Texto

Descripción generada automáticamente

Alejandra Bravo Vega – 201613222

Nicolás Rincón Sánchez – 202021963

**ISIS2203 – INFRAESTRCUTURA COMPUTACIONAL**

**Caso #1 – Manejo de la Concurrencia**

## Entradas y salidas del programa

Como entrada del programa, se definen cuatro entradas por consola:

* N = número de productores
* M = número de repartidores
* TAM = tamaño de la bodega
* Cantidad total de productos

La salida del programa se compone de una serie de impresiones en la consola del estilo:

*Entidad <id>: Información sobre condición. Acción realizada por la entidad.*

*Producto <id>: Estado del producto*

## Arquitectura de la solución

Se definió el siguiente diagrama de clases para modelar el mundo del problema. En particular, se definieron cinco entidades principales (Productor, Despachador, Repartidor, Bodega y Producto) que interactúan entre sí. Las primeras tres clases se modelan como Threads de Java por lo cual heredan de esta superclase. Representan los agentes principales de la planta de producción.

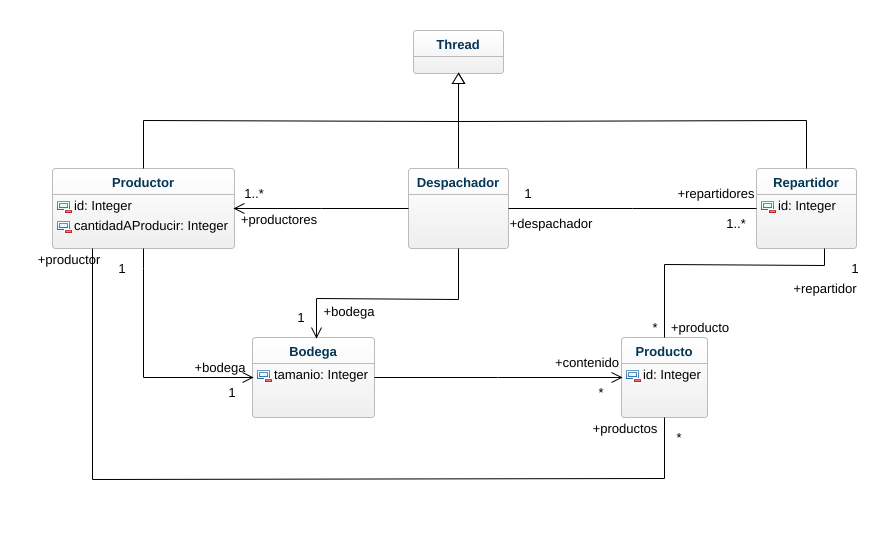


Figura 1. Diagrama de clases de la solución

Los productores tienen asociado un id y una cantidad de productos totales que deben producir. El despachador es uno y sólo uno; aparte, no tiene atributos. Los repartidores tienen un id asociado. La bodega tiene dos atributos: un tamaño y un arreglo con los productos que tenga almacenados en un momento dado. Los productos, por su parte, también tienen un id para facilitar la lectura de los mensajes en consola y el seguimiento del proceso de despacho y repartición.

En términos de las relaciones entre clases, un productor conoce a la bodega y a los productos que produce. El despachador conoce a los productores y a los repartidores, así como a la bodega con la cual interactúa. Los repartidores sólo conocen a los productos que reparten y al despachador, pero jamás conocen a la bodega ni a los productores. Finalmente, un producto conoce tanto a su productor como a su repartidor.

## Funcionamiento global del sistema

El sistema de producción y despacho tiene múltiples hilos, incluyendo los productores, despachadores y repartidores, teniendo un diseño basado en las siguientes clases:

* **Productor**: Produce los productos para después almacenarlos en la bodega.
* **Bodega**: Representa el almacenamiento de los productos.
* **Despachador**: Controla el proceso de despacho y asigna productos a los repartidores.
* **Repartidor**: Entrega los productos a los clientes.
* **Producto**: Representa los productos y su estado (ocupado/disponible).

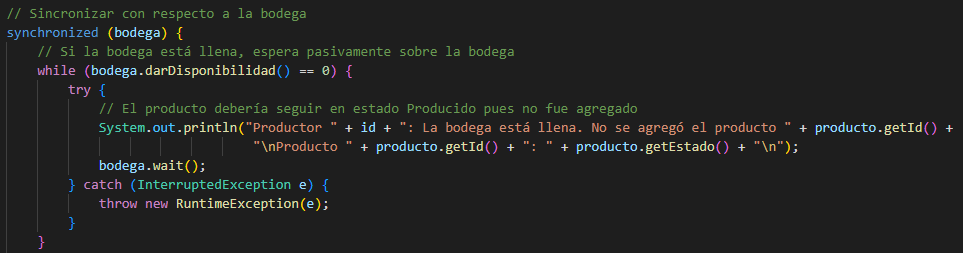
El programa inicia con la clase Principal donde se dan los parámetros de número de productores, repartidores, tamaño de bodega y total de productos a producir, para después iniciar las instancias de Bodega, Despachador, Productor y Repartidor.

Los productores son los encargados de crear los productos y almacenarlos en Bodega, donde si está llena se espera de manera pasiva. De ahí el Despachador monitorea la bodega y asigna productos a los Repartidores disponibles, esperando a medida que los productos llegan para así asignarlos. Finalmente, los Repartidores recogen los productos del Despachador y esperan un tiempo determinado que simula el tiempo de entrega, para después notificar que el proceso ha terminado.

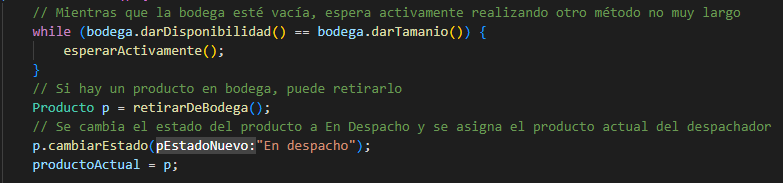
## Explicación de la sincronización entre pares de objetos

La sincronización se realiza utilizando métodos sincronizados y llamadas entre wait ( ) y notify ( ), esto se puede ver por ejemplo cuando los productores están esperando ya que la bodega está llena. Esto se da de la siguiente manera:

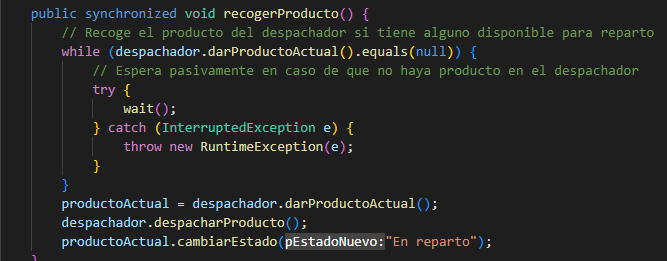
* **Productor – Bodega:** La sincronización aquí es del productor para agregar productos y esperar en dado caso que la bodega esté llena, esto con la bodega haciendo una espera y notificando a todos para dar la espera mientras se desocupa.



* **Despachador – Bodega:** Aquí se da al verificar la disponibilidad de productos en la bodega, esperando si está vacía, usando el despachador una espera activa hasta que se encuentren productos disponibles para despacharlos.



* **Despachador – Repartidores:** Aquí se asignan los productos a los repartidores disponibles, esperando mientras se desocupan para así recoger el producto y notificar cuando se ha realizado la recogida y entrega después de realizar una espera para simular el tiempo de entrega.



## Pruebas de validación

La validación se puede verificar a través de la ejecución del código, ya que este imprime los distintos pasos que se van realizando y asimismo el camino que recorren los productos a través de las distintas estaciones en las cuales se encuentran.

A continuación, se muestra el inicio de la ejecución para diferentes entradas. Se muestra la asignación de productos a producir a cada productor, la producción de los productos, la asociación que se crea entre Productores y Productos, el almacenamiento en bodega y otros mensajes de respuesta particulares:

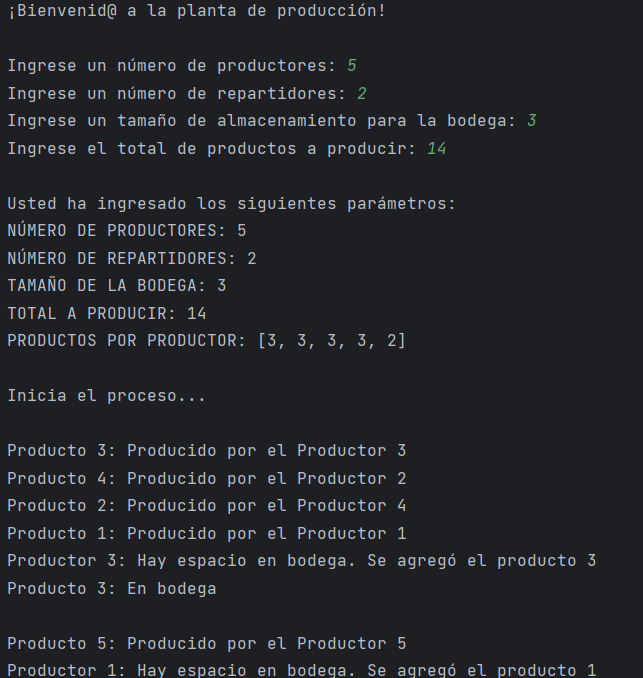


Figura 2. Caso de prueba (5,2,3,14)

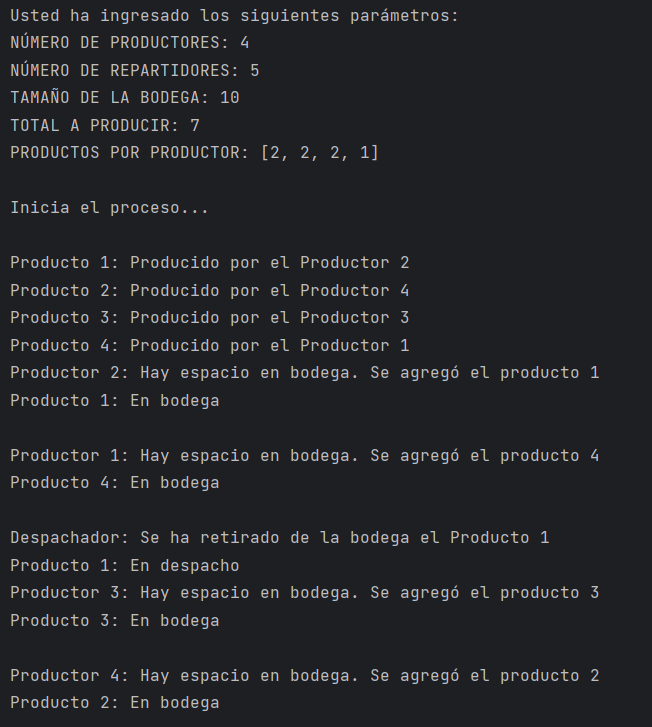


Figura 3. Caso de prueba (4,5,10,7)

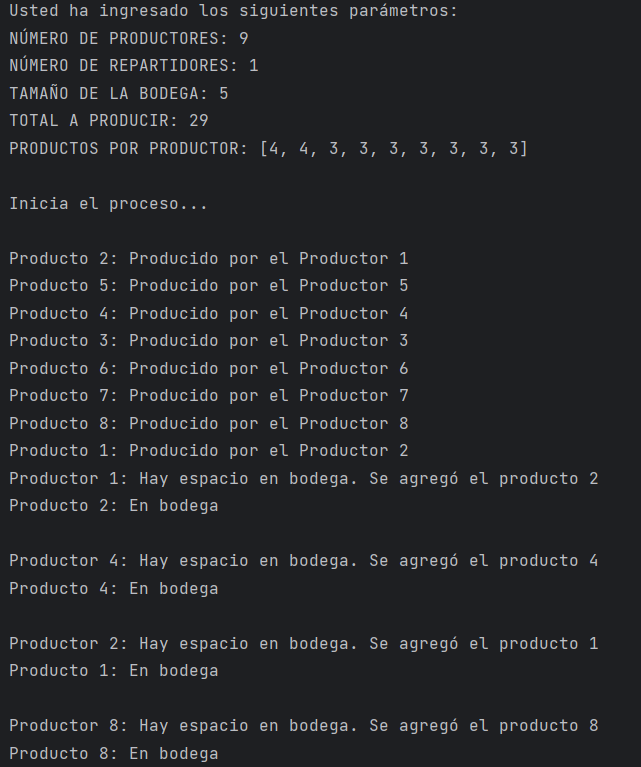


Figura 4. Caso de prueba (9,1,5,29)

**Observaciones finales:**

Se logró implementar adecuadamente al Productor. Así mismo, la bodega funciona adecuadamente en espera activa. Esto se demuestra en algunas ejecuciones en las cuales, si se libera un espacio de la bodega, puede ingresar un producto que antes no había podido ingresar. Así mismo, el despachador es capaz de retirar un producto de la bodega.