

OBSERVACIONES DEL LA PRACTICA

Estudiante 1: José Vicente Vargas Panesso Cod 201815601

Estudiante 2: Daniel Reales Cod 201822265

	Máquina 1	Máquina 2
Procesadores	Intel core i7-7700HQ	2.5 GHz Dual-Core Intel Core i5
Memoria RAM (GB)	16 GB	4 GB
Sistema Operativo	Windows 10	macOS Catalina

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]
0.30	1455695.17	48040.6121
0.50	1455668.73	45381.6038
0.80	1455668.49	45774.021

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

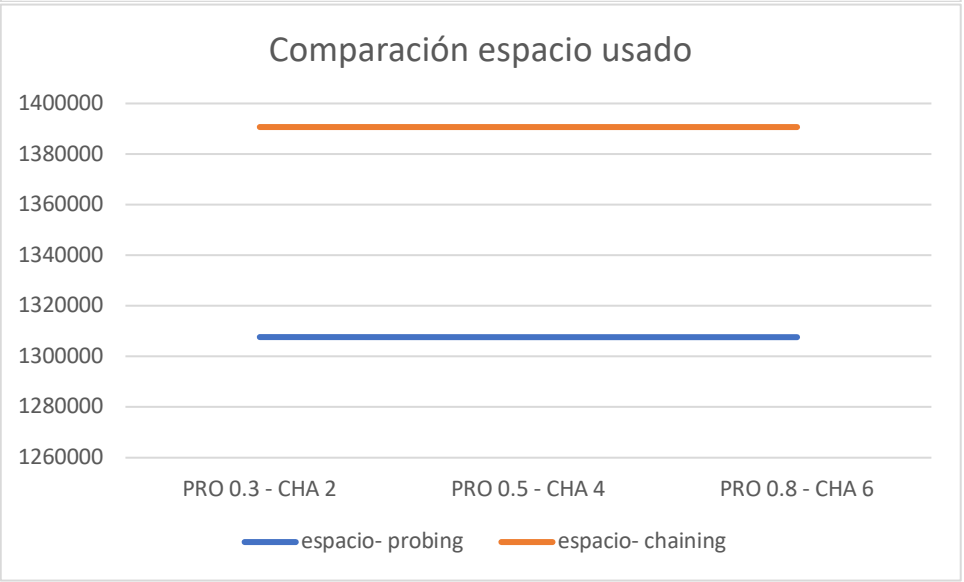
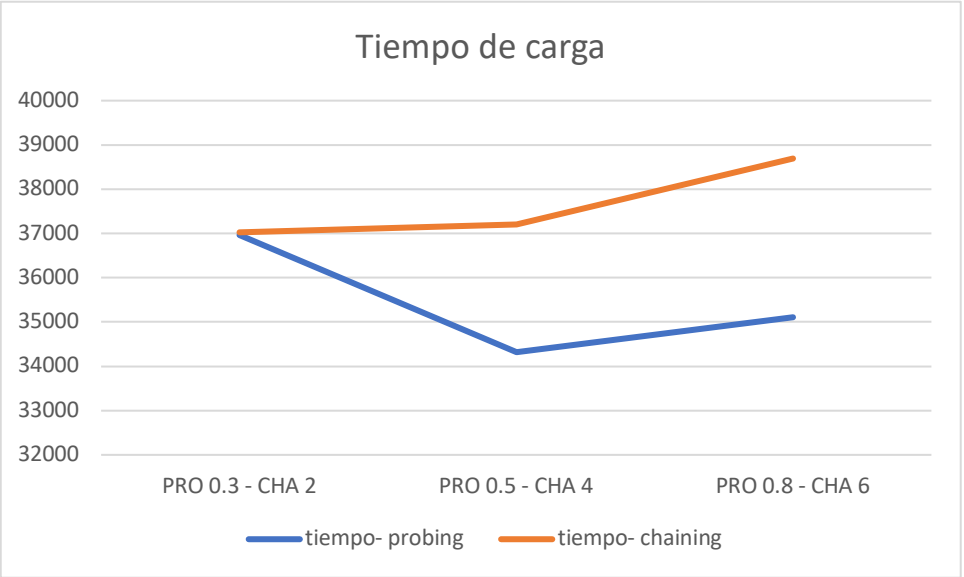
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	1455683.28	46725.2592
4.00	1455678.45	47396.3605
6.00	1455678.34	48909.3766

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

<u>Carga de Catálogo PROBING</u>		
Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	1555518.873	87831.77741
0.50	1455639.014	73280.09651
0.80	1399191.537	78700.6034

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	1448169.168	79812.70779
4.00	1419409.834	78522.68575
6.00	1409792.178	80222.24147

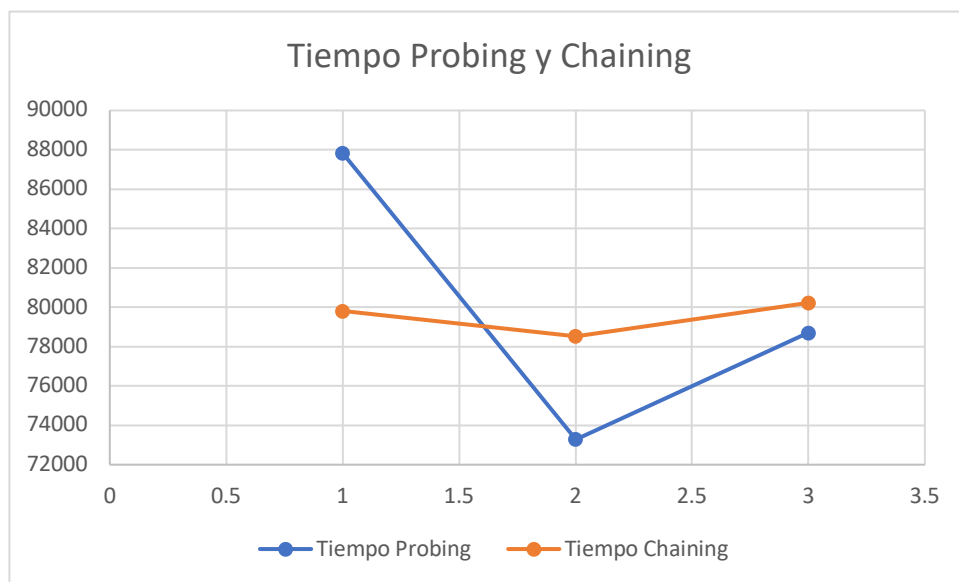
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

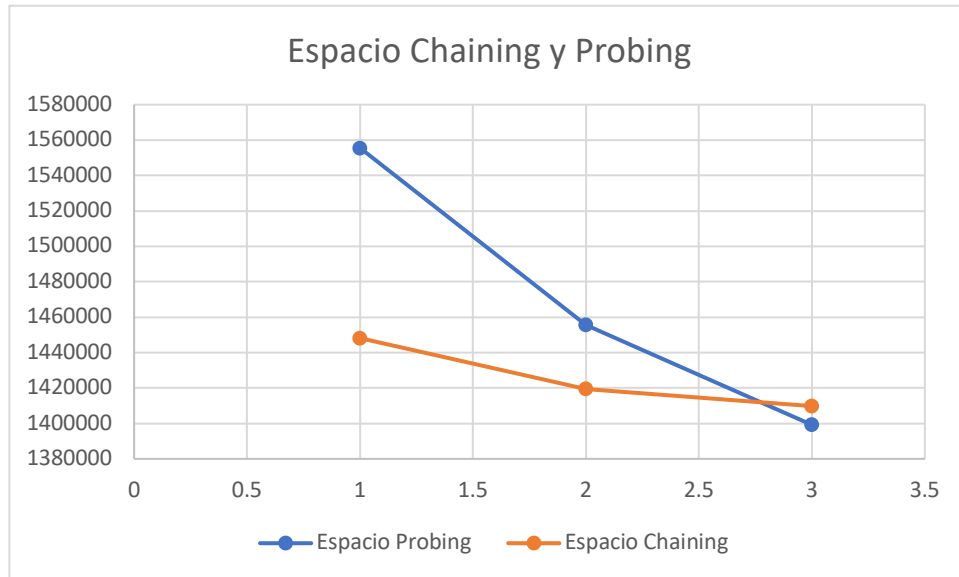
Graficas

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

Nota importante: El eje horizontal no corresponde con valores reales. En cambio, el primer punto de la gráfica de izquierda a derecha corresponde al factor de carga de 0.3 y 2 para probing y chaining, respectivamente. De manera similar, el punto medio corresponde a factores de carga de 0.5 y 4 y el último a factores de 0.8 y de 6.





Preguntas de análisis

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

La función `perf_counter()` tiene en cuenta el tiempo gastado durante el sleep, mientras que `process_time()` no tienen en cuenta este tiempo. Las CPUs de los computadores realizan varios procesos secuencialmente, por lo cual solo una parte parcial del tiempo total gastado corresponde efectivamente a un proceso, y es este tiempo parcial el que se cuenta en la función `process_time()`, mientras que el tiempo total teniendo en cuenta las demás tareas es el que monitorea la función `perf_counter()`.

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

Acorde a la documentación docs.python.org la función `tracemalloc` es utilizada para realizar un seguimiento de los bloques de memoria que aloca el programa. Para iniciar el proceso de seguimiento de memoria se debe llamar al método `start()`. En el contexto del rastreo que se busca realizar en el laboratorio en cuestión, es fundamental determinar el uso de memoria específico a cada operación. Para controlar cuando inicia este proceso es necesario llamar a este método. Análogamente, el método `stop()` finaliza el proceso de rastreo de las alocações de memoria. Adicionalmente, al invocarlo Python borra todos los trazos de bloques de memoria que allocó anteriormente. Este método es clave en el contexto del reto para adecuadamente dar fin al seguimiento de la memoria utilizada por la función a la que se le pretende realizar el seguimiento.

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

Al aumentar el factor de carga usados se observa un aumento en el tiempo de ejecución para ambos manejos de colisiones implementados. De todas formas es importante resaltar que para el caso de lineal probing el mejor tiempo de ejecución se da con un factor de carga de 0.5, ya que tanto para los factores de 0.3 y 0.8 se presentan peores tiempo de carga.

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

Para ambos manejos de colisiones se presenta un aumento de memoria usada. Es importante resaltar que siempre se presenta un menor uso de memoria en el caso de Chaining, esto se podría explicar por el espacio ocupado por la lista en cada una de las llaves.

- 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.

En los resultados se evidencia que el tiempo de ejecución es menor para el esquema de colisiones probing. Esto se debe debido a que, en promedio, el tiempo que toma el recorrido de las listas en el esquema chaining es mayor que el recorrido bajo el esquema probing. Es importante notar, no obstante, que se puede evidenciar un tiempo pico para el factor de carga 0.3 para el esquema PROBING. No obstante, atribuimos este tiempo de carga alto no a un alto nivel de colisiones sino a un posible rehash.

- 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.

Cuando se modifica el esquema de colisiones se evidencia una disminución sustancial al pasar de PROBING a CHAINING. Eso se debe a que el uso de listas para almacenar los elementos en el caso de colisiones bajo Chaining, reduce sustancialmente el espacio necesario para almacenar una determinada cantidad de información.