

El Banco y la Atención de Clientes

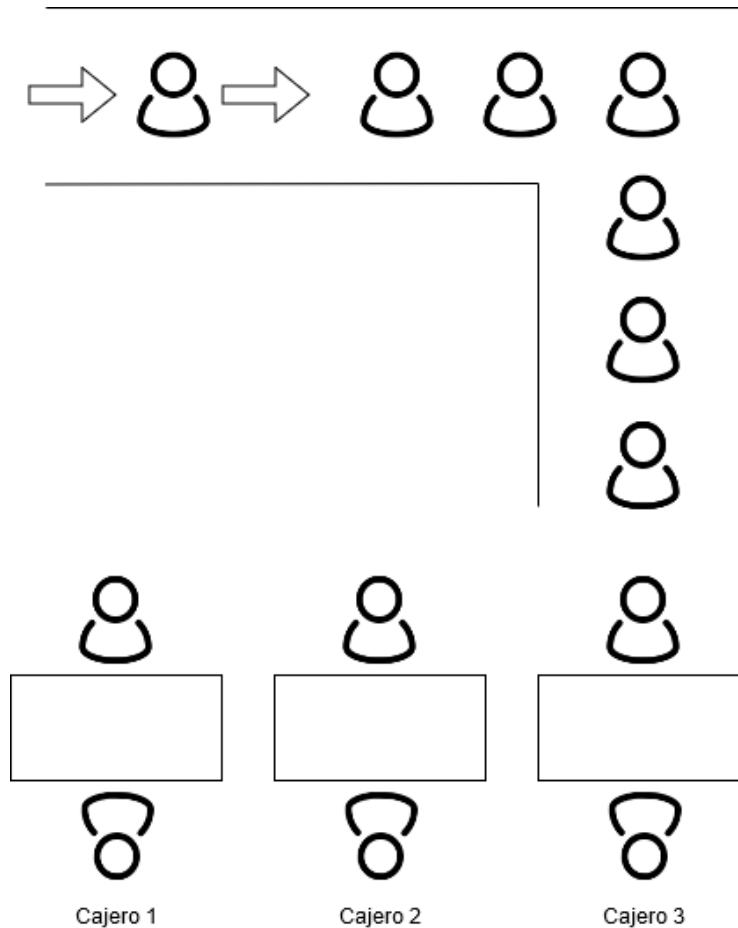
Contexto:

Usted es un reconocido desarrollador de soluciones tecnológicas, escucharon que en el pasado creó un gemelo digital para un supermercado; por lo tanto, un banco lo ha contratado para diseñar una simulación muy similar que modele la interacción entre los cajeros y los clientes. El objetivo final del banco es optimizar la atención al cliente mediante técnicas de simulación y análisis de rendimiento.

Desarrollo:

Usted deberá crear un programa en Java que simule los clientes y los cajeros usando concurrencia y sincronización.

A continuación, podrá visualizar una representación gráfica de la situación que debe simular.



*Imagen de referencia

Para la simulación debe considerar dos factores:

1. Los cajeros tienen un factor de cansancio acumulado, el cual representa que tanto ha trabajado un cajero, afectando su eficiencia.
2. Los clientes tienen un tiempo de procesamiento base, lo que representa cuanto se demora un cajero en atenderlo si el cajero no tiene cansancio acumulado.
3. Hay una única fila de clientes.
4. Los clientes se crean de forma aleatoria.

Estructura del programa:

Deberá crear cuatro archivos, Main.java, Fila.java, Clientes.java y Cajero.java

Main.java:

- Usaremos threads para implementar el comportamiento de los cajeros y el generador de clientes.
- Se encargará de crear los threads, el monitor e iniciarlos.
- Se deben crear cinco cajeros, una fila y un generador de clientes.

Fila.java

- Contendrá el arreglo con los clientes.
- Se encargará de manejar el acceso y actualización sobre el arreglo que contiene los clientes.

GeneradorClientes.java

- Se encargará de crear clientes y añadirlos a la fila. Entre la creación de clientes debe esperar un tiempo aleatorio.
- Cada cliente debe tener un tiempo de procesamiento aleatorio.
- Cada cliente creado debe tener un id único.

Cajero.java

- Deberá implementar todo el comportamiento que realizará el cajero al atender a los clientes.

Simulación:

El creador de clientes deberá tener los siguientes atributos:

- **uid:** Número entero que identificará a cada cliente, comienza en 1 y va aumentando a medida que se crean los clientes.
- **numeroDeClientes:** Numero entero que representa la cantidad de clientes que se van a añadir a la fila.
- **fila:** Objeto Fila, el cual contiene la compartida de clientes.

Sobre el comportamiento de clientes debe cumplir los siguientes requisitos:

1. **Tiempo de procesamiento básico:** Se genera un tiempo de procesamiento básico aleatorio para el cliente, debe ser un número entero entre 0 y 2000 ms.
2. **Creación de clientes:** Los clientes se representarán por medio de un elemento de dos posiciones. En la primera posición se almacena el uid del cliente y en la segunda se almacena el tiempo de procesamiento básico generado anteriormente. El cliente se adiciona a la fila compartida.
3. **Notificación de cliente nuevo:** Imprime un mensaje por cada cliente creado, incluyendo el id del cliente y el tiempo base.

Espera aleatoria entre creación de clientes: Se espera un tiempo aleatorio entre 0 y 500 ms para crear el próximo cliente con `Thread.sleep(tiempoAleatorio)`.

La fila deberá tener los siguientes atributos:

- **filaClientes:** Arreglo dinámico el cual contiene los clientes

Sobre el comportamiento de la fila debe cumplir los siguientes requisitos:

1. **Manejo de incorporación a la fila:** El método de adición debe implementar exclusión mutua para añadir el cliente. Además, debe notifica por si hay cajeros esperando para atender clientes.
2. **Manejo de retiro de la fila:** El método de retiro debe implementar exclusión mutua, Si hay clientes retira uno. Mantiene a los cajeros esperando si no hay clientes para atender.

El cajero deberá tener los siguientes atributos:

- **id:** id de cajero.
- **fila:** Objeto Fila, el cual contiene la referencia compartida a la fila de clientes.
- **factorDeCansancio:** Numero decimal que comienza en 1.0. Este valor va aumentando a medida que el cajero atiende clientes (abajo se indica cómo actualizar este valor).

Sobre el comportamiento del cajero debe cumplir los siguientes requisitos:

1. **Recepción del cliente:** Cuando se ejecuta el cajero, imprime un mensaje indicando que ha comenzado a procesar un cliente.
2. **Tiempo de procesamiento por cliente:** Para procesar a cada cliente se debe usar el tiempo calculado, el cual corresponde a la multiplicación del cansancio acumulado del cajero por el tiempo base de procesamiento del cliente en turno.

$$\text{tiempoCalculado} = \text{tiempo base} * \text{factorDeCansancio}$$

Finalmente se simula el procesamiento con `Thread.sleep(tiempoCalculado)`.

3. **Cálculo de cansancio acumulado:** Se toma el **0.01%** del tiempo base de procesamiento del cliente en turno y se suma al factor almacenado.

4. **Finalización de procesamiento:** Imprime un mensaje por cada cliente procesado, incluyendo el id del cajero, id del cliente, tiempo base, factor de cansancio y el tiempo correspondiente.
5. **Finalización de turno:** Al terminar de atender a todos los clientes el cajero imprime un mensaje indicando que ha terminado.

Ejemplos:

```
.
.
.
El cajero 1 atendiendo al cliente 4 (tiempo de procesamiento básico: 1977ms) factor de cansancio actual: 1.1273 tiempo de procesamiento actual: 2228ms
Cliente 5 agregado a la fila (tiempo de procesamiento: 2482ms)
El cajero 2 atendiendo al cliente 5 (tiempo de procesamiento básico: 2482ms) factor de cansancio actual: 1.1164 tiempo de procesamiento actual: 2770ms
Cliente 6 agregado a la fila (tiempo de procesamiento: 715ms)
El cajero 3 atendiendo al cliente 6 (tiempo de procesamiento básico: 715ms) factor de cansancio actual: 1.1334 tiempo de procesamiento actual: 810ms
Cliente 7 agregado a la fila (tiempo de procesamiento: 2034ms)
Cliente 8 agregado a la fila (tiempo de procesamiento: 1853ms)
El cajero 3 atendiendo al cliente 7 (tiempo de procesamiento básico: 2034ms) factor de cansancio actual: 1.204 tiempo de procesamiento actual: 2450ms
Cliente 9 agregado a la fila (tiempo de procesamiento: 782ms)
Cliente 10 agregado a la fila (tiempo de procesamiento: 1345ms)
Cliente 11 agregado a la fila (tiempo de procesamiento: 925ms)
Cliente 12 agregado a la fila (tiempo de procesamiento: 819ms)
El cajero 1 atendiendo al cliente 8 (tiempo de procesamiento básico: 1853ms) factor de cansancio actual: 1.325 tiempo de procesamiento actual: 2455ms
Cliente 13 agregado a la fila (tiempo de procesamiento: 2446ms)
Cliente 14 agregado a la fila (tiempo de procesamiento: 2200ms)
Cliente 15 agregado a la fila (tiempo de procesamiento: 821ms)
El cajero 2 atendiendo al cliente 9 (tiempo de procesamiento básico: 782ms) factor de cansancio actual: 1.3646 tiempo de procesamiento actual: 1067ms
Cliente 16 agregado a la fila (tiempo de procesamiento: 1968ms)
.
.
.
```

Condiciones de entrega:

- En un archivo .zip entregar el código fuente del programa. El nombre del archivo debe ser: laboratorio3_login1_login2_login3.zip
- Se **debe entregar por Bloque Neón** en el horario definido en la actividad “Laboratorio Taller 2”.
- Las fechas de entrega **no** serán movidas.
- Política de entrega tarde. Para las entregas tarde, se aplicará la siguiente política: por cada 30 minutos, o fracción, de retraso con respecto a la hora de entrega establecida en Bloque Neón habrá una penalización de 0,5/5.
- Está **prohibido** el uso de LLM, en caso de sospechar el uso de IA se le solicitará sustentar su código.
- La calificación de este laboratorio está condicionado a la asistencia, en caso de no asistir a clase o no firmar la hoja de asistencia **su nota será 0**.

HISTORIAL DE REVISIONES

| Fecha | Autor | Versión | Observación |
|------------|----------------------------|---------|---|
| 29/01/2025 | Juan Diego Lugo Sánchez | 1.0 | Creación de enunciado, verificación y versión inicial |
| 29/01/2025 | Sandra Rueda | 1.1 | Correcciones y aclaraciones del enunciado |