

OBSERVACIONES DE LA PRÁCTICA

Daniela Alvarez Rodriguez 202020209 d.alvarezr

Máquina		
Procesadores	AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz	
Memoria RAM (GB)	8,00 GB (5,94 GB usable)	
Sistema Operativo	Windows 10 Home Single	

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

Pruebas de rendimiento LabCollisions

Resultados

<u>Carga de Catálogo PROBING</u>		
Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	293052.121	92161.418
0.50	290920.902	101674.693
0.80	289643.020	142206.599

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

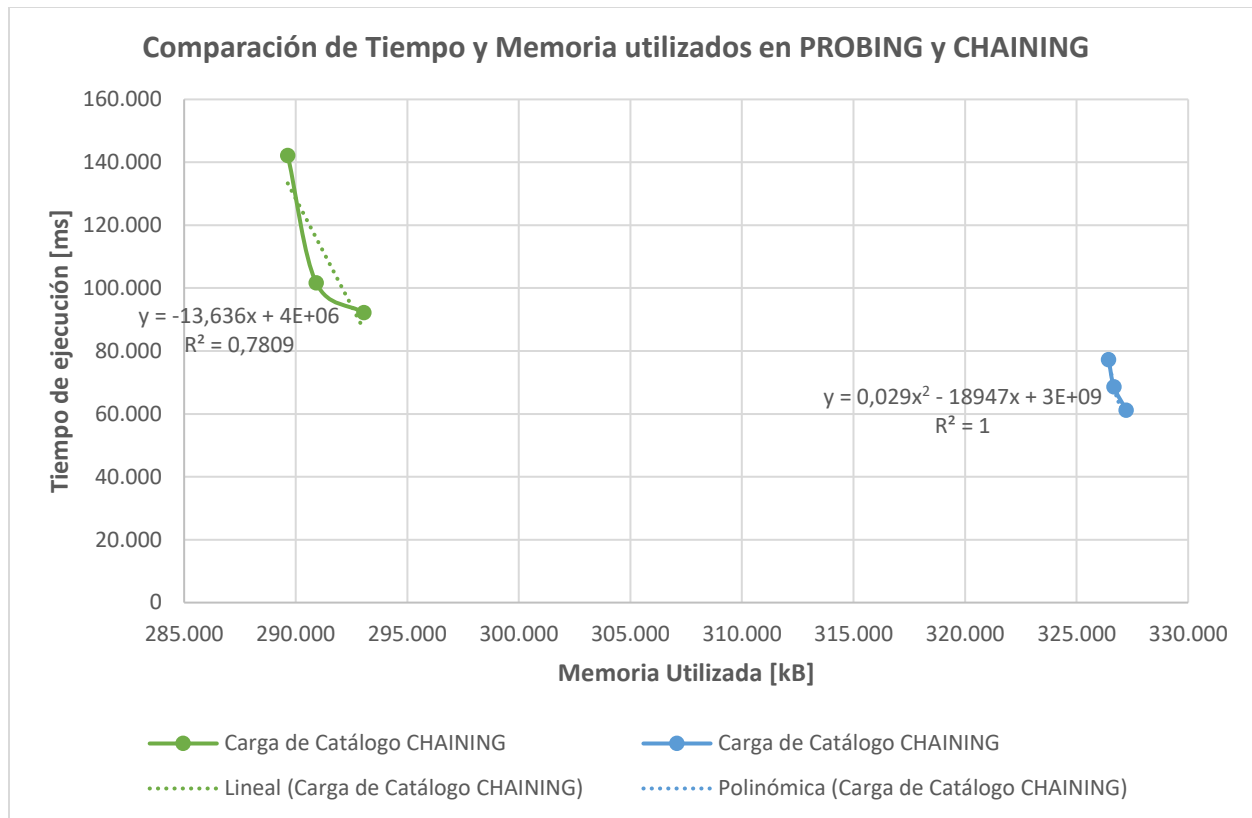
<u>Carga de Catálogo CHAINING</u>		
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	327225.002	61221.780
4.00	326675.666	68601.581
6.00	326436.568	77271.989

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

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



Preguntas de análisis

- 1) ¿Por qué en la función **getTime()** se utiliza **time.perf_counter()** en vez de la previamente conocida **time.process_time()**?
Se utiliza esta función porque es mucho más precisa. Esto teniendo en cuenta que **time.process_time()** incluye el tiempo del sistema y del proceso que está evaluando.
- 2) ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?
Porque de esta manera, no generamos trabajo innecesario al sistema haciéndole medir el tiempo de ejecución cuando ya calculamos el delta.
- 3) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de videos?
En ambas estructuras de tablas de símbolos noté que el factor de carga del Map y el tiempo de ejecución eran proporcionales, es decir, que entre mayor factor de carga, mayor tiempo de ejecución.
- 4) ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar el catálogo de videos?
En ambas estructuras de tablas de símbolos noté que el factor de carga del Map y el consumo de memoria eran inversamente proporcionales, es decir, que entre mayor el factor de carga, mayor tiempo de ejecución.

- 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.
 Note que el tiempo de ejecución en el esquema de CHAINING es mucho menor al de PROBING.
- 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.
 Note que el consumo de memoria en el esquema de CHAINING es mayor al de PROBING.

Pruebas de rendimiento Reto 2

Resultados

<u>Carga de Catálogo PROBING</u>		
Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	4739,966	206,200
0.50	4740,024	218,830
0.80	4740,024	195,598

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

<u>Carga de Catálogo CHAINING</u>		
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	4748,272	199,220
4.00	4748,272	221,899
6.00	4748,272	275,120

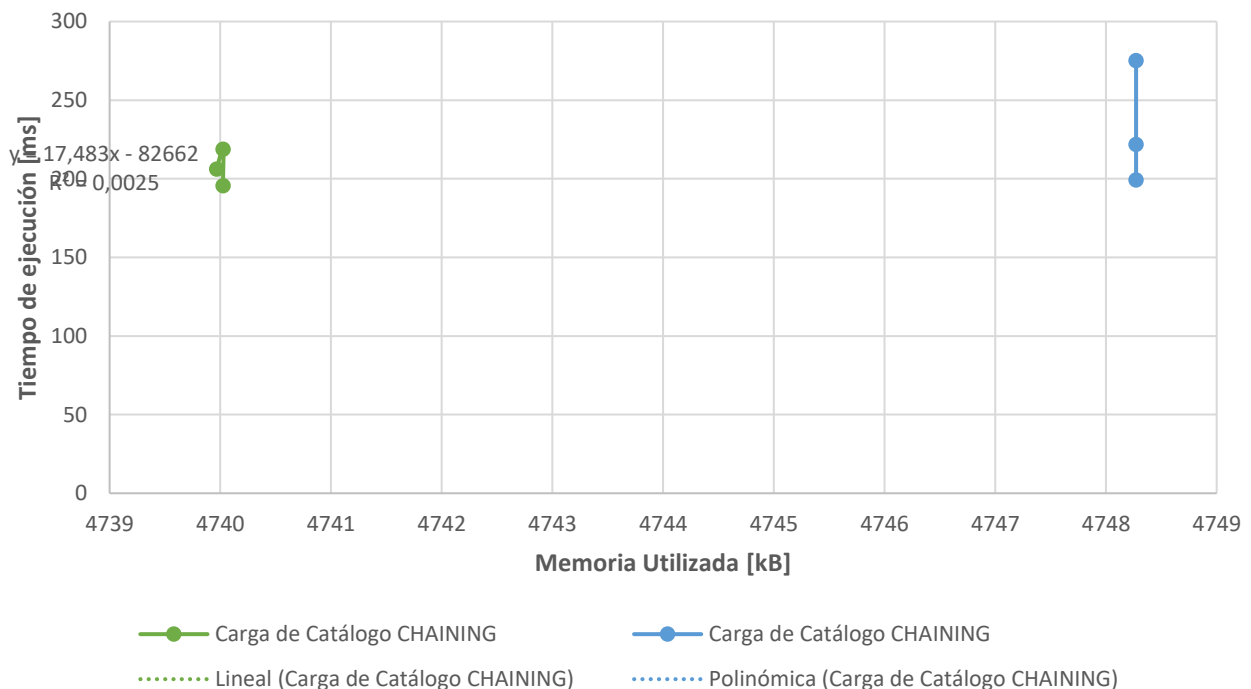
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Máquina.

Graficas

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

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

Comparación de Tiempo y Memoria utilizados en PROBING y CHAINING



Preguntas de análisis

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

Se utiliza esta función porque es mucho más precisa. Esto teniendo en cuenta que **time.process_time()** incluye el tiempo del sistema y del proceso que está evaluando.

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

Porque de esta manera, no generamos trabajo innecesario al sistema haciéndole medir el tiempo de ejecución cuando ya calculamos el delta.

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

En el método CHAINING noté que el factor de carga del Map y el tiempo de ejecución eran proporcionales, es decir, que entre mayor factor de carga, mayor tiempo de ejecución. Por el otro lado, en el método PROBING no vi ninguna relación entre el tiempo de ejecución y el factor de carga, teniendo en cuenta que variaba casi que aleatoriamente.

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

En ambas estructuras de tablas de símbolos noté que, aunque se varió el factor de carga del Map, el consumo de memoria fue constante. Lo anterior posiblemente ocurrió porque el tamaño del archivo de videos era pequeño

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

Como se mencionó anteriormente, en el método CHAINING noté que el factor de carga del Map y el tiempo de ejecución eran proporcionales, es decir que, entre mayor factor de carga, mayor tiempo de ejecución. Por el otro lado, en el método PROBING no vi ninguna relación entre el tiempo de ejecución y el factor de carga, teniendo en cuenta que variaba casi que aleatoriamente. Debido a lo anterior, no diferencié cual esquema fue el mas eficiente en ese sentido.

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

Note que el consumo de memoria fue constante en ambos esquemas, sin embargo, fue mayor en CHAINING.