



## Programación de Sistemas y Concurrencia

Control Bloque 2, Temas 3 - 6  
Curso 2020-2021

Computadores A + Informática D + Software D

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

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

### Descripción del sistema

Se ha inaugurado una Cafetería “take-away” en Teatinos que tiene un aforo limitado (MAX\_AFORO). En la cafetería trabaja un único Barista encargado de preparar el café y cobrar a los clientes. Un Cliente puede entrar a la cafetería si no se ha llegado al aforo máximo, espera a que el Barista sirva un café, paga y sale de la Cafetería. Por otro lado, el Barista espera a que haya algún Cliente, prepara una taza de café y espera a que algún cliente le pague por ella. Vamos a modelar este sistema utilizando las siguientes clases:

- Barista: Es una clase que hereda de Thread y que se encarga de preparar cafés y cobrar a los clientes en la cafetería (se da implementada).
- Cliente: Esta clase hereda de Thread y modela a un cliente que intenta entra a la cafetería. Una vez dentro, espera hasta que consigue un café servido por el Barista, lo paga y sale de la Cafetería (se da implementada).
- Cafetería: Esta clase modela el recurso compartido. En la cafetería puede haber como máximo MAX\_AFORO Clientes. Esta clase proporciona los siguientes métodos que utilizan los Clientes y el Barista:

**public void** entrarCafeteria(**int** id): Un Cliente utiliza este método para entra a la cafetería. Solo puede entrar si no se ha alcanzado el aforo máximo.

**public void** esperarCafe(**int** id): Un Cliente espera en este método hasta que puede coger una taza de café preparada por el Barista.

**public void** pagarySalir(**int** id): Un Cliente utiliza este método para pagar el café que ha cogido y salir de la cafetería.

**public void** prepararCafeyCobrar(): El Barista espera a que haya al menos un Cliente en la cafetería para preparar un café, lo sirve y espera a que un Cliente le pague. En cada ejecución de este método, el Barista preparará un único café y por tanto solo un Cliente será atendido. Los Clientes no tienen que atenderse en orden de llegada.

Además, la clase Cafetería incluye un método principal donde se crea una Cafetería y varios Clientes. La ejecución del programa puede producir, entre otras, la siguiente ejecución:

Cliente 1 entra → Cliente 2 entra → Café servido esperando pago → Cliente 2 coge un café → Cliente 3 entra → Cliente 2 paga y sale → Café servido esperando pago → Cliente 1 coge un café → Cliente 1 paga y sale → ...

Se pide realizar dos implementaciones de la clase Cafetería utilizando, una del tipo 1 y otra del tipo 2:

Tipo 1: Semáforos binarios (5 puntos) o Semáforos generales (3 puntos)

Tipo 2: Monitores (5 puntos) o Locks (5 puntos)