

OBSERVACIONES DEL LA PRACTICA

Iván Camilo Ballén Méndez Cod 202011440

María José Sáenz Rodríguez Cod 202013542

	Máquina 1	Máquina 2
Procesadores	Intel(R) Core (TM) i5-8300H CPU @ 2.30GHz, 2304 Mhz, 4 procesadores principales, 8 procesadores lógicos	AMD R5 2.1 GHz - 3.7 GHz
Memoria RAM (GB)	8.0 GB	8.0 GB
Sistema Operativo	Microsoft Windows 10 Home Single Language	Windows 10 Pro-64-bits

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Maquina 1

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	363506.38	7140.34
0.50	363399.55	7292.91
0.80	363324.74	7645.95

Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 1.

Carga de Catálogo CHAINING

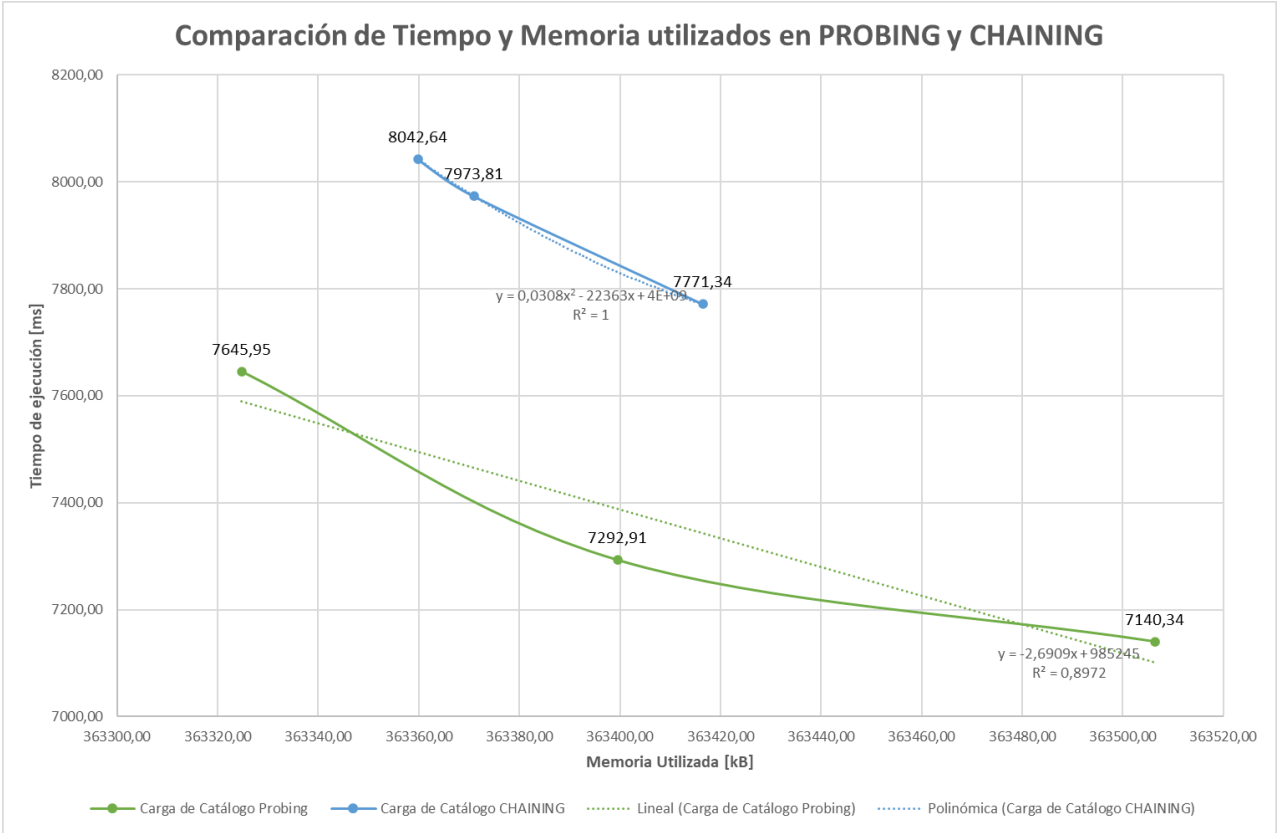
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	363416.47	7771.34
4.00	363370.94	7973.81
6.00	363359.82	8042.64

Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

Graficas

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 1**.

- Comparación de memoria y tiempo de ejecución para PROBING y CHAINING



Maquina 2

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	363439,185	15929,369
0.50	363399,617	16701,351
0.80	363399,56	17355,091

Tabla 4. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 2.

Carga de Catálogo CHAINING

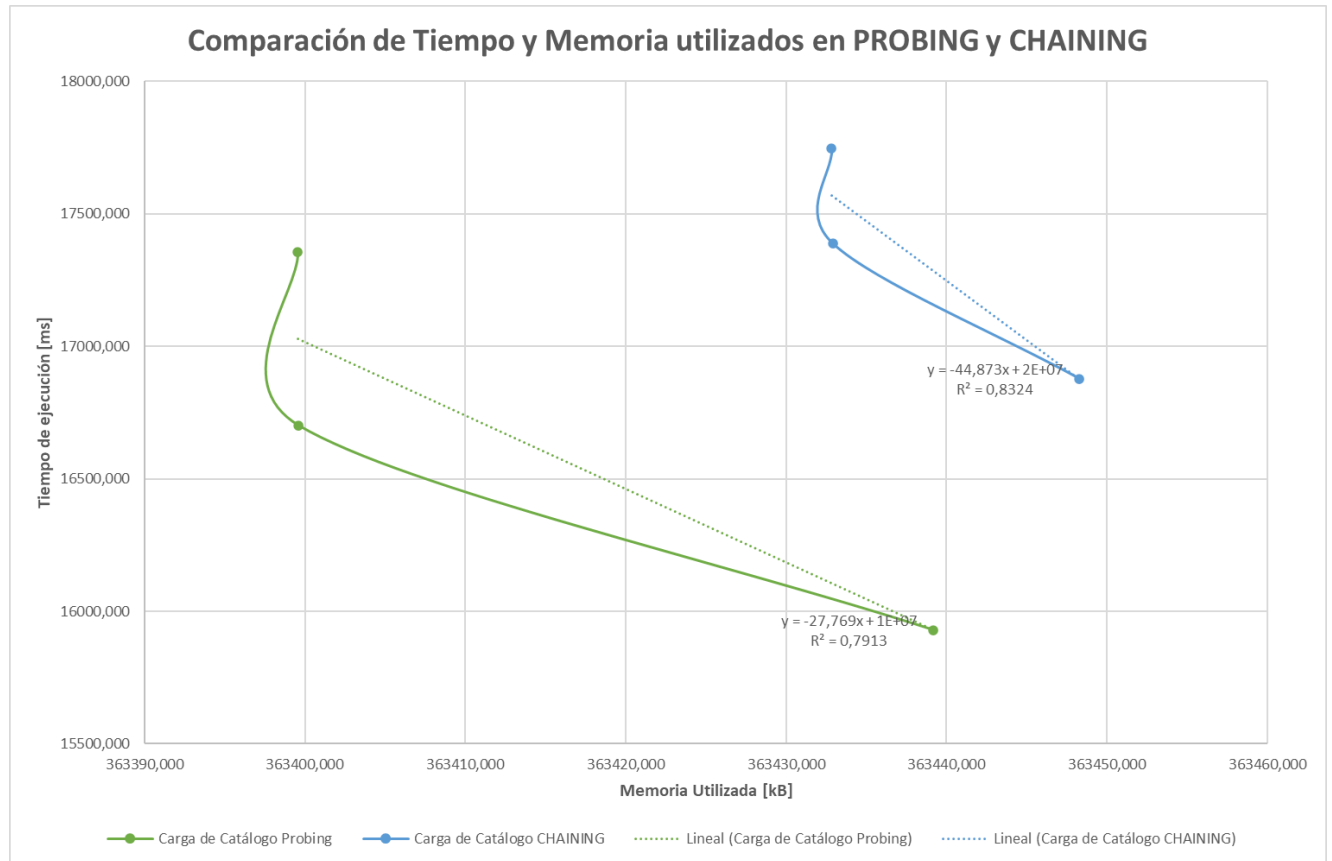
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	363448,285	16877,487
4.00	363432,898	17387,695
6.00	363432,838	17748,814

Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Máquina 2.

Graficas

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 2**.

- Comparación de memoria y tiempo de ejecución para PROBING y CHAINING



Preguntas de análisis

- 1) ¿Por qué en la función **getTime()** se utiliza **time.perf_counter()** en vez de la previamente conocida **time.process_time()**?

En este caso se usa el método `.perf_counter()` en vez del método `process_time()` debido a que el método `.process_time()` no tiene en cuenta el tiempo en “sleep” de la pc. Luego, si el proceso se demora bastante y el pc entra en modo de reposo, obtendremos un tiempo falso que no tiene en cuenta el trabajo que hizo el sistema en modo reposo. Por otro lado, el método `.perf_counter()` tiene en cuenta el tiempo en “sleep” por lo que en caso de que la prueba demore bastante, se van a obtener datos confiables.

- 2) ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?

Según la documentación de la librería `tracemalloc`, los métodos `start()` y `stop()` cumplen la función de preparar y terminar el “rastreo” de memoria. En otras palabras, al ejecutarse la línea `tracemalloc.start()`, el pc inicializa un método para “rastrear” o medir la memoria que se está usando. Cuando se ejecuta el `tracemalloc.stop()`, este método que a lo mejor puede consumir memoria se termina, es decir, se deja de “rastrear” memoria puesto que la medición ya se realizó y seguir con el método activo puede significar una eficiencia menor.

- 3) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de videos?

De acuerdo con los resultados obtenidos por las maquinas, se evidencia que los tiempos de ejecución son mayores cada vez que se aumenta el factor de carga, teniendo en cuenta cualquier mecanismo para la solución de colisiones. Una causa de esto es que en el caso del mecanismo “CHAINING” al ampliar el factor máximo, se crean tablas más pequeñas y por lo tanto un incremento en los tiempos de ejecución. Por otro lado, en el mecanismo “PROBING” cuando se realiza el cambio en los factores, se observa que entre menor sea este las tablas serán mas grandes lo que implica un menor tiempo de ejecución en la carga de datos.

- 4) ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar el catálogo de videos?

Como se expresó en el punto anterior, al aumentar los factores de carga en los dos mecanismos de manejo para las colisiones, las tablas de hash que se crean son mucho más pequeñas, por lo que no hay un desperdicio de memoria y este valor disminuye considerablemente. Para el caso de las maquinas 1 y 2, se percibe como una ventaja, puesto que solo se tienen 8GB para cargar los datos.

- 5) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

Según las dos tablas de valores para las maquinas, se observa un mismo comportamiento y es que en el manejo de colisiones "CHAINING" los tiempos de ejecución son mayores en comparación con el mecanismo "PROBING", debido a que los factores de carga elegidos en "CHAINING" son muy altos. Generalmente, se busca que estos se aproximen a el valor de 1, ya que como se mencionó en el punto 3, entre mayor sea el factor de carga, se generan tablas muy pequeñas aumentando el tiempo para encontrar un dato.

- 6) ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

El consumo de memoria en los dos mecanismos de colisiones disminuye, sin embargo no hay una diferencia notoria, debido a que los valores se compensan con factores como la memoria de las maquinas y los valores de los factores de carga elegidos previamente. Aun así, es importante resaltar que esta disminución para el caso del mecanismo "PROBING" se debe a que entre mayor sea su factor, la tabla de hash tendrá un porcentaje de ocupacion alta, su memoria será menor y en consecuencia, se darán más colisiones. Por otro lado, en el mecanismo "CHAINING" sucede algo similar, donde al tener altos valores correspondientes al factor de carga, se permite tener un menor espacio ocupado en la memoria, no obstante, las consecuencias se ven reflejadas en los tiempos de ejecución.

Aclaraciones:

Las pruebas realizadas en este laboratorio se llevaron a cabo sobre ciertas implementaciones ya adelantadas del reto 2, es decir, no se realizaron únicamente sobre el mapa de categorías desarrollado y entregado en el laboratorio 6 sino que se tuvieron en cuenta 4 maps dentro del catálogo que facilitan el acceso a cierta información requerida en el reto 2. A continuación damos las especificaciones de cada map:

- 1.) catalog['category']: Es un mapa que guarda como llave el nombre de una categoría y como valor su ID.
- 2.) catalog['VideosByCategory']: Es un mapa de mapas que guarda como llave el ID de una categoría y como valor guarda un mapa que tiene como llaves los países de videos y como valor la lista de videos que comparten un mismo país.
- 3.) catalog['VideosByCountry']: Es un mapa que guarda como llaves los países de todo el catálogo de videos y como valor las listas de videos que comparten un país común.

- 4.) Mapa auxiliar de países: este mapa es el que se guarda dentro de cada tupla llave-valor del mapa `catalog['VideosByCategory']` el cual tiene la misma estructura llave valor del mapa `catalog['VideosByCountry']`

Al tener 3 maps adicionales al map de categorías que se tenía originalmente, argumentamos teóricamente ciertos cambios respecto a las pruebas que se debían realizar con solo un map. Primero, esperamos teóricamente un espacio usado mucho más grande debido a que lógicamente, 4 maps y de hecho un map que como valores guarda maps utiliza mucho más espacio que un solo map. Esto también es consecuencia de que el espacio usado lo medimos en la carga total de archivos y no en la carga de cada proceso. También esperamos ver un aumento significativo en el tiempo debido a que estadísticamente la pc estará lidiando con más colisiones promedio que con un solo map. Sin embargo, cabe resaltar que este incremento en las colisiones no necesariamente debe ser 4 veces mayor debido a que no todos los maps adicionales son maps de categorías, esto implica que la frecuencia promedio de colisiones sea diferente para llaves de categorías con respecto a otras llaves que también se usan como lo son los países. Este aumento en las colisiones conlleva un aumento en la medida de tiempo, entre más colisiones, más tiempo toma cargar los archivos. Por último, cabe resaltar que nuestras pruebas no implican un cambio significativo en el análisis de resultados del experimento. Teorizamos como hipótesis que estos 4 maps adicionales solo cambiarán las magnitudes de tiempo y espacio pero estas serán proporcionales a las magnitudes obtenidas si solo tuviéramos un map. Esto lo deducimos de que la complejidad temporal y espacial de un map depende directamente de la estructura y el factor de carga, por lo que a pesar de modificar las proporciones de la prueba, las relaciones deben tender a ser las mismas.