

## Hoja de Trabajo 6

### HashMap Time

Name	Total Time	Total Time (CPU)	Invocations
main	45.4 ms (100%)	38.7 ms (100%)	8,862
Card.toString ()	49.5 ms (109%)	42.7 ms (110.6%)	8,861
Main.secondaryMenu ()	0.019 ms (0%)	0.0 ms (0%)	1

### TreeMap Time

Name	Total Time	Total Time (CPU)	Invocations
main	49.7 ms (100%)	7.45 ms (100%)	8,862
Card.toString ()	53.8 ms (108.2%)	11.5 ms (154.8%)	8,861
Main.secondaryMenu ()	0.016 ms (0%)	0.0 ms (0%)	1

### LinkedHashMap Time

Name	Total Time	Total Time (CPU)	Invocations
main	51.5 ms (100%)	23.0 ms (100%)	8,862
Card.toString ()	55.5 ms (107.9%)	27.1 ms (117.7%)	8,861
Main.secondaryMenu ()	0.048 ms (0.1%)	0.0 ms (0%)	1

- En base al Profiler Visual VM la implementación más rápida es la de hashmap.

- Complejidad de HashMap**

Una implementación de hash debe proporcionar un código hash único para cada objeto, esto otorga una complejidad constante  $O(1)$ . En caso haya una mala implementación habrá una colisión de hash. En este caso, la complejidad sería  $O(n)$ .

Las ventajas de estas estructuras es de que su comportamiento es como el de un Array. Esto quiere decir que el tiempo de acceso a un elemento no depende del tamaño de la estructura, la complejidad será  $O(1)$ .