto y Menor) y las pone en funcionamiento.
- Adulto y Menor: son dos clases que extienden a Thread y reciben como parámetros en el constructor un identificador (id) y una referencia a la Barca (b).
- Barquero: es una clase que extiende a Thread y recibe como parámetro una referencia a la Barca (b) y, en la segunda, versión el número de vueltas (Niter) que tiene que realizar.
- La clase Barca que representa el recurso compartido, cuyos métodos hay que implementar.

### Paseo en Barca-Primera versión:

En esta versión sólo hay  $N$  excursionistas (la capacidad de la barca), por lo que **la barca da un único paseo**. Además, aunque los excursionistas se pueden subir en cualquier orden a la barca, cuando termina el paseo **deben bajarse primero los menores** y, por último, los adultos. En esta versión podría darse el caso de que solo haya excursionistas adultos o menores.

De esta forma el comportamiento del sistema debería ser el siguiente:

- (1)  $N$  excursionistas adultos o menores se suben a la barca (el último en subir avisa al Barquero)
- (2) El barquero pone en marcha la barca y comienza el paseo
- (3) El barquero avisa que se ha terminado el paseo
- (4) Los excursionistas menores se bajan primero, y después los adultos.

En consecuencia, hay que implementar 3 condiciones de sincronización

**CS-Barquero:** El barquero no pone en marcha la barca hasta que no se han subido  $N$  excursionistas.

**CS-Menor:** Un menor que está en la Barca no puede bajarse hasta que no se ha terminado el paseo.

**CS-Adulto:** Un adulto que está en la Barca no puede bajarse hasta que no se ha terminado el paseo y no se hayan bajado todos los menores (en caso de que se hubiera subido alguno).



Para implementar este comportamiento, hay que completar los siguientes métodos en la clase Barca:

/\*

- \* El menor id se sube a la barca. Si es el último pasajero avisa al barquero
  - \* de que la barca está completa.
  - \* Como hay un total de N excursionistas entre menores y adultos, en esta versión un menor que quiera subir podrá hacerlo sin tener que esperar
  - \* A continuación, espera hasta que finalice el paseo.
  - \* Cuando lo avisan se baja de la barca
- \*/

**public void** subeMenor(int id)

/\*

- \* El adulto id se sube a la barca. Si es el último pasajero avisa al barquero
  - \* de que la barca está completa.
  - \* Como hay un total de N excursionistas entre menores y adultos, en esta versión, un adulto que quiera subir podrá hacerlo sin tener que esperar.
  - \* A continuación, espera hasta que finalice el paseo.
  - \* Cuando lo avisan se baja de la barca
- \*/

**public void** subeAdulto(int id)

/\*

- \* El barquero espera hasta que la barca tenga N pasajeros para comenzar el paseo
- \*/

**public void** esperoLleno()

/\*

- \* Cuando termina el paseo avisa a los pasajeros que se pueden bajar.
  - \* Debe garantizarse que siempre se bajan primero los pasajeros menores.
- \*/

**public void** finViaje()

Una ejecución de este sistema con una Barca con capacidad para 10 pasajeros podría ser:

El adulto 1 se ha subido a la barca  
El menor 9 se ha subido a la barca  
El menor 5 se ha subido a la barca  
El adulto 4 se ha subido a la barca  
El menor 7 se ha subido a la barca  
El adulto 7 se ha subido a la barca  
El adulto 6 se ha subido a la barca  
El menor 1 se ha subido a la barca  
El adulto 5 se ha subido a la barca  
El menor 0 se ha subido a la barca  
Comienza el viaje!!!  
Fin del viaje!!!

El menor 9 se ha bajado de la barca  
El menor 5 se ha bajado de la barca  
El menor 7 se ha bajado de la barca  
El menor 1 se ha bajado de la barca  
El menor 0 se ha bajado de la barca  
El adulto 1 se ha bajado de la barca  
El adulto 4 se ha bajado de la barca  
El adulto 7 se ha bajado de la barca  
El adulto 6 se ha bajado de la barca  
El adulto 5 se ha bajado de la barca

## Paseo en Barca-Segunda versión:

Esta versión es una extensión de la anterior, asumiendo que hay más de N excursionistas, por lo que la barca tiene que dar varios paseos para que todos puedan montarse. Para facilitar la implementación, el constructor de la clase Barca recibe un parámetro adicional (**int menoresEnBarca**) que indica el número de menores que se van a subir a la Barca en cada paseo (**menoresEnBarca** es mayor que 1 y menor que la capacidad N de la barca).

De esta forma, en cada paseo se suben a la barca

- **menoresEnBarca** menores y
- (N- **menoresEnBarca**) adultos

Es decir, puede haber más menores y adultos entre los excursionistas, pero en cada paseo se subirán exactamente **menoresEnBarca** menores a la barca y el resto serán adultos. Además, **los menores deben subirse a la barca antes que los adultos**. En la clase Principal se han calculado los valores de las constantes N y **menoresEnBarca** para que con N iter paseos se puedan pasear todos los excursionistas. **NOTA: En esta versión hay que volver a implementar los métodos subeMenor(int id) y subeAdulto(int id) y se reutilizan de la Versión 1 la implementación de los métodos esperoLleno() y finViaje().**

De esta forma, el funcionamiento de esta segunda versión sería:

- (1) **menoresEnBarca** menores se suben a la Barca
- (2) N- **menoresEnBarca** adultos se suben a la barca (el último en subirse avisa al barquero)
- (3) El barquero pone en marcha la barca y comienza el paseo
- (4) El barquero avisa que se ha terminado el paseo
- (5) Los pasajeros menores se bajan y, después, los pasajeros adultos. El último adulto en bajar avisa a los menores que están esperando para dar un paseo de que ya pueden hacerlo.

Por lo tanto, además de las 3 condiciones de sincronización ya implementadas en la primera versión, hay que añadir las dos siguientes:

**CS-2v-Menor:** Un menor tiene que esperar si no hay sitio libre, o si aún no se han bajado los pasajeros del paseo anterior (si no es el primer paseo).

**CS-2v-Adulto:** Un adulto no puede subirse a la barca hasta que no se han subido los menores.

Una posible ejecución de este sistema con una Barca con capacidad para 10 pasajeros (6 adultos y 4 menores) podría ser:

El menor 6 se ha subido a la barca  
El menor 1 se ha subido a la barca  
El menor 11 se ha subido a la barca  
El menor 10 se ha subido a la barca  
El adulto 2 se ha subido a la barca  
El adulto 8 se ha subido a la barca  
El adulto 17 se ha subido a la barca  
El adulto 4 se ha subido a la barca  
El adulto 15 se ha subido a la barca  
El adulto 7 se ha subido a la barca  
Comienza el viaje!!!  
Fin del viaje!!!  
El menor 6 se ha bajado de la barca  
El menor 1 se ha bajado de la barca  
El menor 11 se ha bajado de la barca  
El menor 10 se ha bajado de la barca  
El adulto 2 se ha bajado de la barca  
El adulto 8 se ha bajado de la barca  
El adulto 17 se ha bajado de la barca  
El adulto 4 se ha bajado de la barca  
El adulto 15 se ha bajado de la barca  
El adulto 7 se ha bajado de la barca

El menor 8 se ha subido a la barca  
El menor 4 se ha subido a la barca  
El menor 2 se ha subido a la barca  
El menor 7 se ha subido a la barca  
El adulto 3 se ha subido a la barca  
El adulto 5 se ha subido a la barca  
El adulto 9 se ha subido a la barca  
El adulto 14 se ha subido a la barca  
El adulto 12 se ha subido a la barca  
El adulto 10 se ha subido a la barca  
Comienza el viaje!!!  
Fin del viaje!!!  
El menor 8 se ha bajado de la barca  
El menor 4 se ha bajado de la barca  
El menor 2 se ha bajado de la barca  
El menor 7 se ha bajado de la barca  
El adulto 3 se ha bajado de la barca  
El adulto 5 se ha bajado de la barca  
El adulto 9 se ha bajado de la barca  
El adulto 14 se ha bajado de la barca  
El adulto 12 se ha bajado de la barca  
El adulto 10 se ha bajado de la barca

El menor 0 se ha subido a la barca  
El menor 5 se ha subido a la barca  
El menor 9 se ha subido a la barca  
El menor 3 se ha subido a la barca  
El adulto 11 se ha subido a la barca  
El adulto 6 se ha subido a la barca  
El adulto 16 se ha subido a la barca  
El adulto 1 se ha subido a la barca  
El adulto 13 se ha subido a la barca  
El adulto 0 se ha subido a la barca  
Comienza el viaje!!!  
Fin del viaje!!!  
El menor 0 se ha bajado de la barca  
El menor 5 se ha bajado de la barca  
El menor 9 se ha bajado de la barca  
El menor 3 se ha bajado de la barca  
El adulto 11 se ha bajado de la barca  
El adulto 6 se ha bajado de la barca  
El adulto 16 se ha bajado de la barca  
El adulto 1 se ha bajado de la barca  
El adulto 13 se ha bajado de la barca  
El adulto 0 se ha bajado de la barca