OBSERVACIONES DEL LA PRACTICA

1. María Paula Nizo Vega, m.nizo@uniandes.edu.co, 202213902
2. Paul Paffen Suarez, p.paffen@uniandes.edu.co, 202222496
3. Andres Camilo Caballero Ayala, Ac.caballero@uniandes.edu.co, 202216295

	Máquina 1	Máquina 2	Máquina 3
Procesadores	AMD Ryzen 5 4600H 3.00 GHz	AMD Ryzen 7 4800 H	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
Memoria RAM (GB)	8,00 GB	16,00 GB	8,00 GB
Sistema Operativo	Windows 11 de 64 bits	Windows 11 – 64 bits	Windows 11 Pro

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 Real @LP [ms]
0.1	25064,199	95,772
0.5	25064,199	104,269
0.7	25064,199	107,615
0.9	25064,199	117,424

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

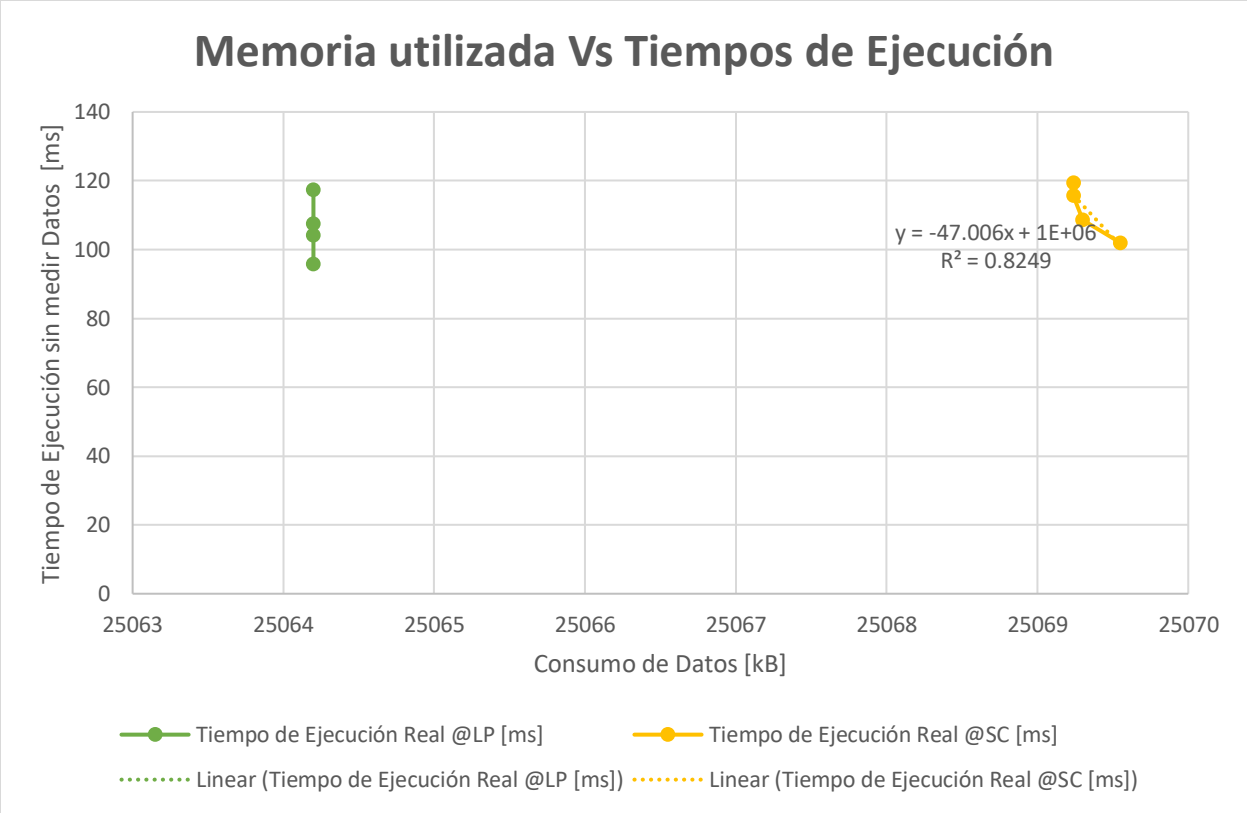
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	25069,551	101,961
4.00	25069,301	108,622
6.00	25069,238	115,667
8.00	25069,238	119,360

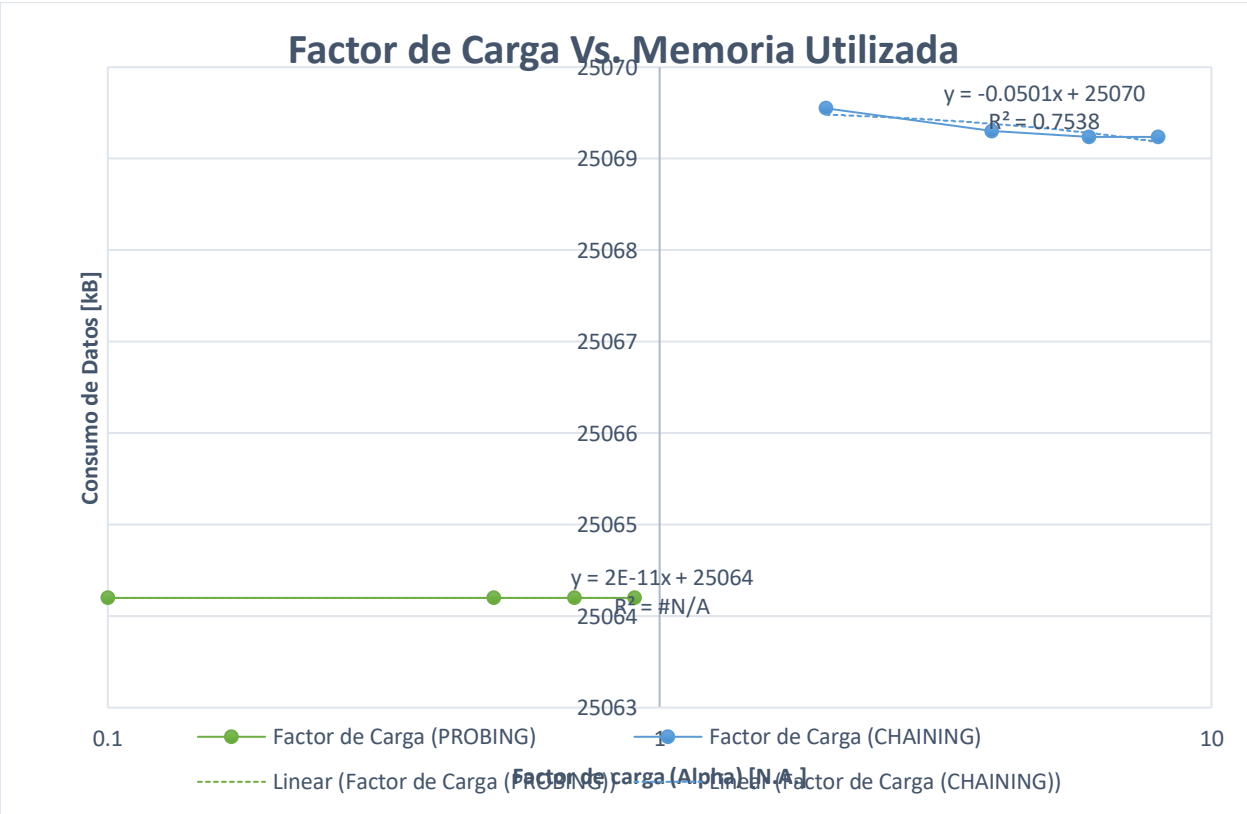
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

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


Memoria utilizada Vs Tiempos de Ejecución
Tiempo de Ejecución sin medir Datos [ms]
140
120
100
80
60
40
20
0
25063
25064
25065
25066
25067
25068
25069
25070
Consumo de Datos [kB]
y = -47.006x + 1E+06
R² = 0.8249
Tiempo de Ejecución Real @LP [ms]
Tiempo de Ejecución Real @SC [ms]
Linear (Tiempo de Ejecución Real @LP [ms])
Linear (Tiempo de Ejecución Real @SC [ms])


Factor de Carga Vs. Memoria Utilizada
Consumo de Datos [kB]
25070
25069
25068
25067
25066
25065
25064
25063
y = -0.0501x + 25070
R² = 0.7538
y = 2E-11x + 25064
R² = #N/A
0.1
1
10
Factor de Carga (PROBING)
Factor de Carga (CHAINING)
Linear (Factor de Carga (PROBING))
Linear (Factor de Carga (CHAINING))

Maquina 2

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	25068.521	92.442
0.5	25068.521	91.346
0.7	25068.521	91.235
0.9	25068.521	96.032

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

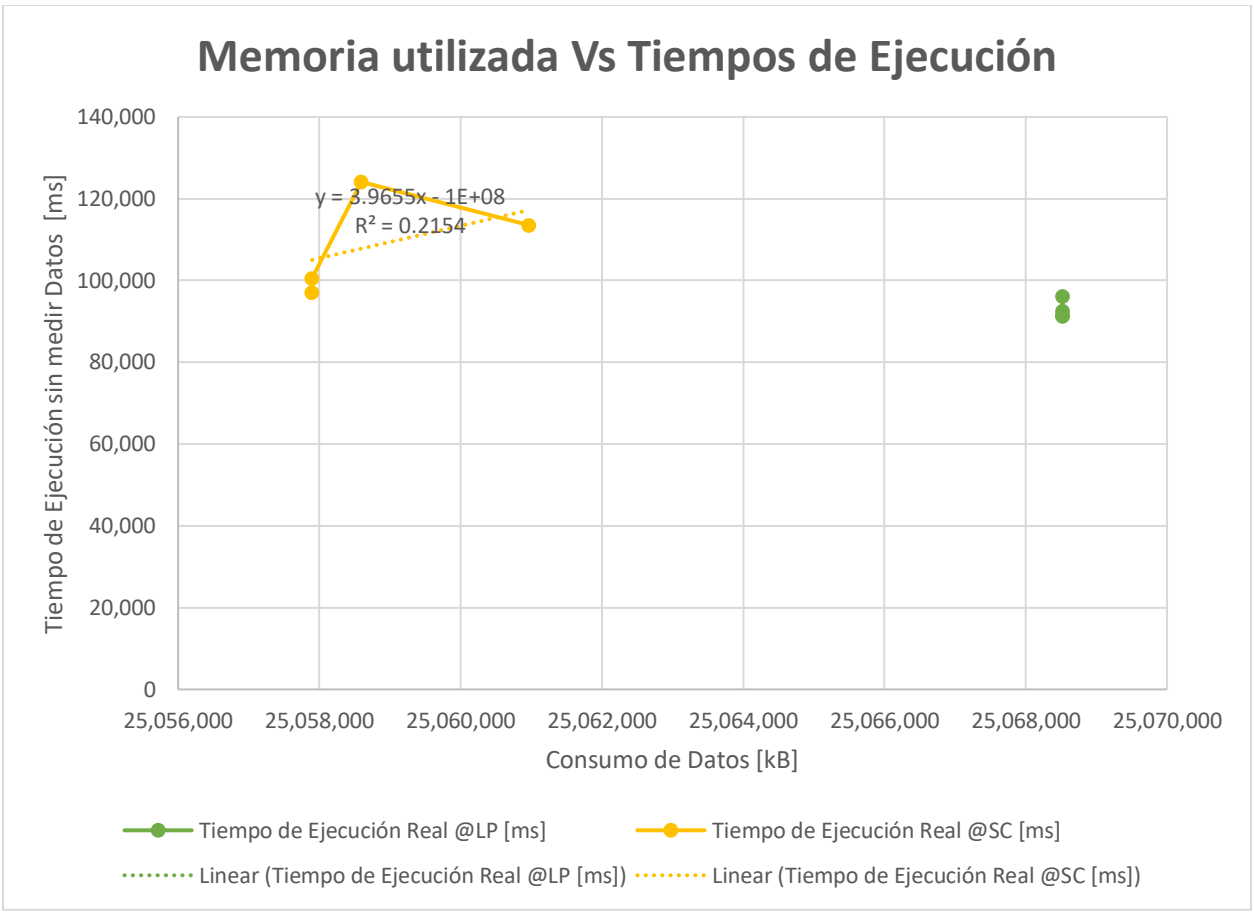
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	25060.968	113.521
4.00	25058.593	124.094
6.00	25057.897	100.484
8.00	25057.897	96.985

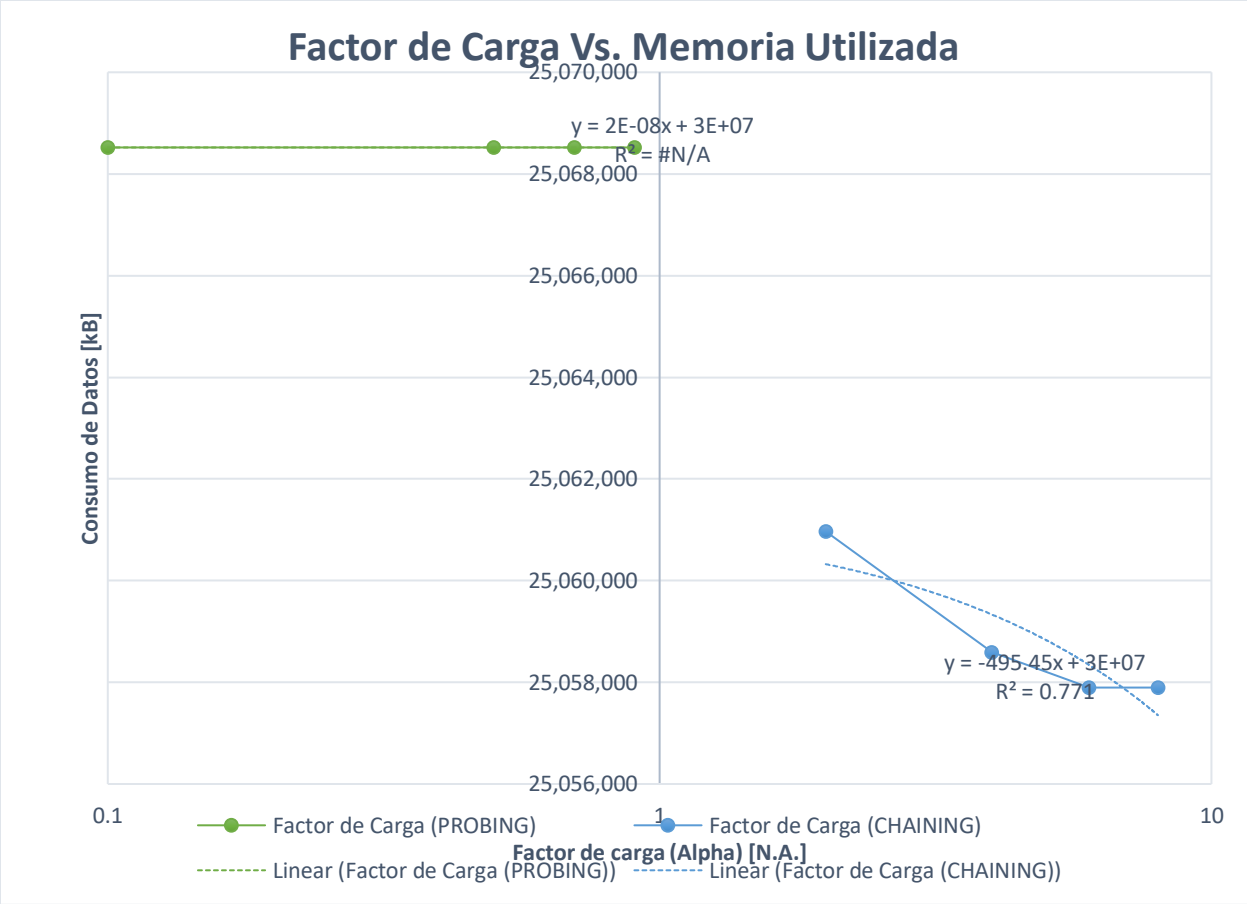
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

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


Memoria utilizada Vs Tiempos de Ejecución
Tiempo de Ejecución sin medir Datos [ms]
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
y = 3.9655x - 1E+08
R² = 0.2154
25,056,000
25,058,000
25,060,000
25,062,000
25,064,000
25,066,000
25,068,000
25,070,000
Consumo de Datos [kB]
Tiempo de Ejecución Real @LP [ms]
Tiempo de Ejecución Real @SC [ms]
Linear (Tiempo de Ejecución Real @LP [ms])
Linear (Tiempo de Ejecución Real @SC [ms])



Maquina 3

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	39350.146	580.661
0.5	39350.146	622.744
0.7	39350.146	641.175
0.9	39350.146	684.573

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

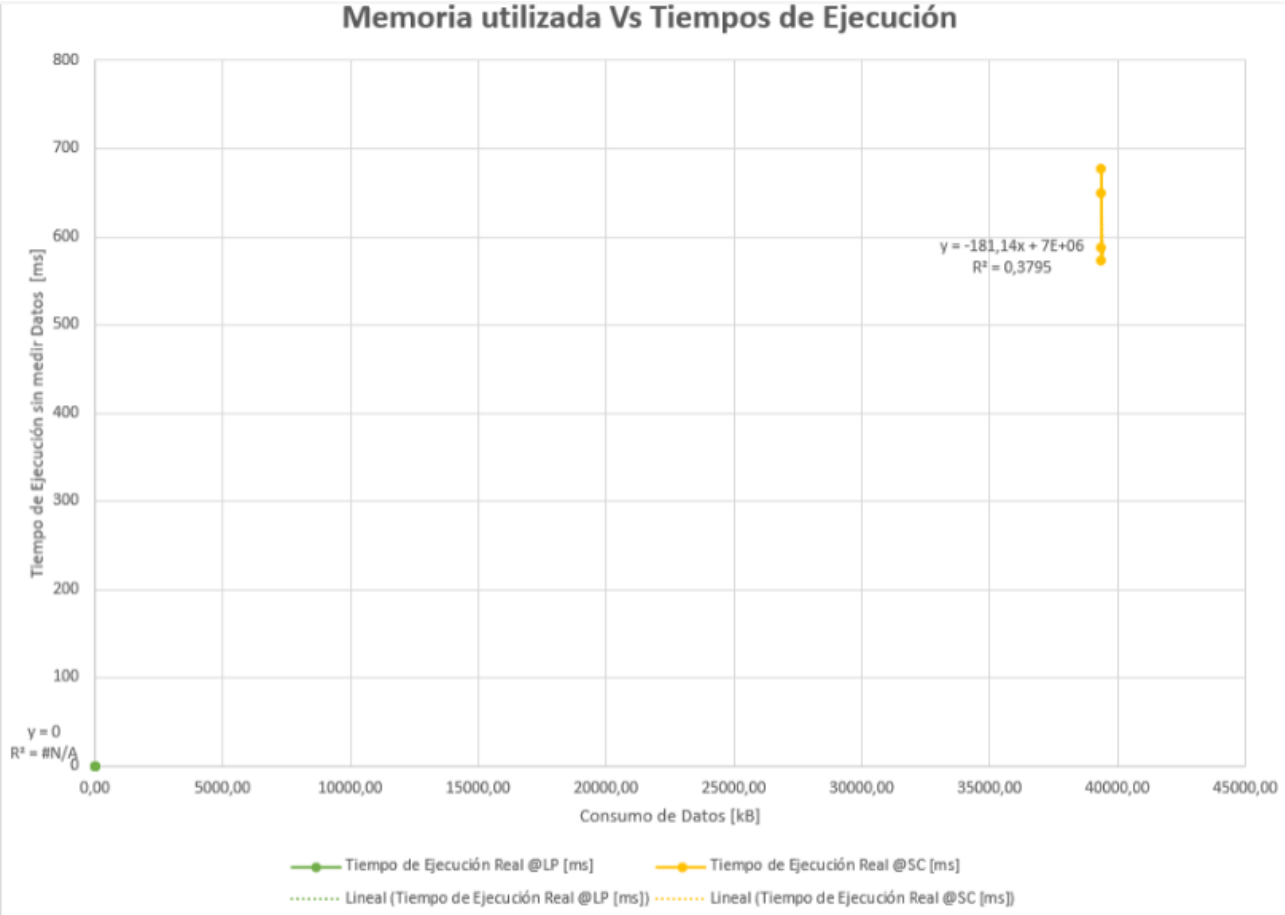
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	39355.061	588.313
4.00	39354.779	573.447
6.00	39354.709	648.938
8.00	39354.709	677.405

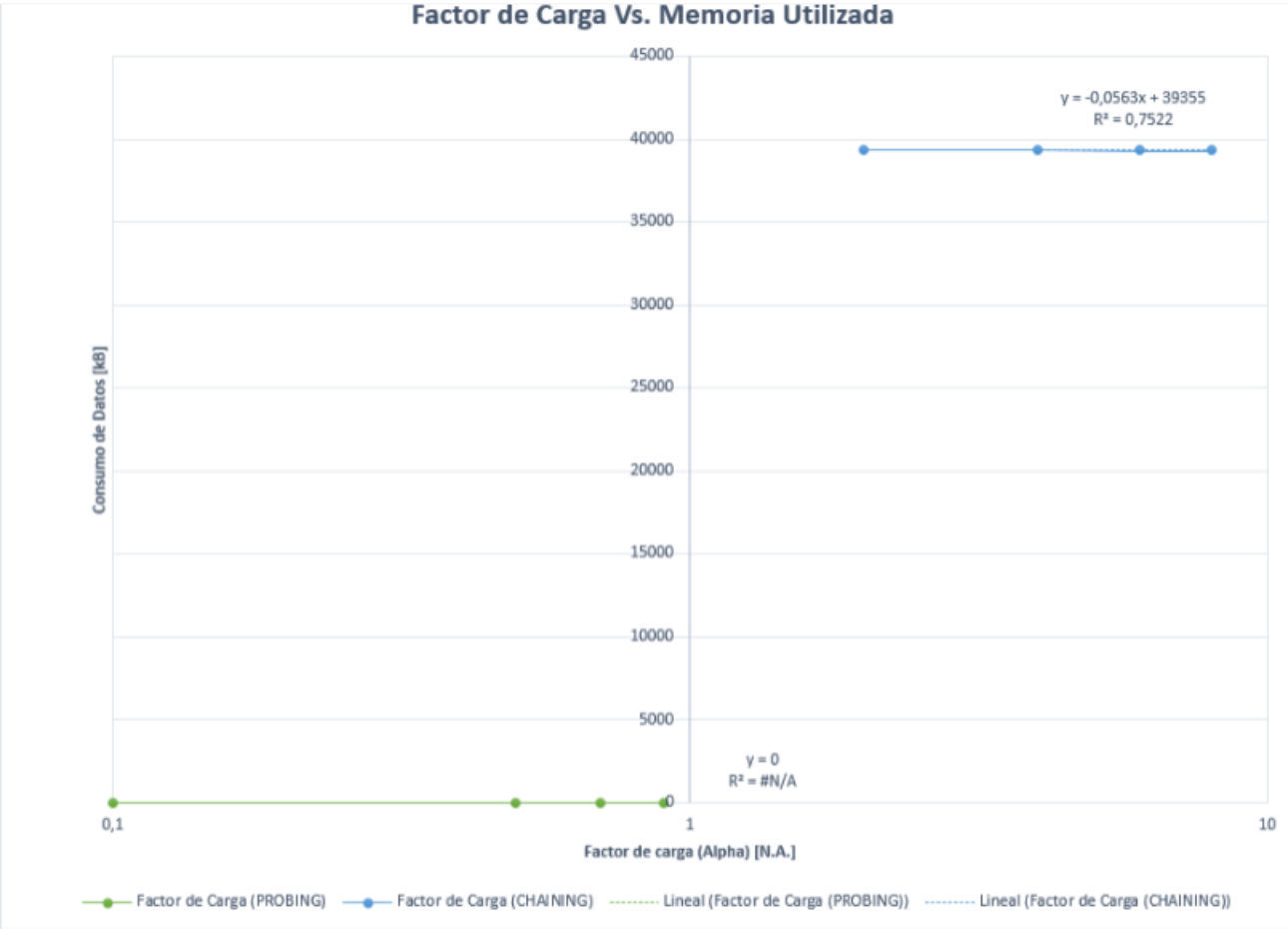
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

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




Factor de Carga Vs. Memoria Utilizada
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
Consumo de Datos [kB]
y = -0,0563x + 39355
R² = 0,7522
y = 0
R² = #N/A
0,1
1
10
Factor de carga (Alpha) [N.A.]
Factor de Carga (PROBING)
Factor de Carga (CHAINING)
Lineal (Factor de Carga (PROBING))
Lineal (Factor de Carga (CHAINING))

Preguntas de análisis

1. ¿Por qué en la función getTime() se utiliza time.perf_counter() en vez de otras funciones como time.process_time()?

La principal diferencia entre time.perf_counter() y otras funciones de medición de tiempo como time.process_time() es que esta proporciona mediciones precisas y confiables del tiempo total transcurrido en la tarea, independientemente de si la tarea está utilizando o no la CPU. Además, es una función que utiliza el reloj de alta resolución del sistema operativo, lo que la hace más precisa que otras funciones de medición, mientras que time.process_time() mide solo el tiempo de CPU utilizado por el proceso actual y puede no ser adecuado para medir el tiempo total de una tarea, ya que la tarea puede estar esperando una entrada/salida, por lo que no está utilizando la CPU.

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

Son importantes porque permiten la medición del uso de memoria, lo que es útil para identificar y solucionar problemas de rendimiento relacionados con la memoria.

La función start() inicia la medición del uso de memoria y la función stop() detiene la medición y devuelve un informe de las estadísticas de uso de memoria. El informe proporciona detalles sobre los objetos de Python que están utilizando la mayor cantidad de memoria.

3. ¿Por qué no se puede medir paralelamente el uso de memoria y el tiempo de ejecución de las operaciones?

No se puede medir paralelamente el uso de memoria y el tiempo de ejecución de las operaciones debido a que la medición de ambos aspectos requiere la recopilación de datos de diferentes fuentes y en diferentes momentos, medirlos paralelamente puede afectar los resultados de la medición. Por ejemplo, si se mide el tiempo de ejecución y el uso de memoria al mismo tiempo, la recopilación de datos para una métrica puede interferir con la recopilación de datos para la otra métrica, lo que puede llevar a resultados inexactos.

4. Teniendo en cuenta cada uno de los requerimientos del reto ¿Cuántos índices implementaría en el Reto? y ¿Por qué?
   En el reto se implementaría 3 índices principales, donde en 2 habría otro subíndice y en 1 un subíndice dentro de otro subíndice para realizar una carga y búsqueda veloces de datos dentro de los registros evitando aumentar la complejidad.
5. Según los índices propuestos ¿en qué caso usaría Linear Probing o Separate Chaining en estos índices? y ¿Por qué?
   Cuando se realice únicamente el índice de subsector económico se usaría el probing ya que sería una gran cantidad de subsectores mientras que para los años y sectores se usaría chaining ya que no se usaría tanto espacio para guardar una gran cantidad de posibles valores.
6. Dado el número de elementos de los archivos del reto (large), ¿Cuál sería el factor de carga para estos índices según su mecanismo de colisión?
   Para los probing se usaría un factor de carga de 0.5 para evitar gran cantidad de colisiones y para los chaining 2.00 ya que al existir pocos elementos se podrán incluir sin preocuparse por colisiones.
7. ¿Qué cambios percibe en el tiempo de ejecución al modificar el factor de carga máximo para cargar el catálogo de contenido Streaming?

Al modificar el factor de carga máximo se nota como va aumentando el tiempo de ejecución que es probablemente ocasionado ya que se aumenta la probabilidad de colisiones, lo que a su vez aumenta el tiempo necesario para buscar un elemento en la tabla hash y puede disminuir el rendimiento debido a la carga en la memoria caché.

8. ¿Qué cambios percibe en el consumo de memoria al modificar el factor de carga máximo para cargar el catálogo de contenido Streaming?

No se percibe ningún cambio en la memoria al aumentar el factor de carga, se mantiene constante.

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

Al modificar el esquema de colisiones, no se perciben cambios significativos en los tiempos de ejecución ya que no hay diferencias mayores a 10 ms entre cada esquema. Esto puede ser ocasionado a que si los elementos en la tabla hash están distribuidos uniformemente, tanto el linear probing como el separate chaining pueden minimizar el número de colisiones y, por lo tanto, el tiempo será similar en ambos casos. Otro caso puede ser si el tamaño de la tabla hash es suficientemente grande en comparación con el número de elementos almacenados, el número de colisiones será menor y, por lo tanto, el tiempo de búsqueda será similar en ambos casos.

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

Respecto al consumo de memoria se percibe que chaining maneja mayor memoria que el probing. Esto ocurre ya que el uso de chaining implica almacenar una lista enlazada de elementos en cada celda de la tabla hash que tiene una colisión. Esta lista puede crecer en tamaño a medida que se agregan más elementos con colisiones a la tabla, lo que puede aumentar el consumo de memoria.

11. ¿Qué configuración de ideal ADT Map escogería para el índice de años (“Año”) ?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.

Se usaría el mecanismo de chaining, un factor de carga de 2.0 y un número inicial de elementos de 20. Esto se debe a que tendría una ventaja en eficiencia ya que se reduce el tiempo de búsqueda y de inserción, ya que no es necesario recalcular la función hash para encontrar una nueva posición de la tabla. En su lugar, se puede simplemente agregar el elemento a la lista enlazada existente en esa posición. Y Un factor de carga de 2.0 significa que la tabla hash se llenará al 50% antes de que se tenga que realizar una reasignación de elementos. Un factor de carga de 2.0 proporciona un buen equilibrio entre el uso eficiente del espacio de la tabla hash y el rendimiento de la tabla.