



**UNIVERSIDAD SIMÓN BOLÍVAR**

Reporte: Estudio experimental  
Laboratorio semana 8

Profesor:  
Guillermo Palma

Estudiantes:  
Arthur Ortega 16-10811  
Baudilio Velasquez 18-10665

Julio de 2023

El siguiente reporte esta basado en el estudio de rendimiento de las tablas de hash, HashTableChaining y CuckooHashTable respectivamente, donde se tomo en consideración el tiempo medio y la desviación estándar respectiva en la ejecución de un arreglo de 5.000.000 de elementos “clave, valor”, y que para obtener estos datos el proceso se realizo un total de 5 veces, esto con el fin de comparar que tan eficientes son estas tablas con un arreglo de esta magnitud. El estudio fue realizado en una computadora con las siguientes características.

**Sistema Operativo:** Debian GNU/Linux 11 (bullseye) 64 bits.

**Procesador:** Intel core i5-4590 a 3,30Ghz(x4).

**Memoria RAM:** 16GB DDR3 1600Mhz.

**Compilador de kotlin:** version 1.8.20 (JRE 11.0.18+10-post-Debian-1deb11u1).

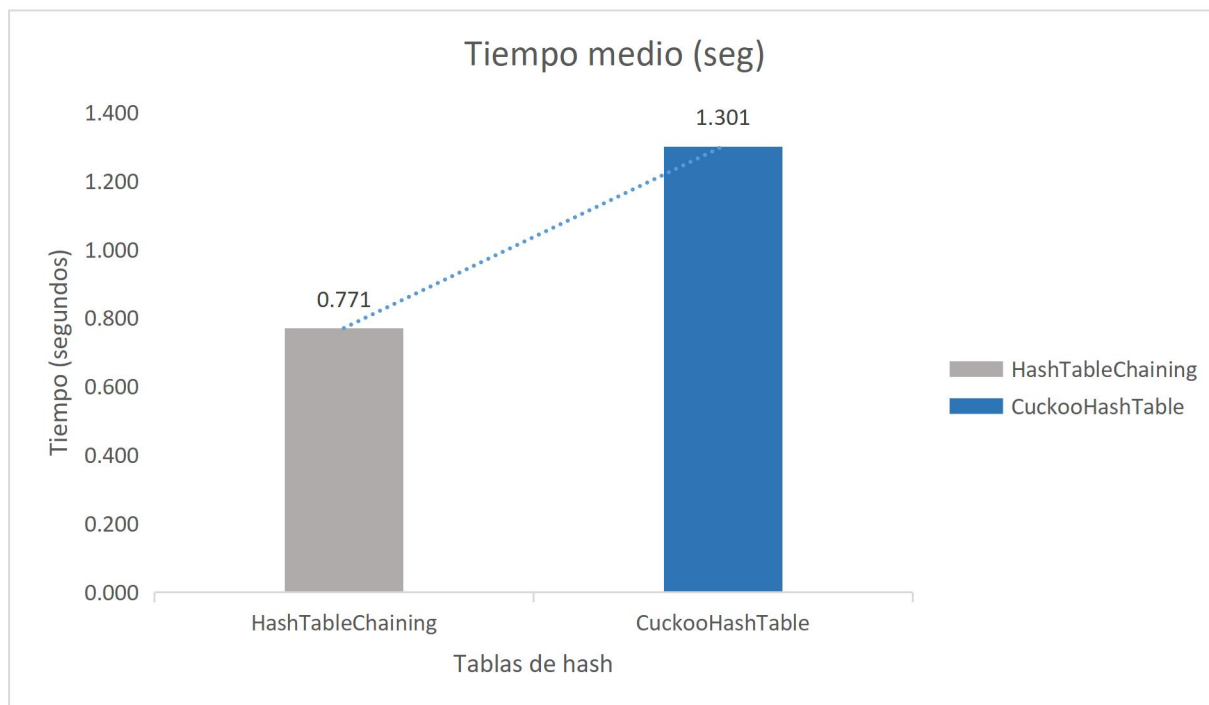
**JVM:** build 11.0.18+10-post-Debian-1deb11u1.

Para el estudio fue utilizado la opción -Xss5m para aumentar el tamaño de la pila de memoria de la maquina virtual de Java, ya que al utilizar cualquiera de los métodos, se generaban demasiadas llamadas recursivas, lo que provocaba que la pila de memoria se desbordara, por lo que se aumento el tamaño de la pila de memoria para evitar este problema.

### Tabla de rendimiento de HashTableChaining y CuckooHashTable

Tabla de hash	Tiempo medio (seg)	Desviacion estandar (seg)
HashTableChaining	0.7706788	0.0349638
CuckooHashTable	1.3012017	0.1465226

### Grafico de rendimiento de HashTableChaining y CuckooHashTable



Para los resultados obtenidos en la prueba, se obtuvo la implementación de tabla de hash con el método de encadenamiento con listas doblemente enlazadas como el más eficiente de ambas, con un tiempo medio de 0.77 segundos, con una desviación estandar ubicada en 0.03 segundos. Por otro lado, la implementación con las tablas de hash utilizando el método de Cuckoo hashing obtuvo un tiempo medio de 1.3 segundos, con una desviación estandar de 0.14 segundos.

Sin embargo, para un tamaño menor de claves (1000 elementos), el método de Cuckoo hashing obtuvo un tiempo medio más eficiente. Esto se puede deber a la cantidad de rehashs que el método de colisiones acabó usando para agregar elementos nuevos en caso de que dos elementos ocupen la misma clave en la tabla.