

OBSERVACIONES DEL LA PRACTICA

Samuel Peña 202028273
Tomas Díaz 202220658
Manuel Pinzon 202125748

	Máquina 1	Máquina 2	Máquina 3
Procesadores	Intel(R) Core (TM) i7-55000 CPU @2.40GHz 2.40GHZ	AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz	12th Gen Intel(R) Core(TM) i9-12900H 2.50 GHz
Memoria RAM (GB)	8.0 GB	8 GB	32.0 GB
Sistema Operativo	Windows 10 Pro – 64 bits	Windows 11 – 64 bits	Windows 11 – 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 Real @LP [ms]
0.1	21820.23	82.02
0.5	21820.78	80.46
0.7	21818.19	79.13
0.9	21817.04	78.81

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	21816.89	81.42
4.00	21816.84	83.18
6.00	21816.80	81.40
8.00	21816.75	80.92

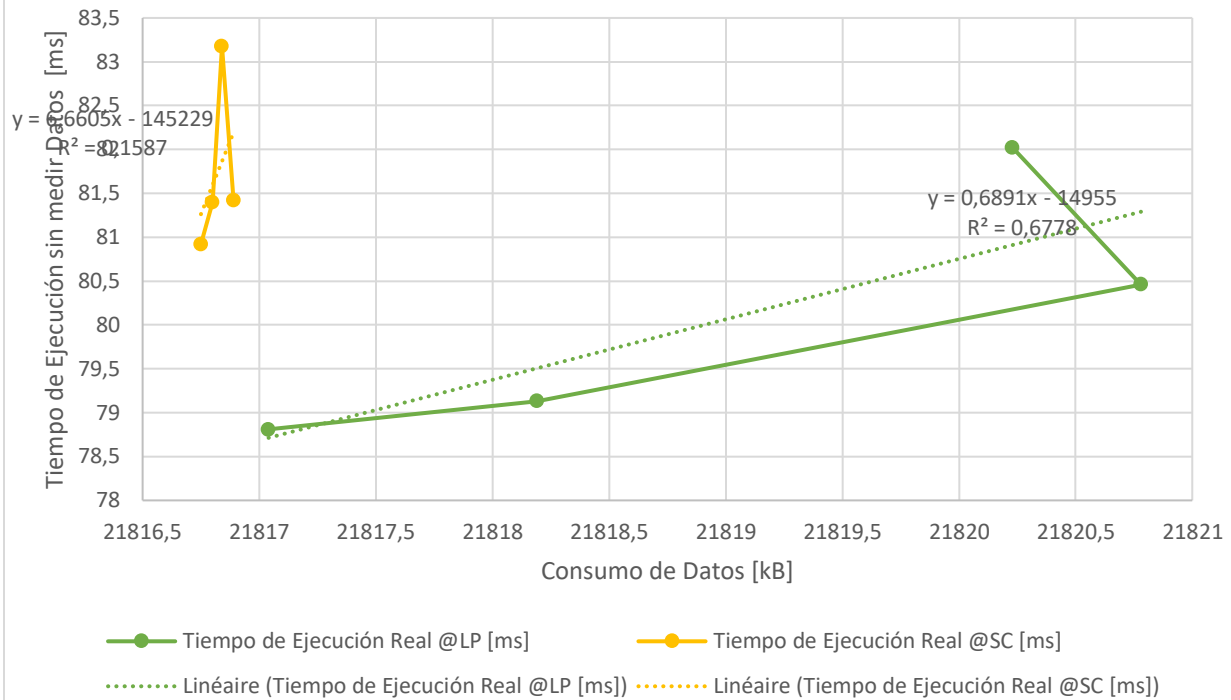
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

Graficas

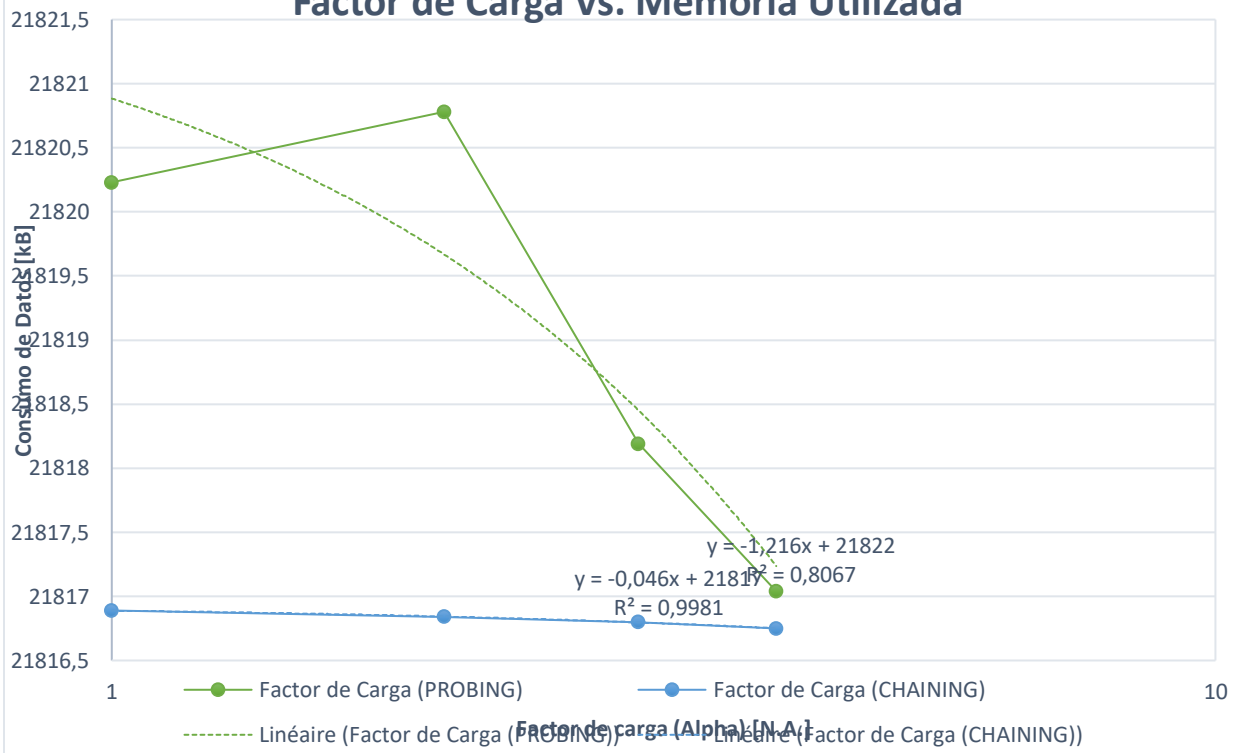
La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 1**.

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

Memoria utilizada Vs Tiempos de Ejecución



Factor de Carga Vs. Memoria Utilizada



Maquina 2

Resultados

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	21817,49	177,25
0.5	21825,34	199,38
0.7	21820,95	186,52
0.9	21817.09	179,48

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

Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	21810,70	183.46
4.00	21823,54	185.64
6.00	21820.80	194,13
8.00	21816.76	267.30

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

Maquina 3

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	21820.06	74.56
0.5	21820.10	77.77
0.7	21817.32	78.29
0.9	21816.27	82.03

Tabla 6. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 3.

Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	21816.89	74.50
4.00	21816.74	79.01
6.00	21816.36	82.54
8.00	21816.06	82.25

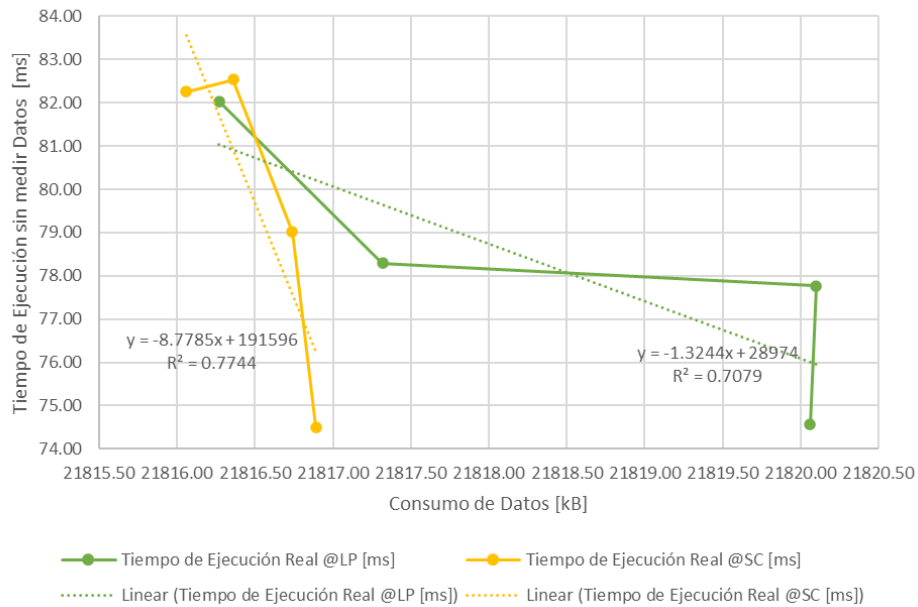
Tabla 7. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 3.

Graficas

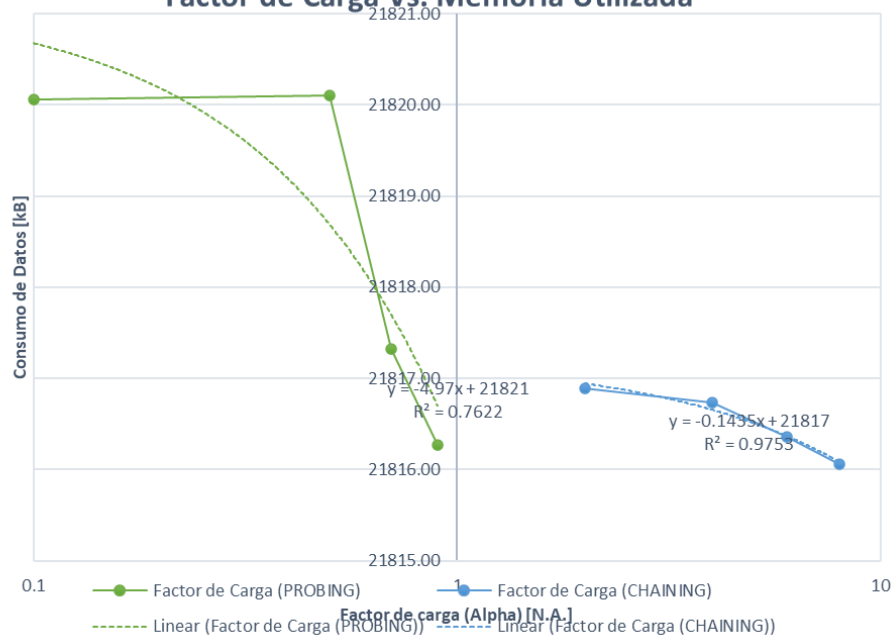
La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 3**.

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

Memoria utilizada Vs Tiempos de Ejecución



Factor de Carga Vs. Memoria Utilizada



Preguntas de análisis

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

Porque `perf_counter` funciona como un cronometro donde solo se toma el inicio y el fin, mientras que `process_time` suma el tiempo de las fracciones de todos los procesos, teniendo que tomar muchas mas mediciones. Al tomar mas mediciones es mas propenso a tener errores.

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

Para encontrar el delta entre dos estados del programa, el `start` establece el limite inferior y el `stop` el limite superior, si no hay `stop` el programa no pararia nunca.

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

Porque al incluir ambas mediciones paralelamente el tiempo tomado no corresponde solo a la carga de datos sino tambien lo que se demora el programa en calcular cuanta memoria ocupa.

4. Teniendo en cuenta cada uno de los requerimientos del reto ¿Cuántos índices implementaría en el Reto? y ¿Por qué?

Los indices depende de la estructura de Hash que se aplique a cada uno de los requerimientos, el numero de indices es dado por $M/fc - 1$ (donde M es el numero de datos y fc el factor de carga). Podemos asumir que el numero de indices estara en un intervalo entre $[409, 3719]$ y $[4546, 40919]$. Ya que el numero de indices depende del factor de carga que se aplique, entonces no se puede establecer un numero arbitrario sin antes saber en que caso se usara Linear Probing o Separate Chaining y con que factor de carga se hara cada uno.

5. Según los índices propuestos ¿en qué caso usaría **Linear Probing** o **Separate Chaining** en estos índices? y ¿Por qué?

Dado los indices propuestos, podemos suponer que cuando el numero de indices sea inferior a 3719 usaremos Separate Chaining y cuando el numero de indices sea superior a 4546 se usara Linear Probing. Esto se debe a que Separate Chaining solo puede tener un factor superior a 1 y para Linear Probing el factor de carga tiene que ser inferior a 1.

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?

Dado que son 4903 elementos, el factor de carga para estos indices debe ser maximo de 0.9 si se usa el mecanismo de Linear Probing y de minimo de 1.1 si usa el mecanismo de Separate Chaining

7. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de contenido Streaming?

El tiempo de ejecucion tiene una correlacion positiva con el factor de carga, esto se ve ya que a medida que el factor de carga aumenta, tambien aumenta el tiempo de ejecucion.

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

En este caso percibimos una correlacion negativa, esto implica que a medida que aumenta el factor de carga, el consumo de memoria disminuye.

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.

Por el numero de elementos que manejamos, no conseguimos percibir algun tipo de cambio en los tiempos de ambos esquemas de colisiones

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.

En este caso, vemos que el esquema de Linear Probing acaba ocupando más memoria que el esquema de Separate Chaining. Esto se debe a que el esquema de Linear Probing tiene un factor de carga inferior a 1, esto implica que tiene que haber mas indices que elementos. Al contrario, Separate Chaining cuenta con un factor de carga superior a 1, esto quiere decir que hay más de un elemento por indice, asi hay menos indices.

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.

Para esta configuracion seria mas apropiado usar una mecanismo de Linear Probing, con un factor de carga de 0.5 y con 20 elementos