

## Infraestructura Computacional

### Caso #1 — Manejo de la concurrencia

**Camilo García Villabona** — 201821149  
**Nicolás Ortega Caicedo** — 201814515

#### Funcionamiento global del sistema

En el programa existen 5 entidades diferentes que interactúan entre sí. Estas son Productor, Buzón, Consumidor, Intermediario y Producto. Cada Productor crea una cantidad (determinada en un archivo de configuración) de Productos de tipo “A” y de tipo “B”, y cada Consumidor “consume” (recibe) la misma cantidad de Productos. Estos productos son transportados a través de los Buzones y los Intermediarios. El funcionamiento general para cada entidad se da así:

- Existen 3 Buzones en total. Uno en el que se almacenan los productos que llegan de los Productores, otro que almacena solo un producto entre los intermediarios, y otro que almacena los productos que reciben y consumirán los Consumidores.
- Los Productores inician y constantemente crean e intentan insertar Productos en su buzón correspondiente. Si este se encuentra lleno esperan semi-activamente hasta que haya un espacio disponible para depositar. Cuando ya han depositado la cantidad de productos definida en la configuración, terminan.
- Existen 2 Intermediarios, cada uno inicia y constantemente intenta sacar un producto del buzón a “izquierda” y lo depositarlo en buzón de su “derecha”. El primer intermediario intenta sacar productos del buzón de Productores y depositarlos en el buzón intermedio. Si no hay ningún producto para sacar, espera pasivamente hasta que un Productor le notifique que ya depositó un producto. De manera similar, el segundo intermediario intenta sacar productos del buzón intermedio y depositarlos en el buzón de Consumidores. Si no hay ningún producto para sacar, espera pasivamente hasta que el primer intermediario le notifique que ya depositó un producto. También, en ambos casos, si no hay espacio en el buzón en el que depositan sus productos, esperan pasivamente hasta que se les notifique que se retiró algún producto del buzón. Cuando cada uno ha pasado la cantidad total de productos del sistema de un buzón a otro, terminan.
- Los Consumidores inician y constantemente intentan sacar Productos de su buzón correspondiente. Si este se encuentra vacío, o no hay productos del tipo que deben consumir, esperan semi-activamente hasta que haya un producto disponible (que sea del tipo que consumen) para retirar. Cuando ya han sacado la cantidad de productos definida en la configuración, terminan.

#### Sincronización productor – intermediario 1

La sincronización entre el productor y el intermediario 1 se da a través del buzón de Productores. Como se mencionó anteriormente, en la ejecución del programa los productores se encuentran intentando insertar productos en su buzón constantemente, y, al mismo tiempo, el intermediario 1 se encuentra intentando sacar un producto del mismo

buzón. Este buzón entonces actúa como monitor para la sincronización de esta parte del “transporte” de productos. Cuando el intermediario 1 intenta retirar un producto del buzón, pero este se encuentra vacío, inicia espera pasiva a través de la línea *wait()*, que se encuentra en bloque sincronizado de código. Cuando los Productores intentan insertar un producto en el buzón, pero este se encuentra lleno, inician espera semi-activa a través de la línea *Thread.yield()*. Una vez los Productores hayan logrado depositar un producto en el buzón, ejecutan la línea *notify()*, que también está en bloque sincronizado, para que si el intermediario 1 se encuentra esperando a que haya un producto, ya pueda seguir su ejecución y retirarlo del buzón.

### **Sincronización intermediario 1 – intermediario 2**

La sincronización entre intermediarios se da a través de esperas pasivas, como se pidió en el enunciado. Como se tienen 2 intermediarios que comparten un buzón (el buzón intermedio), el funcionamiento es tal que el intermediario uno (prodint, en términos de nuestra implementación) se encarga de sacar un producto del buzón de los productores a través del método y dejarlo en el buzón intermedio.

Tras esto, este intermediario hace un *notify()* de tal forma que puede despertar al intermediario 2 si este se encuentra en espera pasiva para que tome el producto depositado en el buzón intermedio y proceda a depositarlo en el buzón de consumidores.

De manera análoga, una vez el intermediario 2 realiza el proceso explicado anteriormente, se encarga de, usando un *notify()*, despertar al intermediario 1 para que intente obtener un producto del buzón de productores y lo lleve al buzón intermedio nuevamente con el fin de repetir el proceso hasta terminar el procesamiento de todos los productos.

Es importante mencionar que se utiliza *notify()* y no *notifyAll()* pues tanto el intermediario 1 como el intermediario 2 solo deben avisar a un único thread, el otro.

### **Sincronización intermediario2 – consumidores**

Para esta sincronización es importante entender 2 partes, la primera corresponde a la capacidad de los intermediarios de dejar los productos en el buzón de consumidores y la segunda a la habilidad de los consumidores para, precisamente, consumir los productos dentro de este buzón. Es importante hacer hincapié en esto pues, cualquiera falla en estos dos escenarios llevaría a fallas en el funcionamiento global de la sincronización de esta parte.

Adicionalmente, es importante mencionar que se necesitó hacer la implementación de un objeto llamado *prod*, con el fin de utilizar su monitor para garantizar el correcto funcionamiento de la sincronización en todos los escenarios posibles para esta parte del sistema.

Para que se entienda mejor el proceso, es mejor empezar por la segunda parte, es decir, el consumo de los productos por parte de los consumidores. Esto se da de tal forma que, al intentar consumir un producto, si el consumidor encuentra que no hay productos por consumir en su buzón o que los productos ahí almacenados no corresponden al tipo que el consume, empieza una espera semi-activa a través del método *Thread.yield()*. Una vez se dan las condiciones correctas para que este consumidor pueda consumir un producto, se

procede a sacar ese producto del buzón y, posteriormente, a notificar al intermediario que se sacó un producto, con el fin que lo remplace, si es posible.

Por otro lado, de manera casi análoga a como sucedía en la sincronización entre intermediarios, el intermediario 2 se encarga de dejar en el buzón de consumidores los productos (siempre y cuando haya espacio) y, en caso de no poder hacerlo, espera pasivamente hasta que se le indique que lo pueda hacer.

Debido a la naturaleza de esta operación, se requirió hacer sincronización sobre la lista almacenados que contiene los productos contenidos en el buzón de productores y, además, para asegurar la terminación del proceso, sincronizar también sobre el objeto prod, aprovechando su monitor.

## Modelo conceptual

