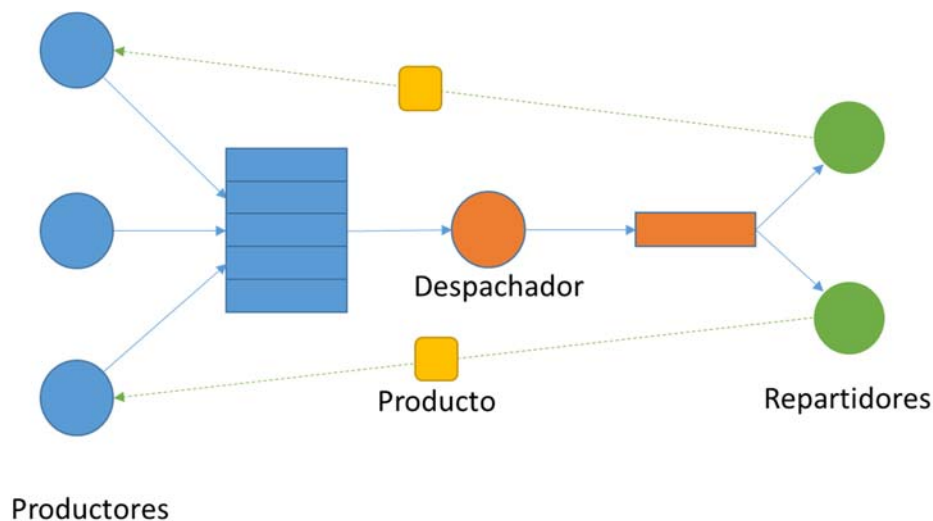


Caso 1 Manejo de la Concurrency

En este caso vamos a simular el comportamiento de una planta de producción y despacho con N threads productores, M threads repartidores y un thread despachador. Los productores generan productos (objetos) y los almacenan en una bodega de capacidad TAM productos de donde los toma el despachador para entregarlos a los repartidores. Los valores N , M y TAM los especifica el usuario (por consola) así como el número total de productos que deben ser producidos en cada ejecución del programa.

El proyecto debe ser realizado en java, usando *threads*. Para la sincronización solo pueden usar directamente las funcionalidades de Java vistas en clase: `synchronized`, `wait`, `notify`, `notifyAll`, `sleep`, `yield`, `join` y las `CyclicBarrier`.

A continuación, se presenta un ejemplo con $N=3$, $M=2$ y $TAM=5$.



Funcionamiento

Los productores generan productos y los ponen en la bodega. Cuando hay algo en la bodega, el despachador toma un producto y se lo entrega a algún repartidor disponible. Si la bodega está llena, los productores esperan a que haya espacio libre (espera pasiva) y si la bodega está vacía, el despachador realiza alguna otra tarea (un método cualquiera no muy largo) antes de volver a revisar la bodega de nuevo (espera activa, pero útil). Un productor debe esperar (pasivamente) a que su producto sea entregado por algún repartidor antes de producir uno nuevo. Tenga en cuenta que, si no hay espacio el productor se duerme sobre la bodega, pero si deposita se duerme sobre el producto.

Los repartidores esperan que el despachador les entregue un producto para ir a entregarlo. Si el despachador no tiene productos para entregar, esperan de manera pasiva a que sea su turno. El tiempo que demora un repartidor en entregar un producto es un valor aleatorio en 3 y 10 segundos.

Cuando el despachador tiene un producto, intenta pasarlo a un repartidor. El despachador no puede pasar un nuevo producto hasta tanto el anterior no haya sido recogido para reparto por algún repartidor. En ese caso el Despachador hace una espera pasiva.

Cada vez que un repartidor entrega un producto, el productor que estaba esperando porque su producto fuera entregado puede continuar a producir un nuevo producto.

El programa debe generar mensajes suficientemente claros para evidenciar el buen funcionamiento de este. En particular debe ser claro las diferentes etapas por donde pasa un producto (producido, en bodega, en despacho, despachado, en reparto, entregado). Recuerde que su programa será analizado por un tercero que no lo conoce y debe entender fácilmente los resultados.

Los productores conocen el número de productos a producir (cada uno), el despachador sabe cuántos productores hay, cuántos productos produce cada uno y el número de repartidores. Los repartidores no conocen ninguno de estos valores, por lo que el despachador debe idear una marca para que los repartidores terminen. Al final del programa todos los threads deben haber terminado su ejecución.

Condiciones de entrega

- En un archivo .zip entregar el código fuente del programa, y un documento Word explicando el diseño (diagrama de clase) y funcionamiento del programa, así como la validación realizada. En particular, para cada pareja de objetos que interactúan (para cada pareja thread-thread o thread-objeto), explique cómo se realiza la sincronización, así como el funcionamiento global del sistema. El nombre del archivo debe ser: `caso1_login1_login2_login3.zip`
- El trabajo se realiza en los grupos definidos en el curso para este caso. No debe haber consultas entre grupos.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes).
- Habrá una coevaluación del trabajo en grupo. Cada miembro del grupo evaluará a sus dos compañeros y las notas recibidas por un estudiante ponderarán la nota final de caso para ese estudiante.
- En el parcial se incluirá una pregunta sobre el desarrollo de alguna de las funcionalidades del caso. La nota obtenida en esa pregunta puede afectar la nota de todos los miembros del grupo.
- El proyecto debe ser entregado por Bloque Neón por uno solo de los integrantes del grupo. **Al comienzo del documento Word, deben estar los nombres y carnés de los integrantes del grupo.** Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente, sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- **Se debe entregar por BNe a más tardar el 11 de septiembre a las 23:55 (p.m.)**

RÚBRICA:

Tengan en cuenta que el código debe correr en los casos de prueba definidos para la sustentación. El funcionamiento y calidad del programa entregado pondera la nota de cada ítem: p.e. no compila; compila, pero no corre; corre, pero no termina, etc.)

- Entradas y salidas claras del programa: 10%
- Arquitectura de operación: 15%
- Funcionamiento de la bodega: 20%
- Funcionamiento del despachador: 20%
- Sincronización entre repartidores y productores: 20%
- Informe: 15%

Cronograma Propuesto:

25 agosto: Lectura del enunciado y entrega del contrato
28 agosto: Arquitectura general de la solución (diagrama de clases)
31 agosto: Estrategía de solución para los procesos
7 septiembre: Implementación + pruebas
11 septiembre: Informe y entrega
13 septiembre: Realizar Coevaluación