
PARADIGMAS DE PROGRAMACIÓN

Práctica de Laboratorio (PECL)

Convocatoria Ordinaria – Enero 2026

VI – Publicado el 04 de noviembre de 2025

Simulación del funcionamiento de una cafetería

Parte 1: Programación Concurrente

Se desea modelar el comportamiento una cafetería. Se tienen las siguientes características generales:

- El sistema deberá modelar el comportamiento de una cafetería.
- Se dispone de los siguientes actores: CLIENTES, VENDEDORES y COCINEROS.
- La cafetería cuenta con varias zonas de actividad para los actores: DESPENSA, COCINA, MOSTRADOR, CAJA, SALA DE DESCANSO, AREA DE CONSUMICIÓN y PARQUE.
- Los CLIENTES, VENDEDORES y COCINEROS deberán ser modelados obligatoriamente como hilos.
- Los elementos consumibles NO serán considerados hilos, aunque pueden modelarse con clases específicas, si se desea.
- El sistema deberá generar 8.000 CLIENTES. Para evitar congestión/saturación, los CLIENTES serán creados de forma escalonada, en intervalos aleatorios de entre 1 y 3 segundos.
- El número de VENDEDORES que el sistema genera es de 500. Para evitar congestión/saturación, los VENDEDORES serán creados de forma escalonada, en intervalos aleatorios de entre 0,5 y 2,5 segundos.
- El número de COCINEROS que el sistema genera es de 500. Para evitar congestión, los COCINEROS serán creados de forma escalonada, en intervalos aleatorios de entre 1 y 2 segundos.
- La generación de los diferentes actores (CLIENTES, VENDEDORES y COCINEROS) al comienzo de la ejecución del programa debe realizarse de forma concurrente, es decir, de forma simultánea.

CLIENTES

Los CLIENTES deben ser modelados como hilos y se identificarán como “C-XXXX”, donde XXXX es un número (id) único como, por ejemplo, C-0001, C-0023, C-1234, etc. Su función es ir a la cafetería, consumir e irse.

Cada uno de los CLIENTES tendrá el siguiente ciclo de vida:

- Inicialmente, tras generarse, acude al PARQUE. No hay límite de aforo.
- Contempla el parque durante un tiempo aleatorio entre 5 y 10 segundos.
- Se acerca a la cafetería. Dicho trayecto le llevará un tiempo aleatorio entre 3 y 9 segundos.
- Llega a la cafetería.
- Si hay aforo (hueco), podrá entrar a la cafetería. Si no, quedarán fuera esperando hasta que puedan entrar.
 - Entre la entrada de la CAFETERIA y el MOSTRADOR puede haber hasta un máximo de 20 personas.
- Los CLIENTES, una vez entran a la cafetería, formarán una cola ordenada para acceder al MOSTRADOR.
 - El aforo del MOSTRADOR es de 5 personas. En este aforo no influyen los VENDEDORES y/o COCINEROS.
- Accede al MOSTRADOR para seleccionar los productos. En el MOSTRADOR puede haber un máximo de 5 CLIENTES a la vez.
 - Pide aleatoriamente entre 1 y 3 cafés.
 - Pide aleatoriamente entre 0 y 4 rosquillas.
- Si no hay elementos disponibles, esperará a que los VENDEDORES vayan reponiendo. Una vez tenga todos los productos, accede a la CAJA para pagar por los productos elegidos.
 - El aforo de la CAJA es de 10 CLIENTES. Serán atendidos por orden de llegada. Se entiende que hay 10 máquinas de autopago para atender a los clientes.
 - Los cafés cuestan 1,50€.
 - Cada rosquilla cuesta 2,50€.
 - El monto total de ingresos de la cafetería deberá ser guardado como una variable compartida.
 - El proceso de pagar requiere un tiempo entre 2 y 5 segundos.
- Se dirige al AREA DE CONSUMICIÓN para tomar los productos adquiridos.
 - El aforo del AREA DE CONSUMICIÓN es de 30 personas. Se debe respetar el orden de llegada para entrar.
 - El proceso de consumición, independientemente del número de productos adquiridos, requiere un tiempo aleatorio entre 10 y 15 segundos.
- El CLIENTE una vez consumidos los productos, abandona la cafetería y finaliza su ejecución.

COCINEROS

Los COCINEROS deben ser modelados como hilos y se identificarán como “B-XXXX”, donde XXXX es un número (id) único como, por ejemplo, B-0001, B-0023, B-1234, etc. Se encargan generar unidades de los productos a comprar.

Los COCINEROS tendrán el siguiente ciclo de vida, que será **repetido infinitamente**:

- Inicialmente son generados en la SALA DE DESCANSO. No hay límite de aforo.
- Primero, descansan durante un tiempo aleatorio entre 5 y 10 segundos.
- Después se dirigen a la COCINA. El trayecto dura entre 1 y 3 segundos.
- Accede a la COCINA. El aforo de esta estancia es de 100 COCINEROS.
- Preparan los productos.
 - Generan aleatoriamente entre 2 y 5 unidades de café.
 - Generan aleatoriamente entre 4 y 8 unidades de rosquillas.
 - El proceso de generación, independientemente de las unidades generadas, lleva un tiempo de entre 5 y 10 segundos.
- Mueven los productos a la DESPENSA:
 - Dirigirse a la DESPENSA requiere un tiempo entre 2 y 5 segundos.
 - Moverá las unidades generadas para guardarlas en unas estanterías. Se supone una capacidad ilimitada de almacenamiento de unidades.
 - Hay 1 estantería para los cafés.
 - Hay 1 estantería para las rosquillas.
 - No es necesario dejar ambas categorías de elementos de una vez; se pueden dejar parcialmente. Por ejemplo, primero dejar los cafés y luego las rosquillas (o viceversa).
 - El aforo de la DESPENSA es de 50 COCINEROS y de 50 VENDEDORES.
- Posteriormente, se dirigen a la SALA DE DESCANSO. El trayecto requiere un tiempo entre 2 y 5 segundos.

VENDEDORES

Los VENDEDORES deben ser modelados como hilos y se identificarán como “V-XXXX”, donde XXXX es un número (id) único como, por ejemplo, V-0001, V-0023, V-1234, etc. Se encargan mover unidades de los productos a comprar.

Los VENDEDORES tendrán el siguiente ciclo de vida, que será **repetido infinitamente**:

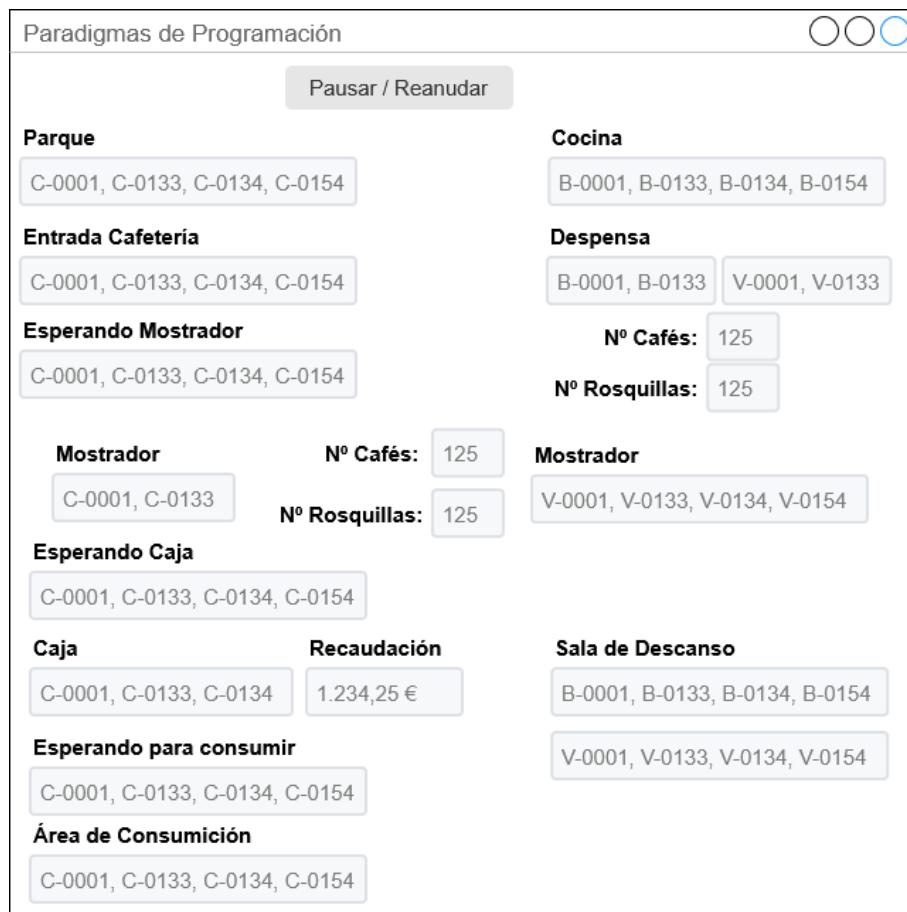
- Inicialmente son generados en la SALA DE DESCANSO. No hay límite de aforo.
- Descansan durante un tiempo aleatorio entre 5 y 10 segundos.
- Se dirigen a la DESPENSA. El trayecto dura entre 1 y 3 segundos.
- Entran a la DESPENSA.
 - El aforo de la DESPENSA es de 50 COCINEROS y de 50 VENDEDORES.
- Seleccionan los productos:
 - Selecciona entre 3 y 6 unidades de café.
 - Selecciona entre 5 y 10 unidades de rosquillas.
 - No es necesario coger todos los elementos de una vez; se pueden coger parcialmente.
 - Una vez recopilados los productos, independientemente de las unidades seleccionadas, se prepara en un tiempo entre 1 y 3 segundos.
- Mueven los productos al MOSTRADOR. El trayecto dura entre 2 y 5 segundos.
- Colocan los productos en el MOSTRADOR.
 - El aforo del MOSTRADOR, para VENDEDORES, es de 20 empleados.
 - Ubicará los productos en los estantes correspondientes. Se supone una capacidad de almacenamiento ilimitada de unidades.
 - Hay 1 estantería para los cafés.
 - Hay 1 estantería para las rosquillas.
 - El proceso de colocar los productos, independientemente de las unidades seleccionadas y los posibles tiempos de espera para acceder a variables compartidas, requiere un tiempo entre 1 y 3 segundos.
- Se dirige a la SALA DE DESCANSO. El trayecto requiere un tiempo entre 2 y 5 segundos.

Otras Consideraciones

Todo el comportamiento del sistema se guardará en un fichero de **log** (un fichero de texto llamado “evolucion_cafeteria.txt”), de forma que sea sencillo analizar lo sucedido. El log guardará todos los eventos que van teniendo lugar, por ejemplo: “*Cliente C-0025 es creado*”, “*Cocinero B-0105 genera 3 cafés y 4 rosquillas*”, “*Vendedor V-0235 repone 4 cafés y 7 rosquillas en el mostrador*”, etc. En cada línea de dicho log deberá constar la marca de tiempo (**fecha y hora**, incluyendo el segundo determinado en el que tuvo lugar el evento) **y el evento** en sí. El sistema de log deberá implementarse como un recurso compartido a utilizar por todo el sistema concurrente y deberá protegerse adecuadamente.

Todo el comportamiento del sistema se mostrará gráficamente por pantalla. Además, se deberá incluir un **botón para pausar/reanudar el sistema**, de forma que sea sencillo hacer el seguimiento de la ejecución del programa.

Un posible ejemplo de interfaz del sistema sería la que se puede apreciar en la siguiente imagen:



Nota importante: los valores indicados en los campos de las capturas de pantalla no tienen por qué ser correctos, ni estar completos. Las imágenes se muestran con el único fin de proporcionar una idea sobre la interfaz final y la información que debe contener.

Parte 2: Programación Distribuida

Basándose en la Parte 1 anterior, incluir un nuevo módulo de acceso remoto implementado mediante RMI o Sockets, que ofrecerá las siguientes operaciones:

- Botón para detener/reanudar la ejecución del programa principal.
- Número de clientes en el parque.
- Número de clientes en el mostrador.
- Número de clientes en la caja.
- Número de clientes en el área de consumición.
- Número de cocineros en la cocina.
- Número de cocineros en la despensa.
- Número de vendedores en la despensa.
- Número de vendedores en el mostrador.
- Número de vendedores y cocineros en la sala de descanso.
- Unidades de café en la despensa.
- Unidades de rosquillas en la despensa.
- Unidades de café en el mostrador.
- Unidades de rosquillas en el mostrador.
- Recaudación actual de la caja.

La consulta de estos datos debe actualizarse cada segundo y debe mostrarse de forma automática, es decir, sin que el usuario deba pulsar ningún botón ni interactuar de forma alguna para actualizar la información.

Un posible ejemplo de interfaz gráfica es el que se muestra en la siguiente figura.



Nota importante: los valores indicados en los campos de las capturas de pantalla no tienen por qué ser correctos, ni estar completos. Las imágenes se muestran con el único fin de proporcionar una idea sobre la interfaz final y la información que debe contener.

Condiciones de entrega

1. *Se deben desarrollar, en total, dos programas:*
 - a. *Un servidor, cuyo código base será el programa desarrollado en la Parte 1, ampliado con la funcionalidad correspondiente para dar soporte al módulo de programación distribuida.*
 - b. *Un programa cliente que permita realizar las operaciones anteriormente mencionadas para consultar el estado del sistema.*
2. *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.*
3. *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 (no utilizar extensión .rar). No se aceptarán trabajos enviados pasada la fecha límite de entrega.*
4. **No se aceptará ninguna práctica que no contenga un proyecto de NetBeans.** La utilización de cualquier otro IDE distinto supondrá la no aceptación de la práctica.
5. *La práctica puede ser realizada de forma individual o en pareja, siempre que los alumnos cuenten con el **mismo profesor de laboratorio**. Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos. En cualquier caso, todos los estudiantes deberán estar obligatoriamente inscritos en algún grupo de laboratorio.*
6. **La memoria deberá incluir, como anexo, el código fuente del programa.** *Si esto no fuera así, la práctica no podrá ser aprobada.*
7. *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.*
8. **Ambas partes (Parte 1 y Parte 2) de la práctica de laboratorio se deberán entregar juntas** (es decir, en un único proyecto y una única memoria), ya que la Parte 2 se construye sobre la Parte 1.
9. *Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo con las especificaciones indicadas en los enunciados.*

- 10.** 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.
- 11.** En la portada de la memoria deberán figurar los datos siguientes:
- a. *Grado en Ingeniería en Sistemas de Información*
 - b. *Curso 20xx/20xx – Convocatoria Ordinaria*
 - c. *DNI – Apellidos, Nombre*
- 12.** La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel donde se detallen los requisitos funcionales establecidos y todos aquellos aspectos relacionados con la funcionalidad de la práctica; 2) diseño general del sistema, donde se describan todas las herramientas de sincronización utilizadas; 3) una descripción detallada de todas las clases que intervienen en el programa, así como una descripción detallada de los atributos y métodos que las integran; 4) un diagrama de clases que muestren cómo las clases están relacionadas; y 5) el código fuente, como anexo.
- 13.** 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.
- 14.** 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 usarse el estudiante como base para responder a las cuestiones que se le planteen en el ejercicio escrito sobre la realización de la aplicación (no se permitirá la compartición de documentación durante la prueba).
- 15.** La resolución de la práctica debe ser genuina y realizada desde cero, es decir, no se podrá utilizar ningún tipo de código fuente de prácticas presentadas a esta convocatoria o en convocatorias anteriores, bien sean propias o de otros compañeros. En caso de detectarse esta situación, la práctica será evaluada con una calificación de 0 Suspenso. Además, se ejecutarán los procedimientos correspondientes para comunicar dicho comportamiento fraudulento, con consecuencias académicas acorde a la normativa vigente