

## Hoja de trabajo 6

Gerardo Pineda 22880

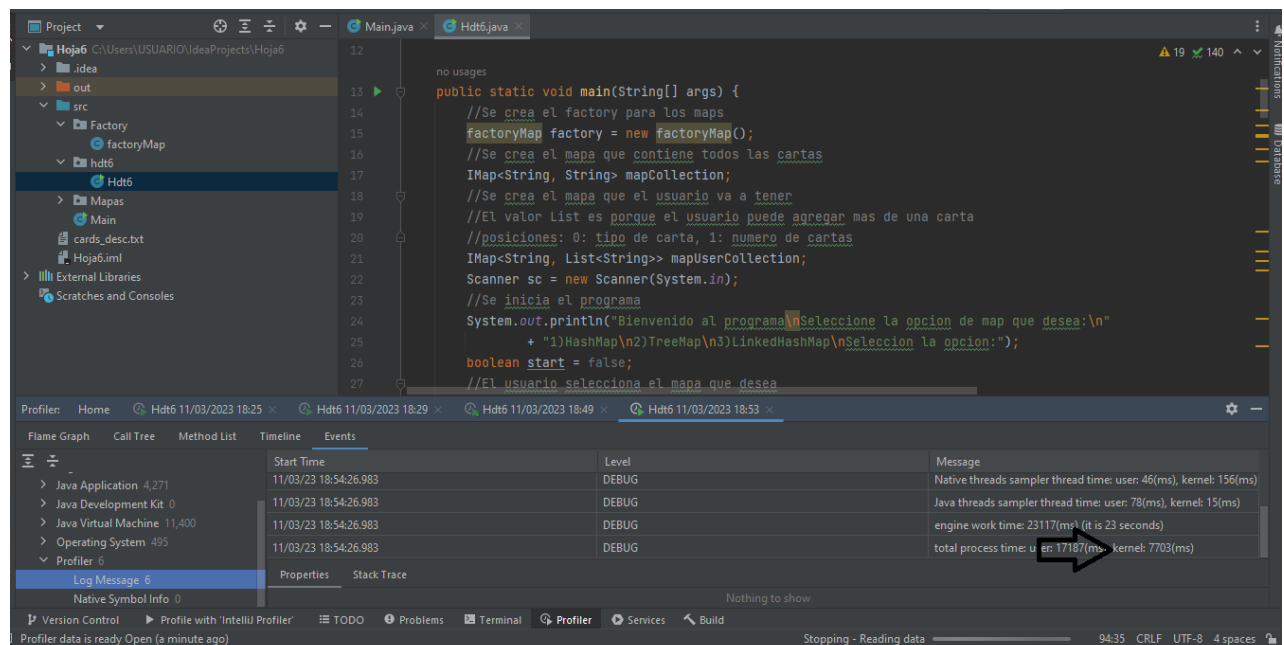
### Repositorio

<https://github.com/Geraxxx5/HDT6.git>

### Profiler y análisis

Se uso el Profiler de IntelliJ para obtener el tiempo en que tarda cada mapa por corrida. Debido a que el Profiler de IntelliJ toma la corrida completa, en vez de ser dinámico el programa se cambiaban variables para que corriera en línea. Por cada mapa se cambió la selección de mapa y selección entre mostrar todas las cartas o mostrar las cartas ordenas por el tipo.

(Esto solo es una imagen muestra)



### Análisis de tiempo

#### HashMap

Todas las cartas: 25612 ms

Ordenadas: 25185 ms

#### TreeHashMap

Todas las cartas: 26436 ms

Ordenadas: 25702 ms

#### LinkedHashMap

Todas las cartas: 25349 ms

Ordenadas: 24020 ms

### **Conclusión**

Como se puede observar el que tomo más tiempo fue el TreeMap esto puede deberse a que como su nombre indica el funciona mediante un árbol binario, lo que hace que al iterar los datos este tenga que recorrer el árbol completo, en vez de ir en línea recta como un árbol enlazado.

LinkedHashMap y un HashMap son muy parecidos al iterar los datos por lo que se puede ver de los tiempos.

Al iterar los datos del hashmap para mostrar los datos, se utilizó un for each. Por lo tanto, este for de por si tiene complejidad  $O(n)$  y para acceder a cada dato es la misma complejidad siempre, por lo que esta complejidad no afecta al resultado, es decir  $O(1)$ . Juntando esto tendríamos una complejidad  $O(n)$  para la iteración de datos.