



UNIVERSIDAD  
DE MÁLAGA

| uma.es

## Programación de Sistemas y Concurrencia

Dpto. de Lenguajes y Ciencias de la  
Computación

Examen 1ª convocatoria ordinaria  
Bloque 2 – Temas 3 - 6  
Curso 2019-2020

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

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

### Descripción del sistema

Supón que una barca viaja de una orilla a otra de un río para transportar pasajeros. En la barca caben 3 pasajeros, porque si suben más se hunde. Además de los pasajeros, en la barca va el capitán que se encarga de conducir la barca de una orilla a otra. Podemos resumir el comportamiento de este sistema como sigue:

1. Un pasajero que quiere cruzar el río tiene que esperar hasta que la barca esté en su orilla y haya un asiento libre.
2. Un pasajero que está en la barca no puede bajarse hasta que haya terminado el viaje.
3. El capitán comienza un nuevo viaje cuando la barca está llena.

El comportamiento del sistema debería ser una secuencia de acciones como:

3 pasajeros se suben a la barca en la orilla norte (1) → el capitán comienza el viaje → el capitán termina el viaje → 3 pasajeros se bajan de la barca en la orilla sur (0) → 3 pasajeros se suben a la barca en la orilla sur (0) → ....

Para realizar la implementación, identificaremos las orillas con valores enteros, de manera que la orilla sur es la 0 y la orilla norte es la 1. Para cambiar la orilla podemos usar la operación  $orilla = (orilla + 1) \% 2$ ;

Se proporcionan las siguientes clases (no hay que modificarlas):

- **Principal**: Contiene el método main en el que se crea una barca, un capitán y varios pasajeros. Inicialmente los pasajeros con identificador par están en la orilla sur (0), y los pasajeros impares en la orilla norte (1).
- **Pasajero**: esta clase hereda de Thread y modela el comportamiento de un pasajero. El pasajero se sube a la barca, y transcurrido un tiempo intenta bajarse. Cuando se baja actualiza la orilla en la que se encuentra, después de un tiempo aleatorio comienza de nuevo el ciclo para volver a cruzar el río.
- **Capitan**: esta clase hereda de Thread y modela el comportamiento del capitán de la barca. El capitán espera a que la barca esté llena y cuando esto ocurre comienza el viaje hacia la otra orilla. Cuando ha llegado al destino indica a los pasajeros para que bajarse.

El ejercicio consiste en **hacer dos implementaciones de la clase `Barca`** que modela el recurso compartido:  
**1) Semáforos binarios 2) Métodos sincronizados o Locks** (a elección del alumno)

**La barca está inicialmente en la orilla norte (1).** La clase `Barca` proporciona los siguientes métodos públicos:

- `subir (int id, boolean pos)`: lo invoca un pasajero para subir a la barca. El segundo parámetro es la orilla en la que se encuentra el pasajero. Por lo tanto, hasta que la barca no esté en esa orilla y haya un asiento libre, el pasajero no podrá subir.

- `bajar (int id)`: lo invoca un pasajero cuando quiere bajar de la barca. El pasajero no podrá bajar hasta que el viaje termina. Cuando hayan bajado todos los pasajeros de un viaje, podrán subir nuevos pasajeros.

- `esperoSuban`: lo invoca el capitán para esperar a que la barca se llene y comenzar el viaje hacia la otra orilla.

- `finViaje`: lo invoca el capitán para indicar a los pasajeros que ya se ha llegado a la otra orilla y tienen que bajarse.

Notas:

1. La única documentación que puede utilizarse es la API de Java que viene en Eclipse.
2. Deben utilizarse los esqueletos que se proporcionan en el campus virtual.
3. Añade comentarios en el código de la clase `Barca` que indiquen para que se utilizan dichas variables, por ejemplo los semáforos.
4. Pon tu nombre y apellidos en cada fuente que entregues.
5. **Debes subir dos archivos `Barca.java`**, uno con la implementación basada en Semáforos binarios y otro con la implementación basada en Monitores o Locks.

Ejemplo de traza de ejecución del sistema:

Viajero 5 se sube al barco en la orilla 1  
Viajero 13 se sube al barco en la orilla 1  
Viajero 3 se sube al barco en la orilla 1  
Empieza el viaje!!!!  
Fin del viaje!!!!  
Viajero 13 se baja del barco  
Viajero 5 se baja del barco  
Viajero 3 se baja del barco  
Barca vacía...pueden subir nuevos pasajeros  
Viajero 16 se sube al barco en la orilla 0  
Viajero 10 se sube al barco en la orilla 0  
Viajero 2 se sube al barco en la orilla 0

Empieza el viaje!!!!

Fin del viaje!!!!

Viajero 16 se baja del barco

Viajero 2 se baja del barco

Viajero 10 se baja del barco

Barca vacía...pueden subir nuevos pasajeros

//El sistema no termina