

OBSERVACIONES DEL LA PRACTICA

Miguel Perdomo - 201913791
Samuel Jaramillo - 202010768
Luisa Zambrano - 201914911

	Máquina 1	Máquina 2	Máquina 3
Procesadores	Interl core i7 9750H @2.6Ghz	1,7 GHz Intel Core i7 de dos núcleos	Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
Memoria RAM (GB)	8GB 2667MHz	8 GB 1600 MHz DDR3	8.00 GB
Sistema Operativo	Windows 10 64 bits	MacOS Big Sur	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	25076,60	107,38
0.5	25076,59	104,35
0.7	25076,96	105,01
0.9	25077,02	99,52

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	25070,74	107,93
4.00	25068,17	103,33
6.00	25062,79	102,28
8.00	25062,04	106,812

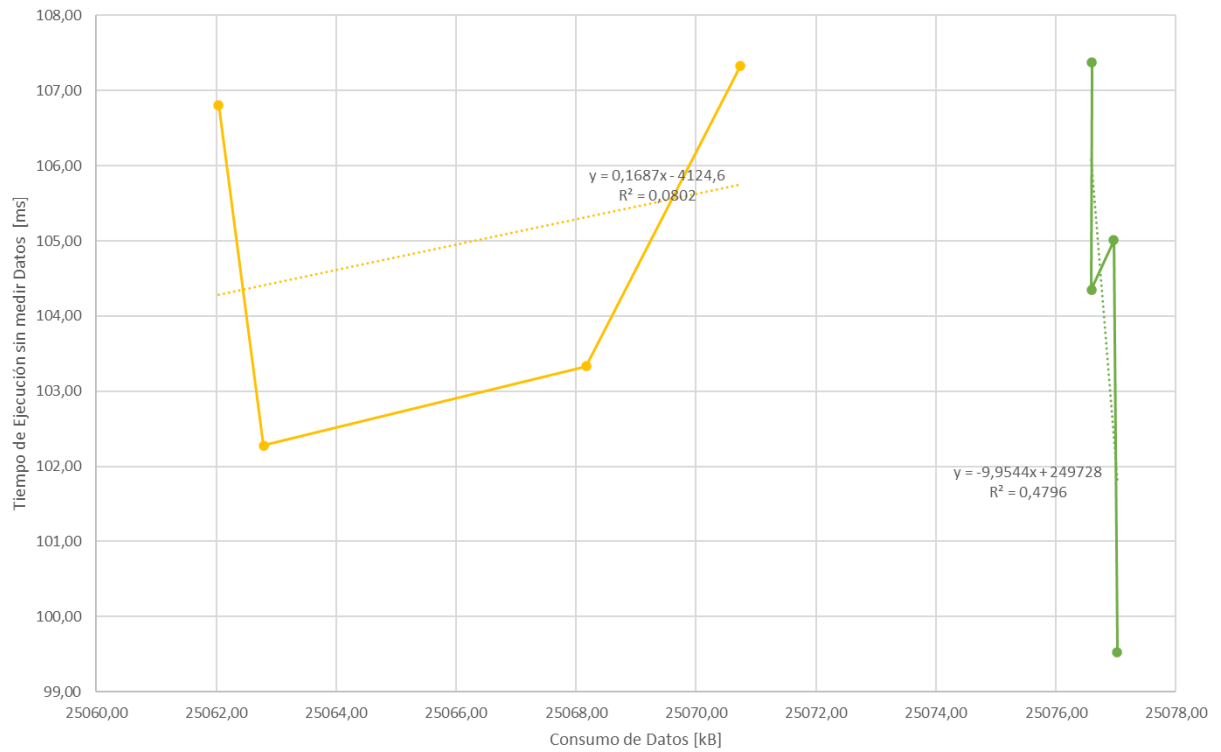
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

Graficas

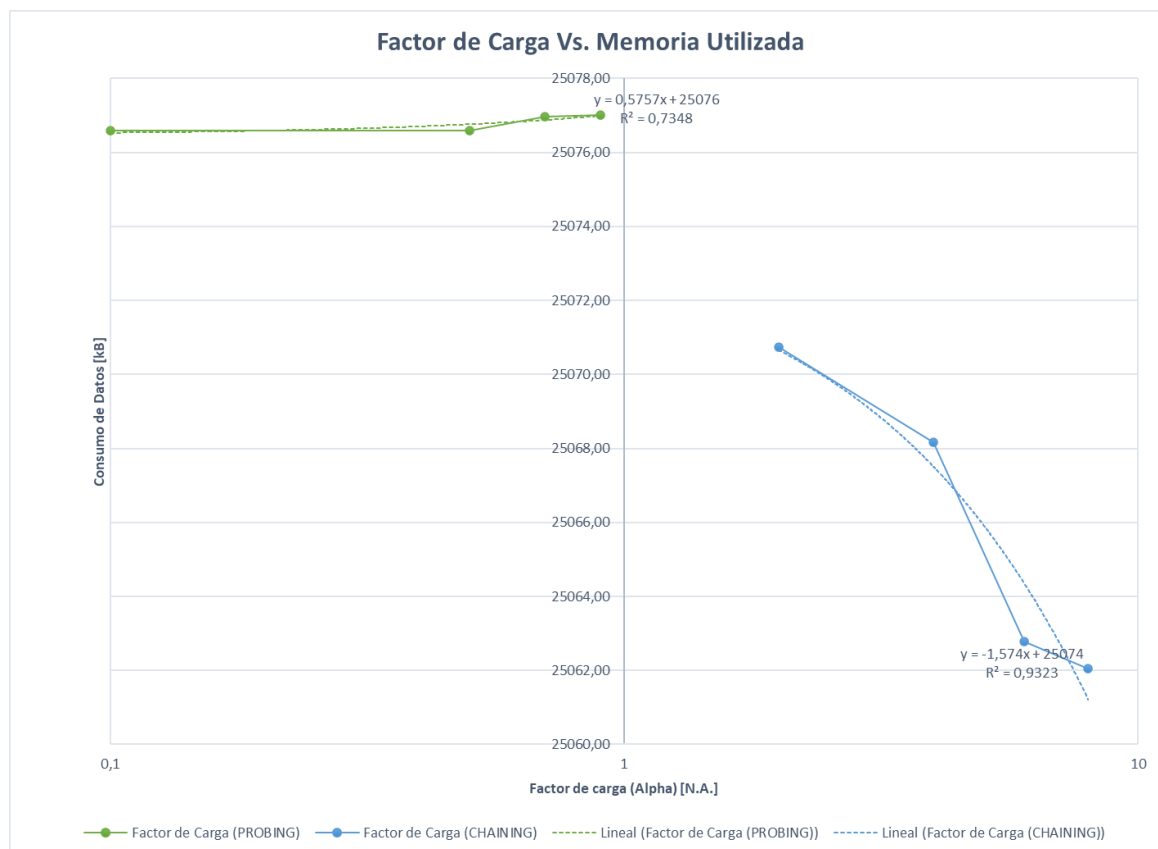
La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 1**.

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

Memoria utilizada Vs Tiempos de Ejecución



—●— Tiempo de Ejecución Real @LP [ms] —●— Tiempo de Ejecución Real @SC [ms]
..... Lineal (Tiempo de Ejecución Real @LP [ms]) Lineal (Tiempo de Ejecución Real @SC [ms])



Maquina 2

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	25069.14	144.21
0.5	25064.12	135.73
0.7	25062.21	131.37
0.9	25062.21	133.56

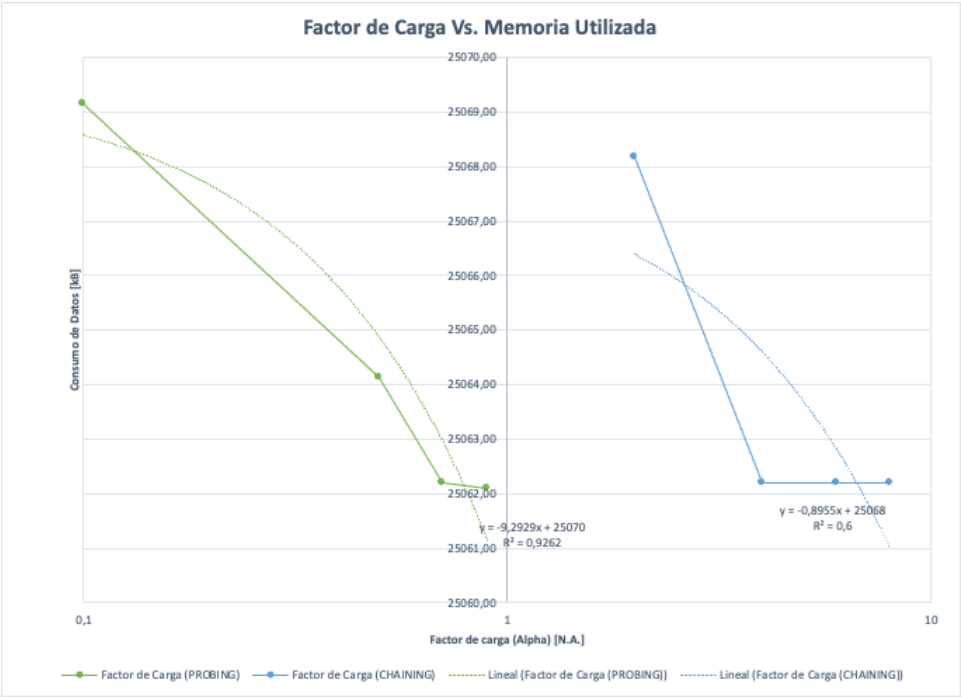
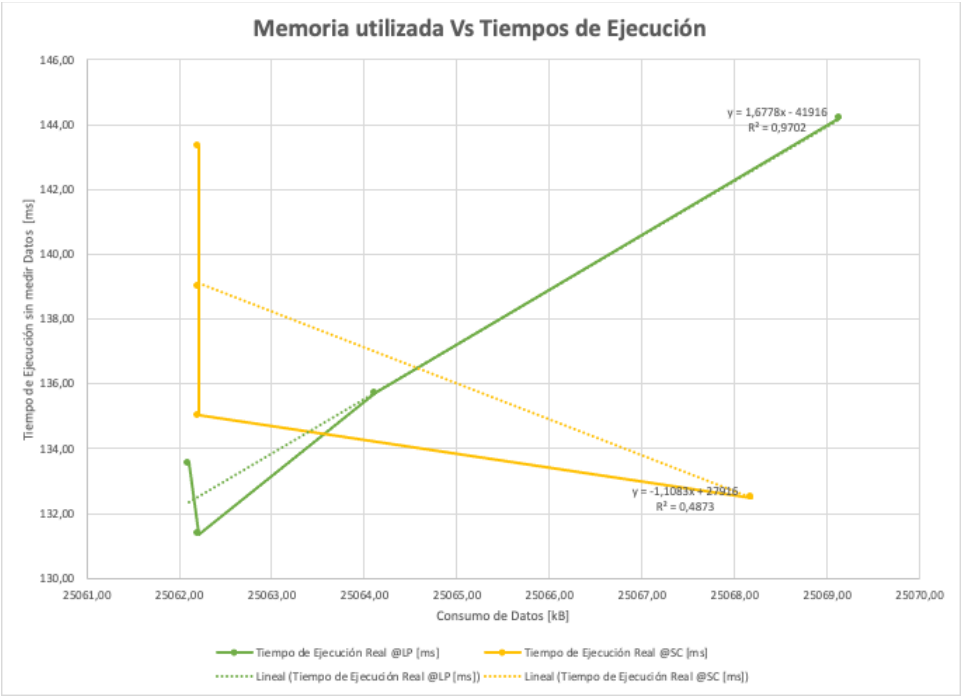
Tabla 4. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 2.

Carga de Catálogo CHAINING

Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	25068.18	132.50
4.00	25062.21	135.02
6.00	25062.21	139.00
8.00	25062.21	143.33

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



Maquina 3

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @LP [ms]
0.1	25064.30	86.36
0.5	25063.13	90.42
0.7	25063.07	93.63
0.9	25063.00	93.79

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

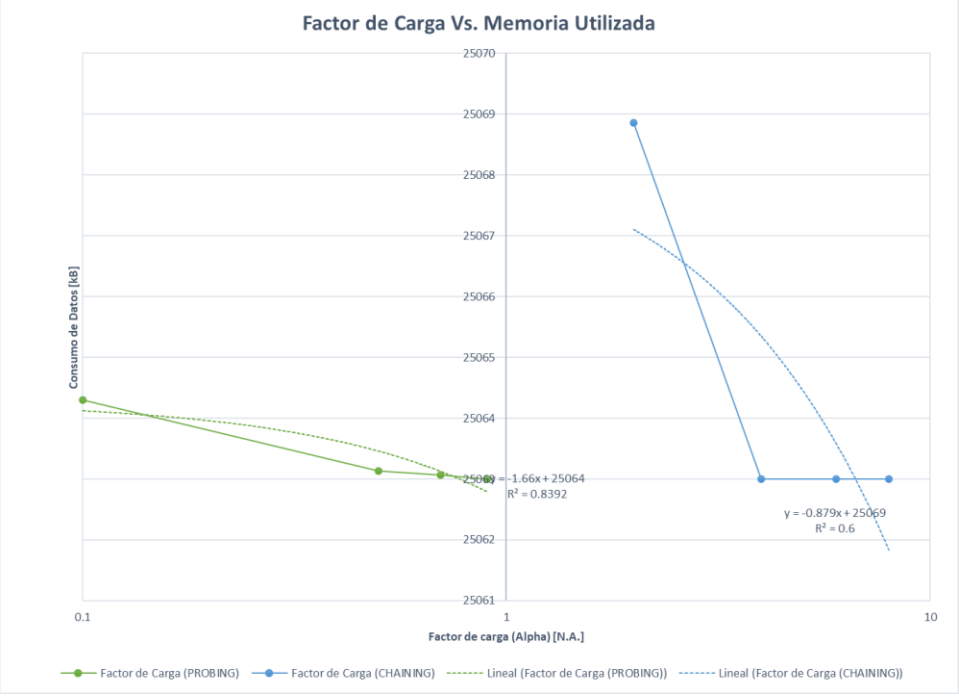
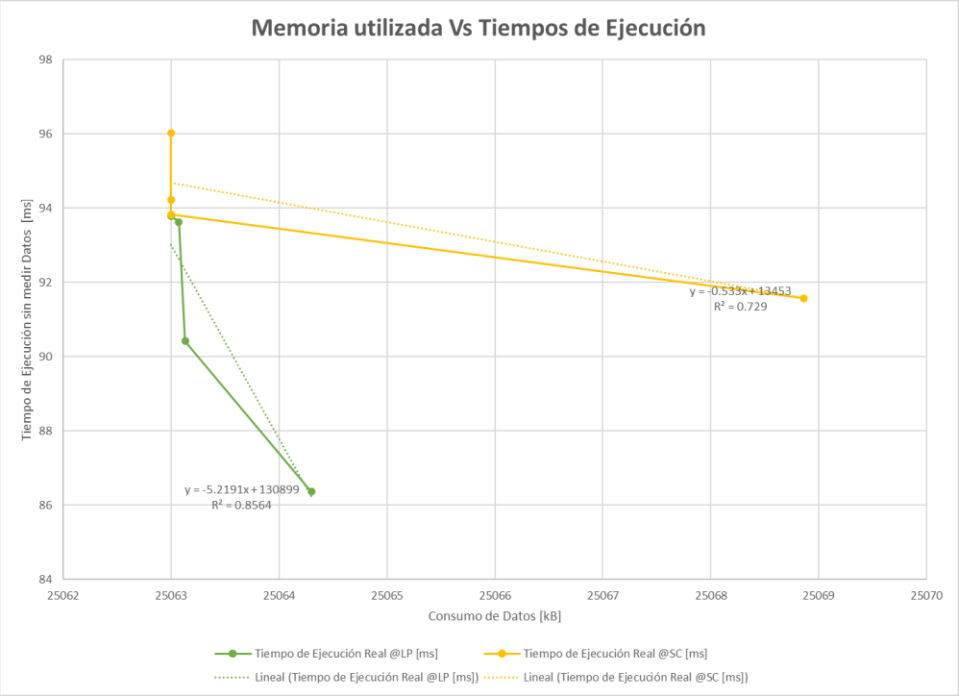
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución Real @SC [ms]
2.00	25068.86	91.57
4.00	25063.00	93.83
6.00	25063.00	94.23
8.00	25063.00	96.02

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.

Grafica

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



Preguntas de análisis

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

En nuestro caso no utilizamos la función **time.process_time()** debido a que esta no nos daría el tiempo real de ejecución. Lo anterior se da porque esta función solo mide el tiempo en el que los hilos del disco duro están activos y no toma el tiempo *muerto* entre el cambio de un hilo a otro. Por lo tanto, al tiempo final se le restaría el tiempo muerto y no se obtendría el tiempo real experimentado por el usuario. **time.perf_counter()**, se utiliza porque sí toma este tiempo muerto y nos da, en consecuencia, el tiempo real experimentado por el usuario.

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

Estas funciones son de importancia debido a que activan o desactivan la actividad de la librería **tracemalloc**. Si no se iniciara con **start()**, esta no sabría que está activa y no la podríamos utilizar. A su vez, si no se cerrara con **stop()**, esta seguiría activa indefinidamente, ocupando así espacio en nuestra memoria y saturando el computador.

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

Ambas cantidades no se pueden medir paralelamente porque la medición de una afecta la veracidad del resultado de la otra. Si se mide el tiempo y la memoria a la vez, el tiempo que se medirá se verá afectado por la medición de memoria y arrojará un tiempo mucho mayor al real debido a que el proceso de medir memoria se demora. Además, el resultado de la memoria se verá afectado porque la medición de tiempo invoca librerías que ocupan espacio en la memoria, afectando el valor real de la memoria utilizada. Es por esto que si se quiere obtener valores reales de las cantidades a medir, se deben medir ambas por separado.

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

Viendo los requerimientos del reto, se podría decir preliminarmente que se implementarían tres índices. Esto se debe a que todos los requerimientos nos piden ya sea información de un año en específico, un sector en específico, un subsector en específico o alguna combinación de las tres. En consecuencia, organizar la información bajo estas tres categorías simplificaría el trabajo de búsqueda y filtrado de la información.

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

Linear Probing se utilizaríamos para el índice de años. Esto se debe a que su factor de carga es mucho menor al Separate Chaining y esto es conveniente si queremos manejar cantidades de datos pequeñas. En el reto la cantidad de años es fija y pequeña, luego, no se tendría que hacer rehash utilizando Linear Probing y se ahorraría espacio en la memoria. Para los otros índices utilizaríamos Separate Chaining debido que son cantidades mayores y para manejarlas es más económico (en términos de memoria) utilizar un factor de carga mayor para evitar hacer rehash.

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?

En ambos casos (Probing y Chaining), las ta que los valores máximos de factor de carga (0.9 en el primero y 8 en el segundo) fueron los que ocuparon menos espacio en memoria. Además, la diferencia en tiempo de ejecución con respecto al menor factor de carga fue de menos de 10 ms. En este sentido, escogeríamos el factor de carga máximo para ambos tipos de dato debido a la poca memoria que se utiliza en ambos casos.

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

A medida que el fator de carga se acerca a 1, disminuye el tiempo de ejecución.

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

A medida que el factor de carga maximo para culaquier caso se acerca a 1, el consumo de datos aumenta, mientras que al alejar el factor de carda de 1, disminuye el consumo de datos.

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.

A cada estudiante las gráficas de PROBING dio diferentes resultados, lo que hace más difícil hacer un análisis sobre cómo varía el tiempo de ejecución. Sin embargo, con CHAINING hay un resultado homogéneo que es que a mayor memoria consumida, menor es el tiempo de ejecución.

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.

Una vez más, a cada estudiante las gráficas de PROBING dio diferentes resultados, lo que hace difícil hacer un análisis sobre cómo varía el tiempo de ejecución, pero teniendo en cuenta las gráficas de la máquina 2 y 3 se ve que a mayor factor de carga, menor es el consumo en memoria. No obstante, en dos casos se ve que el consumo de memoria se estabiliza hasta volverse una constante.

. Para CHAINING se ve que a mayor factor de carga, menor es el consumo en 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.

El ideal es CHANING ya que en las gráficas se ve que después de con un factor de carga alto (ej: 8) y un numero inicial de elementos que sea igual al numero de años.