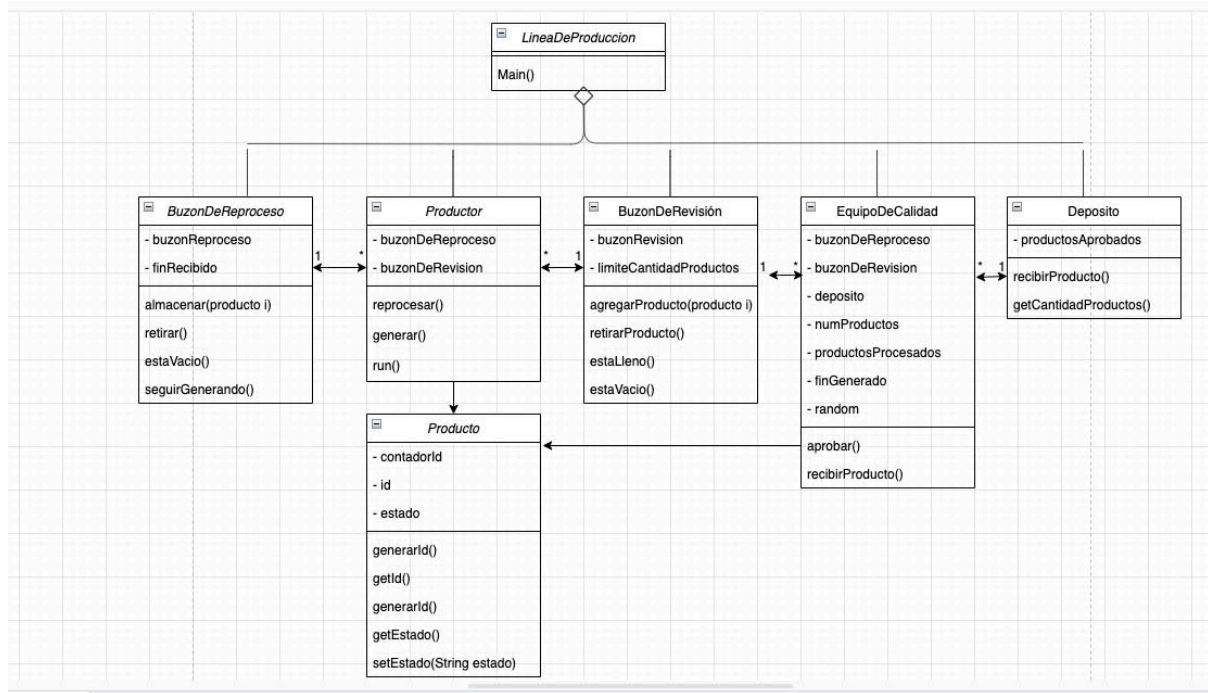


Documentación Caso 1 Infraestructura Computacional

Integrantes:

- Samuel Ayala Ayala s.ayalaa@uniandes.edu.co 202226890
- Lucas Valbuena Leon l.valbuena1@uniandes.edu.co 202311538

Explicación diseño del diseño y funcionamiento del programa



El programa implementa una arquitectura concurrente basada en múltiples hilos (threads) que representan a los operarios (productores y equipo de calidad) dentro de la línea de producción. Para evitar concurrencia y garantizar un acceso seguro a los recursos compartidos, se utilizan los mecanismos de sincronización de Java: `synchronized`, `wait()`, `notify()`, y `notifyAll()`.

En la línea de producción, los productores son responsables de generar nuevos productos o de reprocesar aquellos que han sido rechazados en inspecciones previas. Sin embargo, el sistema prioriza el reprocesamiento de productos defectuosos antes de permitir la generación de nuevos productos. Para garantizar esta condición, cada productor verifica primero el `buzonDeReproceso` antes de iniciar la generación de productos. Solo cuando este buzón está vacío, los productores pueden generar productos nuevos y depositarlos en el `buzonDeRevision`.

Manejo de la sincronización:

El equipo de calidad utiliza `synchronized (buzonDeRevision)` al retirar un producto, garantizando que solo un operario acceda a la vez y deteniéndose con `wait()` si el buzón está vacío. En el buzón de reproceso, cuando un operario deposita un producto rechazado, usa `synchronized (buzonDeReproceso)`, notificando (`notifyAll()`) a los productores para que puedan reprocesar. Los productores, a su vez, utilizan `synchronized (buzonDeReproceso)` al retirar productos defectuosos (múltiplos de 7), asegurando que solo uno acceda a la vez y deteniéndose con `wait()` si el buzón está vacío.

Métodos para asegurar el stop a la ejecución del programa:

Para asegurarnos de que los productores no sigan generando más productos de los que deben y el programa pare en algún momento, se le avisa a través del método `seguirGenerando()` en buzón de reproceso, este método maneja un booleano que cambia de valor cuando se lee "FIN" mientras se está almacenando. El mensaje "FIN" se genera cuando equipo de calidad aprueba el número de productos que se quería procesar. Finalmente, el productor sólo puede generar más proyectos si `seguirGenerando()` está en `True`.

El diseño basado en buzones permite una gestión fluida del flujo de productos, asegurando que cada operario realice su tarea en el momento adecuado sin generar cuellos de botella. La implementación de un control estricto sobre la generación y el reprocesamiento de productos optimiza el rendimiento del sistema, mientras que el mecanismo de detención basado en "FIN" permite una finalización ordenada de la ejecución.

Validaciones

Se realizaron múltiples ejecuciones del programa bajo distintas configuraciones de tamaño del buzón de revisión, número de productores y número total de productos a producir. Los principales escenarios evaluados fueron:

1. Tamaño del buzón de revisión: 2
 - Número de productores: 3
 - Total de productos a producir: 2

Resultados Observados

1. Los productores generaron los productos correctamente.
 - Se generaron exactamente 2 productos y fueron enviados al buzón de revisión.
 - Como el tamaño del buzón era 2, los productores tuvieron que esperar espacio disponible.
2. El equipo de calidad procesó los productos en orden.
 - Retiró productos del buzón y los aprobó o rechazó según correspondía.
 - Los productos aprobados fueron almacenados en el depósito.
3. El sistema generó el producto FIN correctamente.
 - Cuando los productos aprobados en el depósito alcanzaron el total esperado (2), se generó el producto FIN.
 - Los productores detectaron el FIN y finalizaron correctamente.
4. Finalización exitosa del proceso.
 - Todos los hilos terminaron sin generar productos adicionales.
 - No hubo bloqueos ni productos en espera indefinidamente.

El sistema funcionó correctamente con un buzón de revisión pequeño (2), mostrando sincronización efectiva entre los hilos y un correcto manejo de la condición de finalización.

2. Tamaño del buzón de revisión: 10

- Número de productores: 3
- Total de productos a producir: 9

Resultados Observados

1. Se generaron los 9 productos esperados.
 - El buzón de revisión permitió almacenar más productos sin que los productores se detuvieran con frecuencia.
 - No hubo congestión inmediata en la producción.
2. El equipo de calidad retiró y procesó los productos correctamente.
 - Se observaron productos procesados y almacenados en el depósito sin retrasos significativos.
 - Sin embargo, el conteo de productos procesados mostró una discrepancia temporal debido a la concurrencia de los hilos.
3. El producto FIN se generó correctamente.
 - Cuando los productos aprobados en el depósito alcanzaron los 9 esperados, el sistema generó el FIN.
 - Los productores y el equipo de calidad detectaron FIN y terminaron.

El sistema manejó correctamente un buzón de revisión más grande (10), permitiendo una producción más fluida y reduciendo bloqueos innecesarios. La única discrepancia encontrada fue un conteo desfasado en los productos procesados, que fue corregida asegurando una mejor sincronización entre hilos.

Conclusiones

- La ejecución del programa con los mismos parámetros puede producir resultados distintos debido a la concurrencia y el orden de ejecución de los hilos.
- Un buzón de revisión pequeño genera congestión y espera en los productores, mientras que un buzón más grande agiliza el flujo, pero puede retrasar la revisión de productos.
- La implementación del buzón de reproceso mejora la eficiencia del sistema al permitir la recuperación de productos defectuosos.
- Para optimizar el rendimiento del sistema, es recomendable encontrar un equilibrio entre el tamaño del buzón, el número de productores y la capacidad del equipo de calidad.