

# Programación Concurrente 2025

## Trabajo Práctico N° 1

### Condiciones

- El trabajo es grupal, de 4-5 alumnos (3 o 6 es una excepción)
- La defensa del trabajo es con el grupo completo
- La evaluación es individual (hay una calificación particular para cada integrante)
- Solo se corrigen los trabajos que hayan sido subidos al aula virtual (LEV)
- Los problemas de concurrencia deben estar correctamente resueltos y explicados.
- El trabajo debe implementarse en lenguaje Java.
- Se evaluará la utilización de objetos y colecciones, como así también la explicación de los conceptos relacionados a la programación concurrente.

### Enunciado

Se debe desarrollar un sistema concurrente para la gestión de entregas en una empresa de logística que realiza envíos de productos comprados en una plataforma de e-commerce. El sistema consta de varios procesos que deben ejecutarse simultáneamente y que respetan las siguientes reglas de funcionamiento.

El sistema mantiene una matriz que representa las posiciones de los casilleros en los centros de almacenamiento, de donde salen los pedidos. Cada casillero puede estar en estado: **vacío**, **ocupado**, o **fuera de servicio**. Además, cada casillero posee un contador que aumenta cada vez que se ocupa.

Adicionalmente, el sistema mantiene los siguientes registros:

- Pedidos en preparación
- Pedidos en tránsito
- Pedidos entregados
- Pedidos fallidos

El funcionamiento del sistema posee cuatro etapas:

**Preparación de Pedido:** Este proceso se encarga de recibir los pedidos de los usuarios. Se tienen tres hilos que ejecutan este proceso. Cada hilo intenta seleccionar un casillero aleatorio en la matriz, verificando que esté disponible. Si el casillero no está vacío, el hilo debe buscar otro casillero que sí lo esté. Una vez ocupado el casillero, el mismo se marca como ocupado y se registra el pedido en el registro de **pedidos en preparación**.

**Despacho de Pedido:** Este proceso es ejecutado por dos hilos, y se encarga de despachar los pedidos del listado de **pedidos en preparación**. Cada hilo toma un pedido aleatorio del registro de **pedidos en preparación** y realiza una verificación de los datos del pedido y del

usuario. Se establece una probabilidad del 85% de que la información sea correcta y un 15% de que sea incorrecta. Si la información fue correcta, el casillero vuelve al estado vacío, y el pedido se elimina del registro de **pedidos en preparación** y se agrega al registro de **pedidos en tránsito**. De lo contrario, el casillero pasa a estado fuera de servicio y el pedido se marca como fallido, se elimina del registro de **pedidos en preparación** y se agrega al registro de **pedidos fallidos**.

**Entrega al Cliente:** Tres hilos se encargan de ejecutar este paso. Cada hilo selecciona un pedido aleatorio del registro de **pedidos en tránsito** y con una probabilidad del 90%, lo confirma. Si el pedido es confirmado, se elimina del registro de **pedidos en tránsito** y se agrega al registro de **pedidos entregados**. Si el pedido no es confirmado, se elimina del registro de **pedidos en tránsito** y se agrega al registro de **pedidos fallidos**.

**Verificación Final:** Al finalizar la ejecución, se debe verificar el estado final de los pedidos para asegurar que las operaciones se hayan realizado correctamente. Este proceso selecciona de manera aleatoria un pedido del registro de **pedidos entregados**, y con una probabilidad del 95%, el pedido es verificado. Si el pedido fue verificado, se debe eliminar del registro de **pedidos entregados** y se debe insertar en el registro de **pedidos verificados**. En caso contrario, se elimina del registro de **pedidos entregados** y se inserta en el registro de **pedidos fallidos**. Este proceso es ejecutado por dos hilos.

**Consideraciones:**

- ☐ Cada proceso tiene una demora fija por iteración, pero distinta entre procesos, a ser definida por el equipo.
- ☐ Cada pedido debe ser accesible por un solo hilo a la vez para evitar conflictos de asignación simultánea.
- ☐ Cada pedido debe ser revisado por un solo hilo a la vez (independientemente del proceso).
- ☐ Cada pedido puede ser procesado una sola vez.
- ☐ Los procesos de preparación, despacho, entrega y verificación deben ejecutarse de forma concurrente para simular un entorno de pedidos realista.
- ☐ Los tiempos de espera para realizar cada operación deben ser aleatorios y configurables por el grupo.
- ☐ Al iniciar el programa, todos los hilos deben ser lanzados para que comiencen su ejecución.

El sistema debe contar con un LOG con fines estadísticos, el cual registre cada 200 milisegundos en un archivo:

- Cantidad de pedidos fallidos.
- Cantidad de pedidos verificados.

Además, al finalizar, el log debe imprimir una estadística de los casilleros y el tiempo total que demoró el programa.

## Ejercicios

- 1) Hacer un diagrama de clases que modele el sistema de datos con TODOS los actores y partes.
- 2) Hacer un diagrama de secuencias que modele las interacciones del sistema.
- 3) El sistema tiene una matriz con 200 casilleros.
- 4) El sistema debe procesar 500 pedidos.
- 5) Se deben configurar los tiempos de los procesos de modo tal que el programa demore entre 15 y 30 segundos.
- 6) Hacer un análisis analítico detallado de los tiempos que el programa demora. Luego contrastarlo con múltiples ejecuciones obteniendo las conclusiones pertinentes.
- 7) El grupo debe poder explicar los motivos de los resultados obtenidos. Y los tiempos del sistema.
- 8) Debe haber una clase Main que al correrla, inicie los hilos.

## Entregar:a

- a) 1 archivo de imagen con el diagrama de clases en buena calidad.
- b) 1 archivo de imagen con el diagrama de secuencias en buena calidad.
- c) El código fuente Java (proyecto) de la resolución del ejercicio. El proyecto debe incluir librerías y extensiones necesarias, y debe poder correr en cualquier máquina independientemente del sistema operativo o IDE utilizada.
- d) 1 log y la explicación de los resultados.
- e) **Un informe obligatorio**, en formato pdf, con estructura formal (carátula, integrantes, desarrollo, etc.), donde se detalle:
  - i) Todo el trabajo realizado.
  - ii) Decisiones de diseño tomadas y su justificación (tiempos, etc)
  - iii) Las conclusiones obtenidas en base a los resultados
  - iv) Pueden incluir cualquier otra explicación que crean pertinente para el trabajo

Subir al LEV el trabajo **TODOS** los participantes del grupo.

## Fecha de entrega

Hasta Lunes 5 de Mayo de 2025 - 23:59 hs.

## Fecha de defensa

A coordinar por grupo desde el Martes 6 de Mayo de 2025.