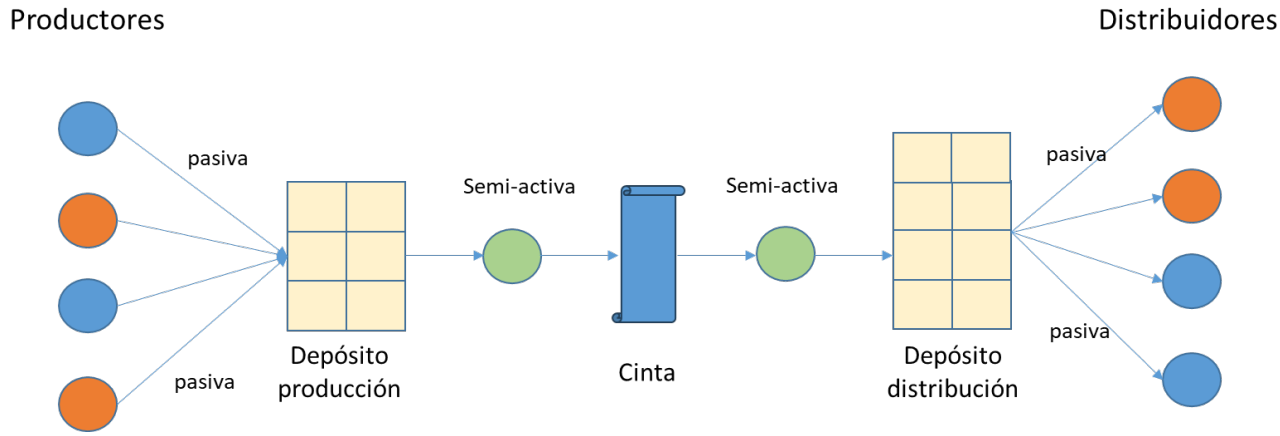
Universidad de los Andes
Departamento de Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional

Caso 1
Manejo de la Concurrencia

Queremos implementar una arquitectura de productores y consumidores en una fábrica, como la que muestra la figura:



- En la fábrica los operarios productores producen dos tipos de productos ("A" y "B") y los operarios distribuidores distribuyen esos productos. Tanto los productores como los distribuidores son especializados, es decir, solo trabajan con un tipo de producto.
- Los productores almacenan los productos en un depósito de producción. De ahí son llevados a una cinta transportadora y de la cinta pasan a un depósito de distribución de donde son retirados de **una cola única** por los distribuidores.
- Hay 4 operarios productores (2 tipo "A", 2 tipo "B"), 4 operarios distribuidores (2 tipo "A", 2 tipo "B") y dos operarios internos: uno para mover productos del depósito de producción a la cinta y otro para mover de la cinta al depósito de distribución.
- El depósito de producción puede almacenar hasta `capDepProd` productos y en el depósito de distribución caben hasta `capDepDist`. En la cinta transportadora solo cabe 1 producto a la vez.

Objetivo

Diseñar un mecanismo de comunicación para implementar la arquitectura descrita. Para este caso, todos los operarios serán *threads* en la misma máquina (en realidad debería ser un sistema distribuido; este es solo un prototipo).

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

Funcionamiento

- Los operarios azules son especializados en productos tipo “A” mientras que los naranjas son especializados en productos tipo “B”.
- Los productores producen un producto a la vez: solo pueden producir un nuevo producto una vez han almacenado el anterior. Los productos son almacenados en el depósito de producción. Si este se encuentra lleno, deben esperar a que haya espacio (espera pasiva). buffer
- Los distribuidores retiran del depósito de distribución los productos a distribuir. Un distribuidor busca productos del tipo que él distribuye: si el depósito está vacío o no tiene productos de ese tipo, debe esperar a que haya un producto de su tipo para distribuir (espera pasiva).
- Cada productor produce `numProductos` productos en total. Una vez ha producido todos sus productos, genera un producto vacío para indicar que ya no producirá más y termina su ejecución. Los productores que producen productos tipo “A” al final generan un producto tipo “FIN_A”, mientras que los productores que producen productos tipo “B” al final generan un producto tipo “FIN_B”.
- Los distribuidores no conocen el número de productos totales a consumir, ellos consumen indefinidamente hasta que encuentran un producto tipo “FIN_A” o “FIN_B” según sea el tipo de productos que consumen. En ese momento dejan de distribuir y terminan su ejecución. Note que, al ser el mismo número de productores y distribuidores para cada tipo de producto, cada distribuidor recibirá un producto de terminación para garantizar su terminación.
- Los 2 operarios internos mueven productos hacia o desde la cinta transportadora. Como se espera que estos operarios estén siempre ocupados, todas sus operaciones de retiro o almacenamiento son hechas con espera semi-activa (yield). Para que un operario interno termine, debe haber visto pasar 4 productos marcados de fin (2 “FIN_A” y 2 “FIN_B”).

El número de productos (`numProductos`) y la capacidad de los depósitos (`capDepProd` y `capDepDist`) deben ser leídos por consola.

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, 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 el 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
- El proyecto debe ser entregado por BloqueNeón 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.
- Tenga en cuenta las reglas establecidas en el programa del curso sobre la obligatoriedad del trabajo en grupo.
- **Se debe entregar por BoloqueNeón a más tardar el lunes 9 de septiembre a las 23:55 (p.m.)**

Cronograma Propuesto:

22 ago.:	Lectura del enunciado
24 ago.:	Entrega del contrato de trabajo en grupo
28 ago.:	Arquitectura general de la solución (diagrama de clases) y estrategia para los procesos
05 sept.:	Implementación + pruebas
09 sept.:	Informe y entrega
10 sept.:	Realizar Coevaluación