OBSERVACIONES DEL LA PRÁCTICA

Juan David Salguero 201923136
David Molina 202125176

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 7	Intel Core i5
Memoria RAM (GB)	8GB	8GB
Sistema Operativo	Windows 11- 64 bits	macOs Catalina - 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]
0.10	1457370.668	85257.201
0.50	856225.733	50135.425
0.90	789464.280	50432.964

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

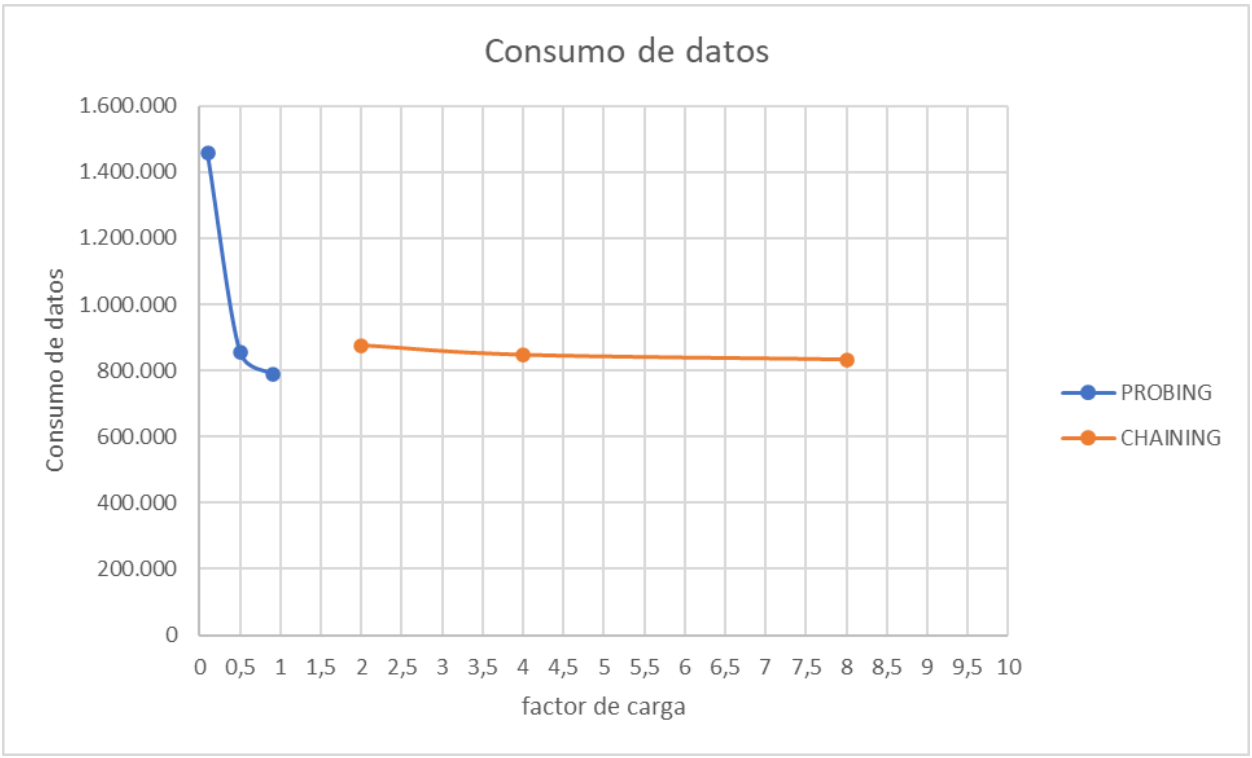
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	876375.843	68776.584
4.00	847585.280	56537.031
8.00	833198.241	59297.024

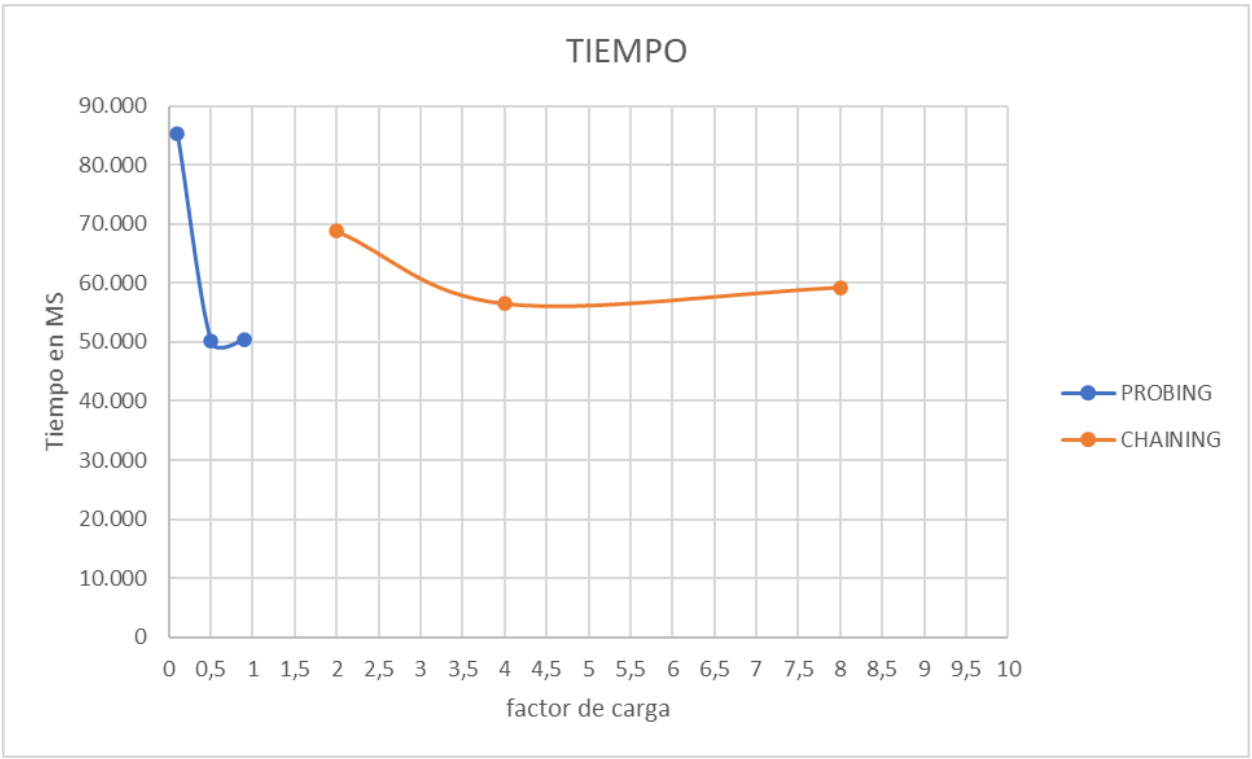
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

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]**
0.10	1457383,863	69520,902
0.50	856226,598	39230,092
0.90	789465,980	44370,126

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

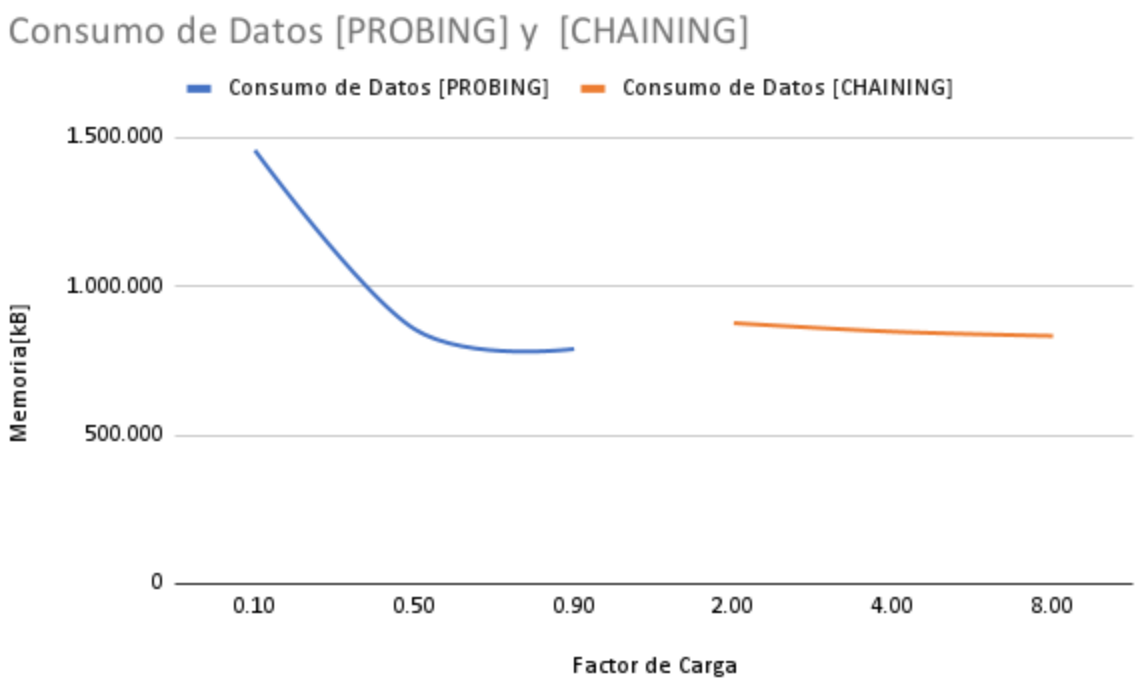
Factor de Carga (CHAINING)	**Consumo de Datos [kB]**	**Tiempo de Ejecución [ms]**
2.00	876395,810	53444,922
4.00	847605,677	44951,452
8.00	833218,231	41415,437

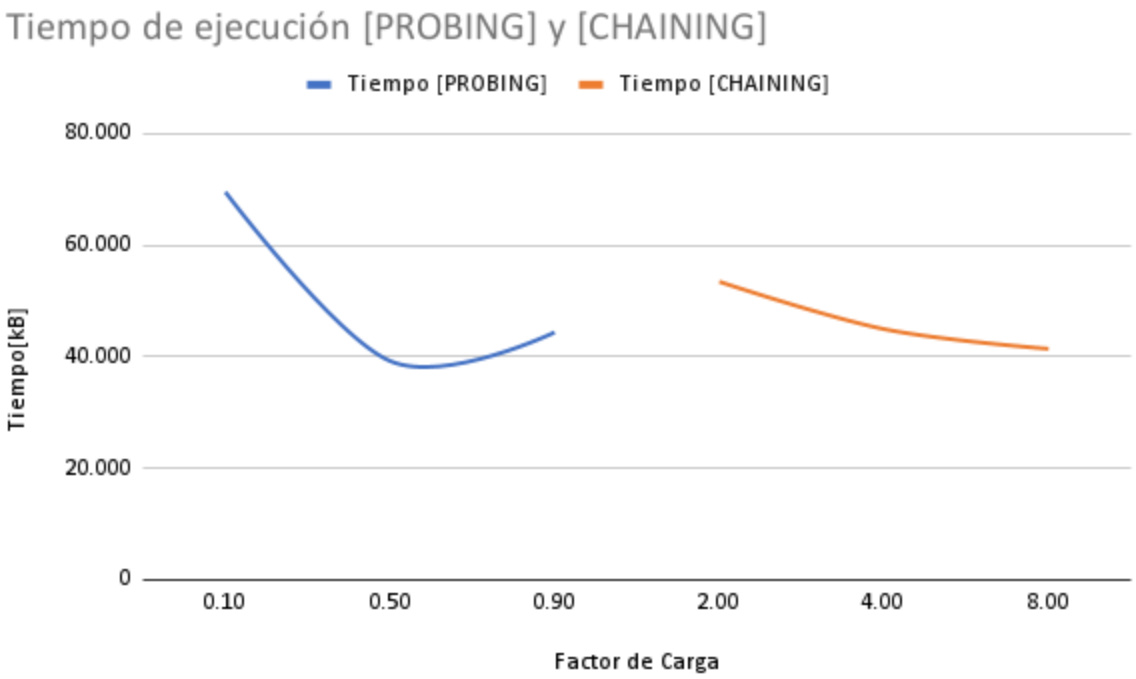
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

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





Preguntas de análisis

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

process_time() se encarga de retornar un float correspondiente al tiempo transcurrido mientras que el cpu del computador se ocupa de una función en específica. Para propósitos de esta actividad, y para obtener un tiempo más exacto con el transcurrido en la realidad, getTime() usa perf_counter() para así obtener el tiempo transcurrido de la carga de datos como si fuera un cronómetro y no un medidor de tiempo para el cpu.

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

La función start() le indica a tracemalloc cuando debería empezar a medir las diferencias en la memoria. La función stop() detiene los cálculos de memoria realizados por tracemalloc. Con estas dos funciones le podemos indicar al programa el intervalo en el que debería medir una diferencia del uso de memoria.

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

Los artistas, tracks y álbumes se guardarán en 2 índices, cada uno. Se opina que organizar cada artista y track por índices de popularidad sería beneficioso para los req. 2 y 3. Por su parte los álbumes por índices de año, sería lo más efectivo para buscar valores en específico, sea un top de popularidad o un año en particular. Por otra parte, el segundo índice de cada tipo guarda cada elemento de los tres tipos de datos mencionados por su id, ya que varios elementos, sean artistas o tracks, se conectan unos con los otros a través de sus ids.

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

Los índices de popularidad y año serán del tipo CHAINING, ya que varios valores contienen un mismo o ranking. Por otra parte, el índice basado en id será del tipo PROBING, esto se debe a que se entiende que el id de cada elemento es único, por ello, cada elemento debería tener un posición diferente en la tabla de hash.

5) Dado el número de elementos de los archivos Spotify (large), ¿Cuál sería el factor de carga para estos índices según su mecanismo de colisión?

Los índices de popularidad y año tienen que, por naturaleza, guardar varios artistas o álbumes en una sola llave. Lo ideal sería crear una lista como valor de la llave (ranking o año) que guarde cada elemento correspondiente a su clasificación, para así no tener un factor de carga mayor a 4. Por otra parte, el índice de datos organizado por el id de cada elemento tendrá un índice de carga de 0.5.

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

No parece haber un patrón en el tiempo de ejecución respecto a su esquema de colisiones. Lo único remarcable es el hecho de que un factor de carga muy pequeño como 0.1 tiene un tiempo de ejecución mayor al resto. No obstante, la diferencia de tiempo en factores de carga mayores o 01 no es constante.

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

En general, se percibe que entre menor sea el factor de carga más espacio se utiliza en la máquina. No obstante, cabe remarcar que la diferencia de kB entre factores de carga altos no es mucha.

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

No parece haber un patrón en el tiempo de ejecución respecto a su esquema de colisiones. En los dos tipos de esquemas se mantiene un tiempo casi promedio respecto al tamaño de datos. El único pico de tiempo se nota cuando se mantiene un factor de carga muy pequeño en PROBING; por su parte, un esquema CHAINING parece ser más rápido entre mayor sea su factor de carga.

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

No parece haber un patrón en el consumo de energía respecto a su esquema de colisiones, en general se mantiene casi constante según el tamaño de los datos en CHAINING. Por otra parte, el uso de la memoria parece aumentar entre más pequeño sea el factor de carga en el tipo PROBING.

10) ¿Qué configuración ideal de ADT Map escogería para el **índice géneros musicales**?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.

Suponiendo que la variedad de géneros no supere a 80, y teniendo en cuenta un factor de carga de 0.5 con un esquema de tipo PROBING para evitar colisiones, el número inicial de elementos sería un poco superior a 160.