

Examen Práctico 1ª Evaluación

Módulo	Programación
Duración: 80 minutos Puntuación: 6 puntos	

Normas de realización

- La entrega se realiza mediante un único fichero llamado con la primera inicial de tu nombre y tu primer apellido. Por ejemplo, para Ana Albo el **nombre** será aAlbo.java
- Todo examen entregado fuera de plazo no se corrige.
- Se permite el uso de los apuntes, o códigos utilizados por el propio alumno, así como el apoyo del API de Java en caso de ser necesario.
- Todo alumno que tenga abierto otro contenido que no sea el indicado en el punto anterior será expulsado del examen.
- Todo alumno que copie de otro será expulsado del examen.
- No se podrá abandonar el aula antes de que hayan transcurrido 30 minutos del comienzo del examen.

Enunciado

En una peluquería hay 2 barberos y 4 sillas para los clientes. Cuando un cliente llega a la barbería se sienta en una silla y espera a ser atendido. Pero puede que no haya ninguna silla libre, en cuyo caso se marcha.

Escribe un programa **multihilo** en Java que simule el comportamiento de la barbería teniendo en cuenta que se generan clientes continuamente, hasta un máximo de 30 clientes (con un intervalo de tiempo aleatorio), algunos encuentran silla (y se sientan a esperar a ser atendidos) y otros no. Los que no consigan silla desaparecen (terminan su ejecución).

No debe haber inanición, es decir ningún cliente debería quedarse en una silla esperando un tiempo infinito a ser atendido.

Se recomienda definir 4 clases:

- Peluqueria: Representa a la peluquería. Tiene el método main y se ocupa de lanzar los hilos de los barberos y los clientes. Cuenta con dos barberos, un array de clientes y un gestor de sillas. Cuando finalizan todos los clientes se cierra la peluquería.
- Barbero: Representa a un barbero. Recibe como parámetro en su constructor un objeto compartido GestorSillas.

- Cliente: Representa a un cliente. Recibe como parámetro en su constructor un objeto compartido GestorSillas.
- GestorSillas: Cuenta con:
 - un array de sillas (booleano) que indica si la silla está libre (true) u ocupada (false).
 - otro array de booleanos que indica si el cliente de esa silla está siendo atendido o no.
 - un método sincronizado getSillaLibre: que devuelve la posición de la silla libre (si hay alguna), para que el cliente pueda sentarse. Si no hay ninguna silla libre, este método devuelve -1, el cliente se marcha y finaliza su ejecución.
 - un método liberarSilla, que marca la silla indicada como libre (cuando el barbero ha terminado de atender al cliente que estaba en ella).
 - un método asignarCliente: que asigna al barbero uno de los clientes que están esperando a ser atendidos.

Será fundamental dividir el problema en distintos métodos que deben estar comentados en su cabecera para una mejor organización y lectura del código.