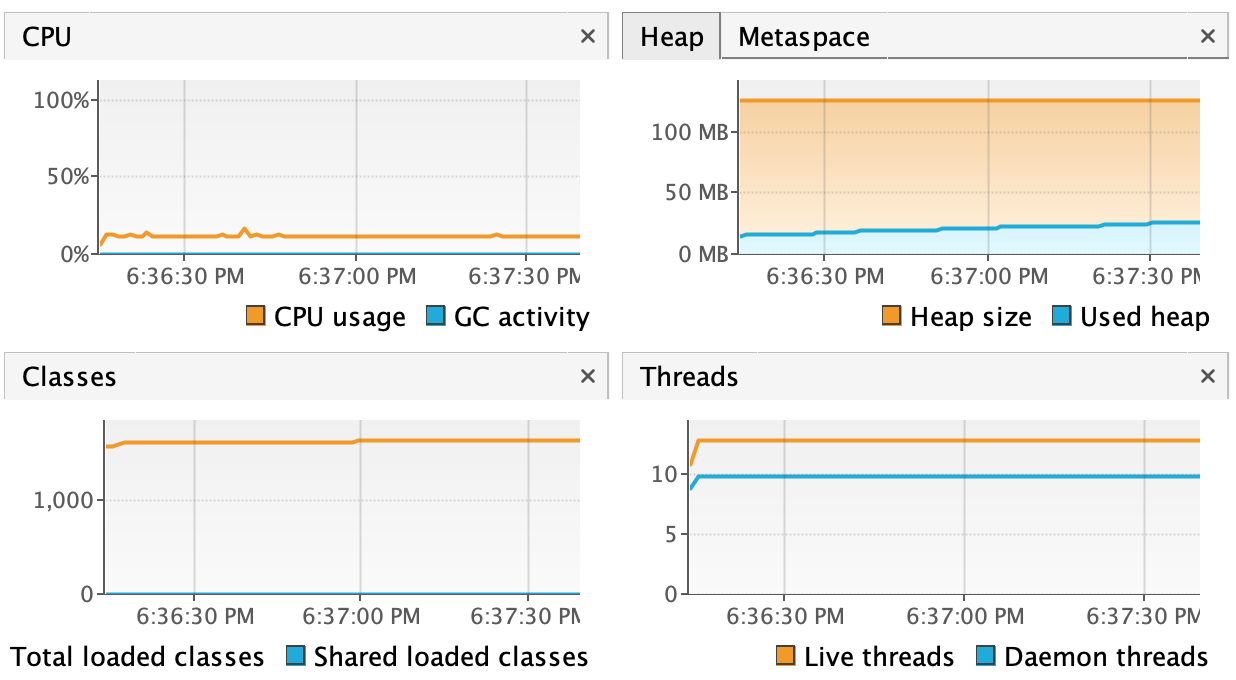
Laboratorio 3

Wilmer Arley Rodríguez Ropero

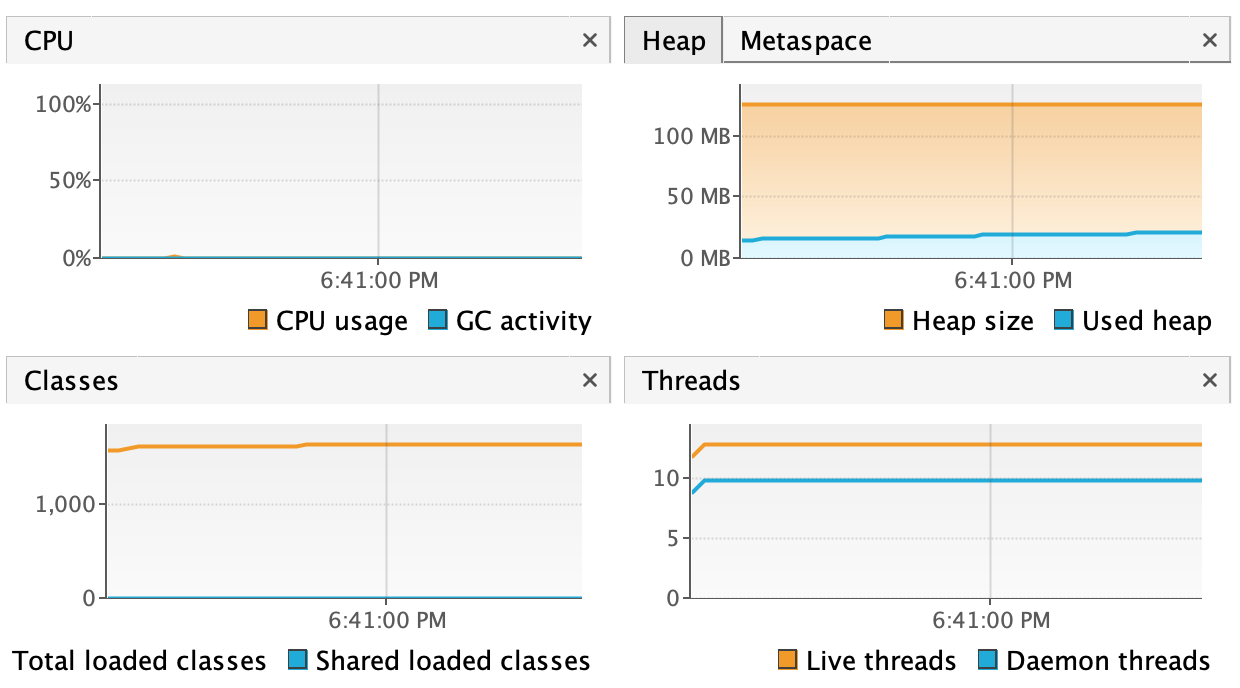
Daniel Sebastián Ochoa Urrego

**Parte 1**



El consumo se debe a que la clase consumidora revisa constantemente a que el productor haya agregado algo a la lista, mientras que el productor agrega elementos cada segundo. Viendo la aplicación, podemos ver que el uso de la CPU ronda un 12%

1. Para hacer que el programa usara menos la CPU hicimos que la clase consumidora esperara el mismo tiempo que productor antes de consumir un numero

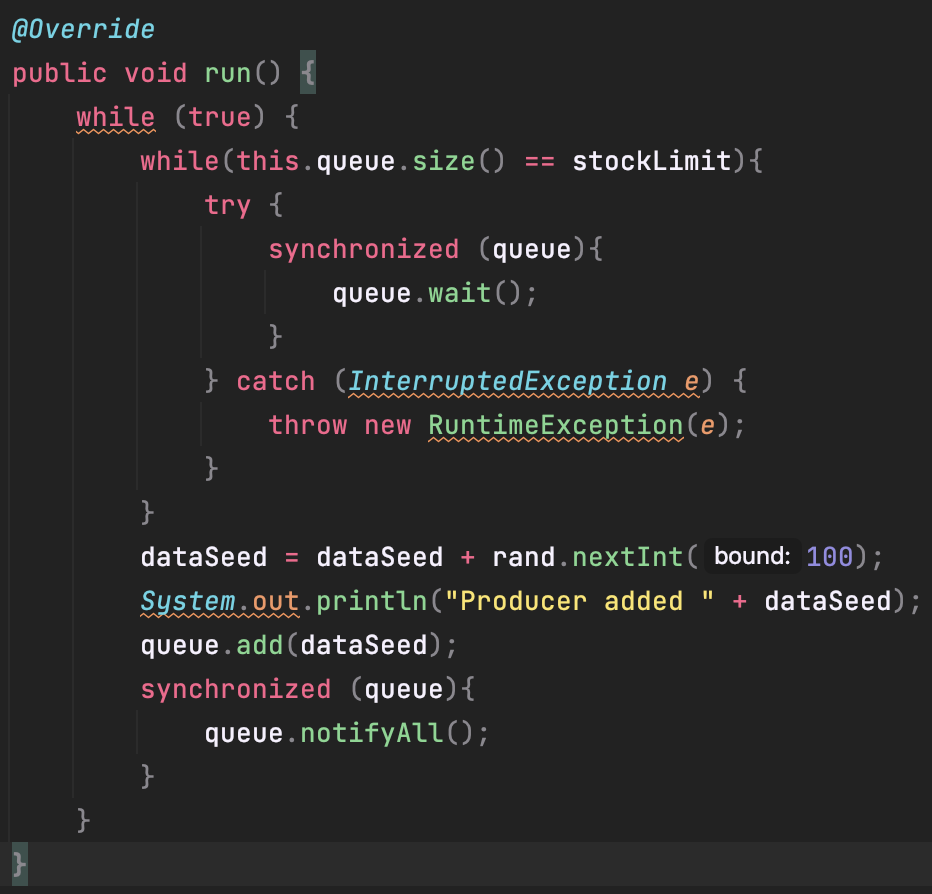


Y si revisamos el uso de la CPU vemos que este bajo a rondar el 0.1%

1. Para añadir esta funcionalidad creamos un atributo que representara el máximo stock en la clase que contiene el main



Y la clase productora la modificamos de la siguiente manera para que tuviera en cuenta el valor límite de producción, y se produjeran elementos en el momento en que se consuman, sin hacer sobre procesamiento de datos



Y la clase consumidora quedo de la siguiente manera haciendo uso de wait/notify



Y al correr el programa y ver su rendimiento en JVisualVM podemos ver como se tiene la misma funcionalidad de antes a un rendimiento mucho mejor

Graphical user interface, application, table, Excel

Description automatically generated

**Parte 2**

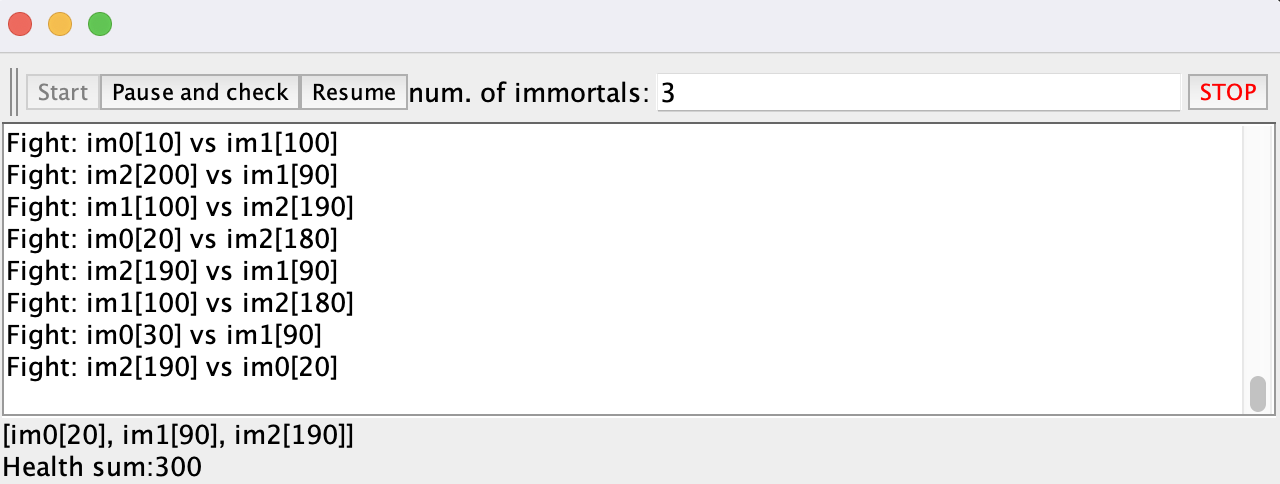
* Asumiendo que la vida de los inmortales sea siempre de 100, para N jugadores se tendrá que la vida de todos será igual a 100\*N
* No, el numero varía mucho, a veces sube y a veces baja, además de que no se pausa el juego
* Para implementar la pausa creamos lo siguiente en el actionListener del botón



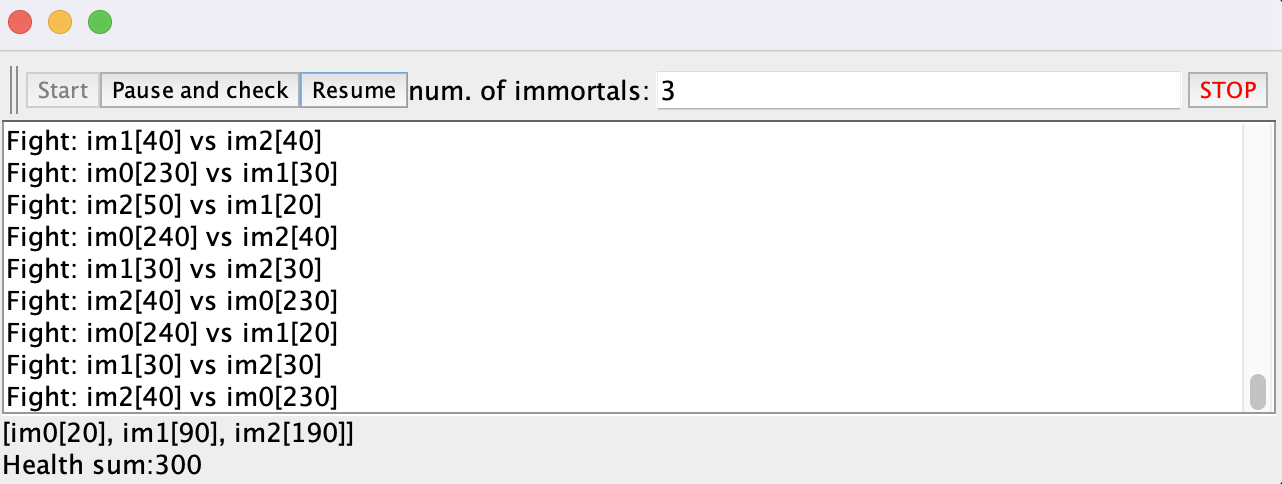
Y en los hilos de los inmortales creamos el método checkPause que revisa si los hilos tienen que parar y realizar lo correspondiente si es así



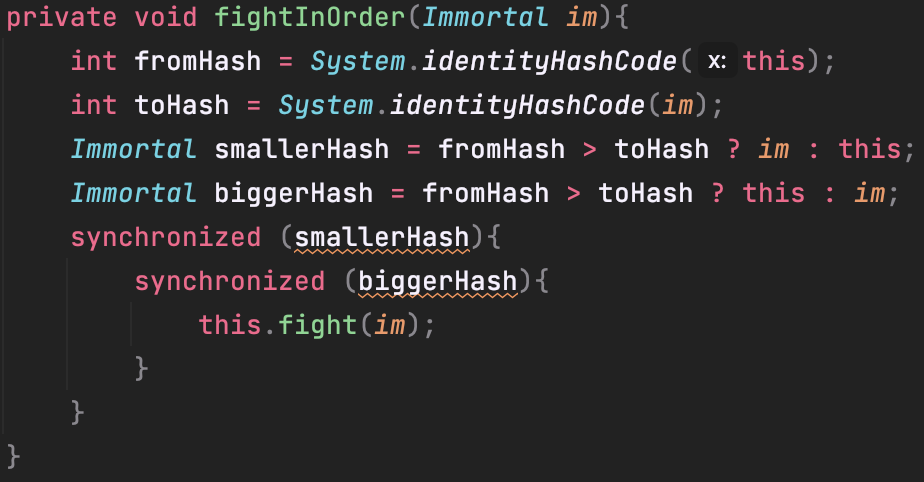
Y al pausar la aplicación podemos ver que los hilos en efecto se detienen y se imprimen su estado



Y al resumir continúan con su ejecución



* Y al hacer click en el botón también podemos ver como aún no se cumple este invariante
* Luego para arreglarlo vimos que otra región critica donde fallaba el programa era en las peles, se debe asegurar que los inmortales peleen entre ellos sin que se crucen los cálculos en el proceso, para ello implementamos el método fightInOrder



Y al ejecutar el programa aun con 1000 hilos, vemos que este invariante ahora si se mantiene

