

Análisis con el profiler

HashMap

```
✓ Tests passed: 1 of 1 test - 267 ms
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\JetBrains\
Wave-Motion Cannon=Hechizo
Revival of Dokurorider=Hechizo
Final Ritual of the Ancients=Hechizo
7=Hechizo
Cursed Bill=Hechizo
Shared Ride=Hechizo
```

TreeMap

```
✓ Tests passed: 1 of 1 test - 358 ms
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\JetBrains\
"A" Cell Breeding Device=Hechizo
"A" Cell Incubator=Hechizo
"A" Cell Recombination Device=Hechizo
"A" Cell Scatter Burst=Hechizo
1st Movement Solo=Hechizo
7=Hechizo
7=Hechizo
```

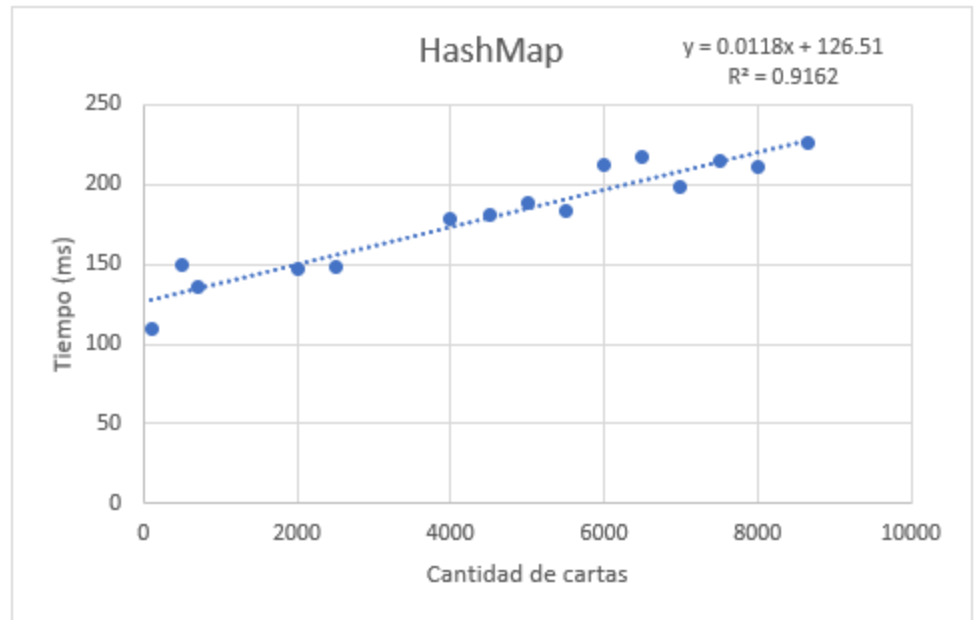
LinkedHashMap

```
✓ Tests passed: 1 of 1 test - 258 ms
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagent:C:\Program Files\JetBrains\
"A" Cell Breeding Device=Hechizo
"A" Cell Incubator=Hechizo
"A" Cell Recombination Device=Hechizo
"A" Cell Scatter Burst=Hechizo
1st Movement Solo=Hechizo
7=Hechizo
7=Hechizo
```

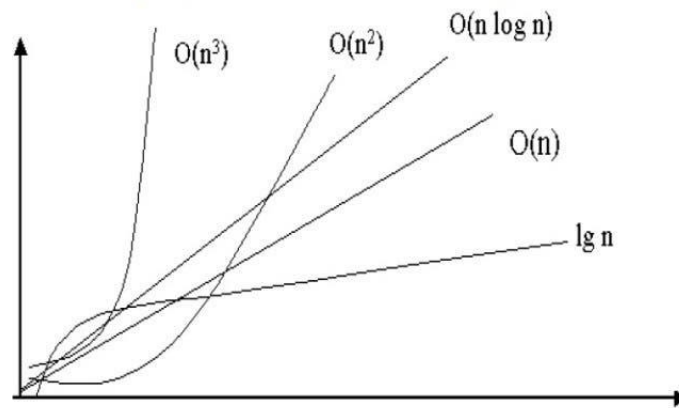
El tipo de Map que menor tiempo se tardó en realizar el ordenamiento completo de todas las cartas fue el LinkedHashMap con 258ms, el segundo más rápido fue el HashMap con 267ms y el más lento fue el TreeMap con 358ms.

Análisis del HashMap:

Datos	Tiempo
8660	226
8000	211
7500	215
7000	199
6500	217
6000	212
5000	189
5500	183
4500	181
4000	179
2000	147
2500	148
700	136
500	150
100	110



Complejidades típicas de un algoritmo



Tras analizar la grafica de las diferentes pruebas realizadas para el HashMap se puede observar una complejidad que tiene como tendencia $O(n)$.