
PROGRAMACIÓN AVANZADA

Práctica de Laboratorio PEC3L - Parte 1

Convocatoria Ordinaria – Mayo 2019

Publicado el 18 de marzo de 2019

Simulación del funcionamiento de un supermercado

Se pretende simular el comportamiento de **un supermercado** que cuenta con un aforo de 20 compradores.

El comportamiento del sistema a tener en cuenta es:

- Los compradores, cada uno de los cuales se representará por un hilo, llegarán al supermercado (serán creados) con un tiempo aleatorio de entre 0,2 y 1 segundo. Cada comprador estará identificado por un identificador único, por ejemplo: “Comprador-1”, “Comprador-2”, etc.
- Los compradores, para entrar en el supermercado, una vez esté lleno, esperarán siguiendo una estrategia tipo FIFO, es decir, que irán entrando por orden de llegada.
- Una vez entra un comprador al supermercado, puede decidir (aleatoriamente) si ir a la carnicería, a la pescadería o a coger artículos de los estantes. Hay un carnicero y un pescadero (que deben ser modelados como hilos) encargados de atender a los compradores que lleguen a la carnicería y pescadería respectivamente. El carnicero tarda entre 1,5 y 2,5 segundos en atender a un comprador, y el pescadero tarda entre 2 y 3 segundos. El tiempo que tarda un comprador en coger artículos de los estantes varía entre 1 y 11 segundos.
- Como sólo hay un carnicero y un pescadero, sólo puede haber un comprador en la carnicería y en la pescadería (uno en cada uno), de tal forma que el resto de compradores deben esperar en una cola independiente para cada una de ellas, siguiendo una estrategia FIFO. En cambio, los 20 compradores pueden estar en los estantes a la vez sin ninguna restricción.
- Una vez que un comprador termina en los estantes, pescadería o carnicería, se dirige a la zona de cajas para pagar. En la zona de cajas hay una cola de espera común donde deben esperar los compradores su turno para pasar a pagar a una de las cajas. Hay dos cajas y, por tanto, sólo pueden ser atendidos dos compradores a la vez.
- Cada cajera tarda entre 3 y 5 segundos en cobrar a cada comprador. Una vez que el comprador ha pagado, abandona el supermercado y termina.

- Tanto el carnicero, como el pescadero y las cajas deben esperar si no hay ningún comprador al que atender.
- Se realizará una interfaz gráfica que muestre en todo momento el comportamiento de cada uno de los agentes involucrados. En dicha interfaz debe existir la opción de pausar y reanudar la simulación.

El comportamiento de los compradores se generará aleatoriamente mediante las funciones random de Java, y **todo el comportamiento del sistema se guardará en un log** (un fichero de texto llamado “evolucionSupermercado.txt”), **además de mostrarse gráficamente** por pantalla, de forma que sea sencillo analizar lo sucedido. El log guardará los eventos que van teniendo lugar, por ejemplo: “Comprador1 entra en el supermercado”, “Comprador2 espera en la carnicería”, “Pescadero atiende al Comprador3”, “Cajera1 cobra al Comprador2”, “Comprador5 abandona el supermercado”, etc. En cada línea de dicho log deberá constar la marca de tiempo (incluyendo el segundo determinado en el que tuvo lugar el evento) y el evento en sí.

El fichero de log será generado mediante un hilo separado de todos los demás, y recibirá los mensajes a imprimir mediante un buffer de capacidad ilimitada y no bloqueante. De esta forma, aunque la escritura en el fichero presente retrasos, no afectará al rendimiento del resto de los hilos.

El sistema creará, como mínimo, 20.000 compradores. La simulación finaliza una vez hayan terminado de comprar todos los compradores y hayan abandonado el supermercado. Esta situación también se grabará en el log.

Para seguir la evolución del sistema, será preciso que se imprima el estado en el log cada vez que ocurra un evento. La interfaz gráfica permitirá visualizar el estado del sistema que contendrá, como mínimo, los datos:

- Qué compradores están esperando a entrar al supermercado.
- Qué compradores están esperando para la carnicería.
- Qué compradores están esperando para la pescadería.
- Cuántos compradores están en los estantes.
- A qué comprador está atendiendo el carnicero.
- A qué comprador está atendiendo el pescadero.
- Qué compradores están esperando para pagar.
- A qué compradores está atendiendo cada cajera.

Además, antes de finalizar el programa, el log deberá recoger la siguiente información:

- El tiempo medio de servicio en la carnicería y en la pescadería.
- Los tiempos acumulados totales de servicio en carnicería y pescadería.
- El número total de personas que han entrado y que han salido del supermercado.
- El tiempo medio que han tardado los clientes que han salido del supermercado: desde que llegaron a la cola hasta que salieron del supermercado, y desde que entraron hasta que salieron.

Se deberá incluir un botón para finalizar el programa. Cuando se pulse dicho botón, se dejará de seguir creando compradores nuevos, y los compradores que esperaban a entrar en el supermercado, así como los que esperaban en las colas de la carnicería y pescadería, no deberán llegar a entrar.

Una posible interfaz del sistema sería la que se puede apreciar en la siguiente imagen:

The screenshot shows a window titled "Supermercado" with a light blue background. It contains several text labels and input fields for a simulation:

- Personas esperando para entrar al supermercado:** A text box containing "Comprador00057, Comprador00058, Comprador00059, Comprador00060, Comprador00061, Comprador00062, Comprador00063, Comprador00064".
- Esperando para la carnicería:** A text box containing "Comprador00038, Comprador00040, Comprador00041".
- Esperando para la pescadería:** A text box containing "Comprador00036, Comprador00037, Comprador00039, Comprador00042".
- Carnicero atendiendo a:** A text box containing "Comprador00035".
- Pescadero atendiendo a:** A text box containing "Comprador00036".
- Nº compradores en estantes:** A text box containing "6".
- Esperando en cola para cajas:** A text box containing "Comprador00034, Comprador00033, Comprador00030, Comprador00029, Comprador00031, Comprador00032".
- Cajera1 atendiendo a:** A text box containing "Comprador00027".
- Cajera2 atendiendo a:** A text box containing "Comprador00028".
- At the bottom, there are two buttons: "Pausar simulación" and "Finalizar ejecución".

Se podrán utilizar todos los mecanismos vistos en clase para resolver todos los problemas de comunicación y sincronización que se plantean en este enunciado. No obstante, se deben utilizar los mecanismos de sincronización y comunicación que resuelvan el problema **de la forma más eficiente y óptima** posible.

Condiciones de entrega

1. La práctica se realizará (*opcionalmente*) **por parejas** y deberá ser entregada antes de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos

archivos: la memoria de la práctica en formato PDF o DOC y el proyecto Netbeans completo, comprimido como ZIP o RAR. No se aceptarán trabajos enviados pasada la fecha límite de entrega.

2. Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.
3. **La memoria deberá incluir, como anexo, el código fuente del programa. Si esto no fuera así, la práctica no podrá ser aprobada.**
4. La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.
5. Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.
6. Para aprobar, se debe desarrollar la solución haciendo uso de buenas prácticas de programación. Por ejemplo, es necesario que todos los nombres de las clases comiencen por una letra mayúscula y todos los nombres de atributos y métodos comiencen por una letra minúscula; los atributos deberán ser privados, y sólo se podrá acceder a ellos mediante métodos getter y setter.
7. En la portada de la memoria deberán figurar los datos siguientes:
 - a. **Grado en Ingeniería** _____ (Informática o de Computadores)
 - b. **Curso 2018/2019 – Convocatoria Ordinaria**
 - c. **DNI – Apellidos, Nombre**
8. La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel; 2) diseño general del sistema y de las herramientas de sincronización utilizados; 3) las clases principales que intervienen con su descripción (atributos y métodos); 4) un diagrama de clases que muestren cómo están relacionadas; y 5) el código fuente, como anexo.
9. Dicha documentación, exceptuando el código, no deberá extenderse más de 20 páginas. La calidad de la documentación – presentación, estructura, contenido, redacción – será un elemento básico en la evaluación de la práctica.
10. Para la defensa de la práctica, si el profesor de laboratorio así lo estimara necesario, deberá presentarse una copia en papel de la memoria, impresa por las dos caras y grapada. Este documento podrá ser utilizado por el estudiante como base para responder a las cuestiones que se le planteen en el ejercicio escrito sobre la realización de la aplicación.
11. Para mostrar el funcionamiento de los programas, es conveniente que cada estudiante utilice su propio ordenador portátil, en previsión de posibles problemas al instalarlos en alguno de los ordenadores del laboratorio.