

## Enunciado Práctica Evaluable 2

# Almacén de pedidos online

### Objetivo

Que el alumno ponga en práctica los conocimientos teóricos y prácticos adquiridos en la Parte II (Tema 5 en adelante) de la asignatura de Programación Concurrente.

### Normativa, fecha y mecanismo de entrega

La práctica podrá realizarse de forma individual o por parejas.

La fecha límite para la entrega de la práctica será el día indicado en la actividad correspondiente en el Aula Virtual, a través del cual se realizará la entrega de la práctica. Se deberá entregar un .zip cuyo contenido será el proyecto Eclipse en el que se ha realizado la práctica. El nombre del fichero .zip tiene que ser igual al identificador del alumno en la URJC (la parte antes de la @ del correo electrónico del alumno en la URJC). En caso de que la práctica haya sido realizada por dos alumnos, se incluirán ambos nombres separados por “-” y ambos alumnos deben entregar la práctica de manera independiente.

Además del código fuente, se deberá elaborar una breve memoria explicativa del mismo en formato PDF. La memoria deberá guardarse en la carpeta src/main/resources y tendrá el nombre “memoria.pdf”. Deberá incluir como mínimo, las sincronizaciones y las estructuras de datos usadas.

Los alumnos que hayan cursado esta asignatura el curso anterior y que hubieran aprobado la práctica 2 en dicho curso no tendrán que volver a realizarla. Esta práctica será convalidada por defecto con la práctica 2 del curso anterior, asignando la nota que se obtuviera. No obstante, si el alumno desea realizar la práctica nuevamente, puede hacerla y entregarla en forma y fecha.

### Enunciado

Una tienda de muebles a medida, llamado **Ukea**, decide simular en un programa informático concurrente la interacción que se produce en cada una de sus tiendas. Esta simulación les permitirá analizar de forma más precisa su funcionamiento y podrán optimizar sus procesos.

Se modelarán diversos tipos de personas:

- **Cliente:** Representa a los clientes que van a realizar pedidos en una tienda.
- **Empleado:** Representa a todos los tipos de empleados de Ukea. Esta categoría se divide en las siguientes:
  - **EmpCogePedidos:** Empleados encargados de atender a los clientes para tomarles el pedido y, cuando ha sido dispensado, cobrarles.
  - **EmpElaboraProductos:** Existen algunos productos especiales que necesitan una preparación previa, ya que se componen a su vez de varios productos. Los empleados encargados de elaborar productos se encargan de esta labor.

- **EmpPreparaPedidos:** Empleados encargados de entregar los pedidos a los clientes con todos los productos que han pedido. Lo hacen en un dispensador de pedidos.
- **EmpEncargado:** Encargado de la tienda, que supervisa el funcionamiento de la misma.

La tienda recibe pedidos las 14 horas que está abierta, no obstante, los trabajadores solo trabajan en dos turnos de 7 horas cada uno.

A continuación, se describe el ciclo de vida de cada uno de los procesos:

- **Cliente:** Un cliente podrá hacer un pedido en cualquier instante del horario de apertura. Cuando no hace un pedido realizará su vida normal (se supone que hace compras cada cierto tiempo aleatorio).
- **Empleados:** Todos los empleados realizarán su vida normal y luego irán al trabajo. Empezarán a trabajar cuando el encargado se los indique y acabarán cuando el encargado le diga que tienen que dejar de trabajar. En ese momento se marcharán de la tienda. Realizarán su vida cotidiana y volverán a trabajar cuando el encargado les avise.
- **EmpCogePedidos:** Este empleado atenderá pedidos de los clientes. Recogerá un pedido (cada vez) y comprobará que hay existencias en el almacén. Si es así (hay unidades de todos los productos pedidos) informará a los **EmpElaboraProductos** para que comiencen con la elaboración de los productos especiales (si los hay) y a los **EmpPreparaPedido** para que realicen su tarea (preparar el pedido y dárselo al cliente). Cuando el pedido ha sido puesto en el dispensador, avisará al cliente y procederá a cobrar el importe del pedido.
- **EmpElaboraProductos:** Este empleado estará a la espera de que le pidan que elabore un producto especial. Cuando reciba la petición, elaborará el producto pedido (retirando los productos del almacén) y lo dejará en una zona intermedia para que sean recogido por el **EmpPreparaPedido**.
- **EmpPreparaPedido:** Este empleado se encarga de recoger los productos del almacén que no necesitan una preparación especial y de esperar a que los **EmpElaboraProductos** elaboren los productos más complejos. Cuando el pedido está completo le entrega por un dispensador el pedido al cliente. Si no hay dispensadores libres, deberá esperar. Cuando haya dejado el pedido en el dispensador, avisará al **EmpCogePedidos** para que avise al cliente que su pedido está preparado y listo para recoger.
- **EmpEncargado:** Este empleado se encargará de abrir el almacén y la tienda por la mañana y de cerrarlos por la noche. Cuando la tienda está abierta al público, revisará a intervalos regulares el trabajo de los empleados y el comportamiento de los clientes. Entre otras cosas, revisará el número de dispensadores libres y el tamaño de la cola de clientes.

Para implementar el programa se ofrecen un conjunto de clases que forma la arquitectura de la aplicación. La clase principal es la clase **Personal**, que contiene el ciclo de vida de todos los procesos. Las clases auxiliares **Pedido**, **Producto**, **ProductoEspecial** se utilizan en la implementación de la clase **Personal**. La clase **Ukea** es la encargada de realizar todas las sincronizaciones y comunicaciones entre los hilos del programa que representan personas.

A continuación, se muestra el código fuente de **Personal**:

```
package ukea;

public class Personal {

    private static final int NUM_CLIENTES = 8;
    private static final int NUM_EMP_COGE_PEDIDOS = 4;
    private static final int NUM_EMP_ELABORA_PRODUCTOS = 3;
    private static final int NUM_EMP_PREPARA_PEDIDO = 3;
    private static final int NUM_DISPENSADORES = 2;

    private static Ukea ukea = new Ukea(NUM_DISPENSADORES);

    // ----- Métodos procesos -----

    public static void empCogePedidos() {
        ...
    }

    public static void empElaboraProductos() {
        ...
    }

    public static void empPreparaPedido() {
        ...
    }

    public static void cliente() {
        ...
    }

    public static void encargado() {

    }

    // ----- Main -----
    public static void main(String[] args) {
        //Creación de hilos que realizan llamadas a los métodos correspondientes
    }
}
```

## Se pide

Se pide implementar completamente en Java un programa concurrente que simule el funcionamiento de una tienda **Ukea**. Para ello se deberá implementar completamente la clase Ukea y Personal, así como el resto de clase auxiliares. Además, si lo estimas conveniente, el código que se proporciona con este enunciado podrá ser modificado para cumplir con ciertos requisitos del enunciado.

Para la implementación se deberán tener en cuenta las siguientes consideraciones:

- Cualquier decisión respecto al enunciado, no especificada en el mismo, debe explicarse adecuadamente en la memoria.
- Deberán utilizarse las clases de la API de Java más adecuadas para implementar la sincronización y comunicación de la clase Ukea. Queda excluido el uso de la biblioteca SimpleConcurrent (empleada en la primera parte de la asignatura).
- El número de empleados de cada tipo debe poder configurarse fácilmente. Por ejemplo, mediante una serie de constantes.
- Cuando el encargado notifica a sus empleados el fin de la jornada, podrá usar interrupciones.
- El número de pedido es un contador que se iniciará en 1 cada vez que se ejecute el programa y se incrementa por cada pedido.
- El horizonte temporal en el que llegan los pedidos y la velocidad de llegada de los pedidos debe poder configurarse también mediante algún parámetro.
- Se recomienda asignar un factor rectificación de tiempo, de modo que se pueda simular una ejecución de 24 horas en unos cuantos segundos.
- **Toda la información** de lo que está sucediendo debe mostrarse por pantalla. Por ejemplo, cuando un determinado cliente hace un pedido, se mostrará los datos del cliente y del pedido. También debe mostrarse cuando el pedido es recogido, cuando se está elaborando, cuando está depositado en un determinado dispensador, etc.
- **NOTA:** Para realizar la simulación, se supondrá lo siguiente:
  - Cada producto se identifica por un valor entero único y tiene un precio.
  - En el almacén hay 10 productos (identificados del 0 al 9) y 15 unidades de cada producto.
  - Los pedidos de cada cliente se generan de forma aleatoria. Como mucho, compra 5 productos en cada pedido. Los productos especiales se generan con un número de dos cifras, que significa que están compuestos por una unidad de cada producto asociado al identificador. Es decir, si el cliente pide un producto especial con el número 23, eso significa que está compuesto por un producto con id= 2 y con un producto con id=3.
  - Los clientes tienen 300 euros para gastar. Cuando no pueden pagar el pedido, se anula.
  - Si un pedido de un cliente no puede servirse (porque el cliente no pueda pagar o porque no haya suficientes unidades en el almacén) el producto se anula.