

OBSERVACIONES DEL LA PRACTICA

Juan Camilo Colmenares Ortiz - 202011866
Juan Andrés Ospina Sabogal - 202021727

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 7 4800H 2.90 GHz	Intel® Core™ i5-7267U @3.1GHz
Memoria RAM (GB)	16 Gb	8.0 Gb
Sistema Operativo	Windows 10	macOS 11.2.1 Big Sur

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	1488326.74	24818.868
0.50	1488326.78	24523.953
0.80	1488326.78	24178.195

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	1488336.162	24545.8503
4.00	1488335.578	24111.891
6.00	1488335.578	23836.7897

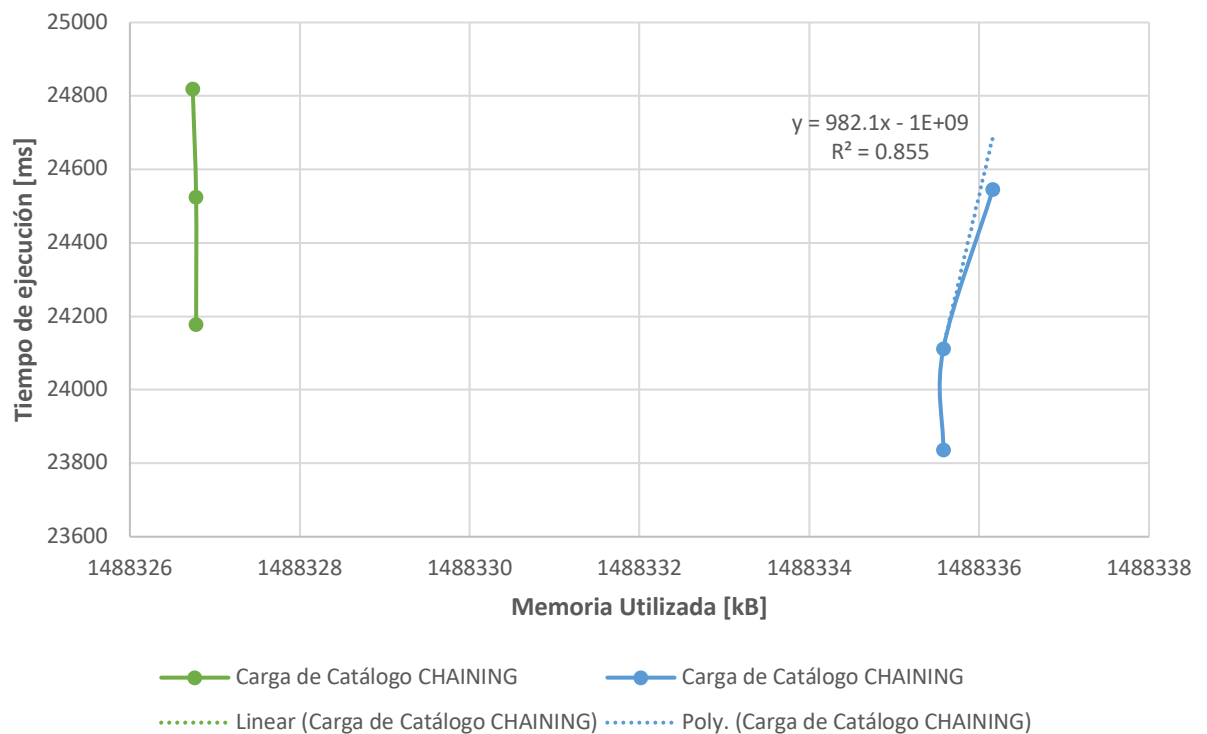
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

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

Comparación de Tiempo y Memoria utilizados en PROBING y CHAINING



Maquina 2

Resultados

Carga de Catálogo PROBING

Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	1488326.07	37631.33
0.50	1488326.07	29803.81
0.80	1488326.07	34235.72

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

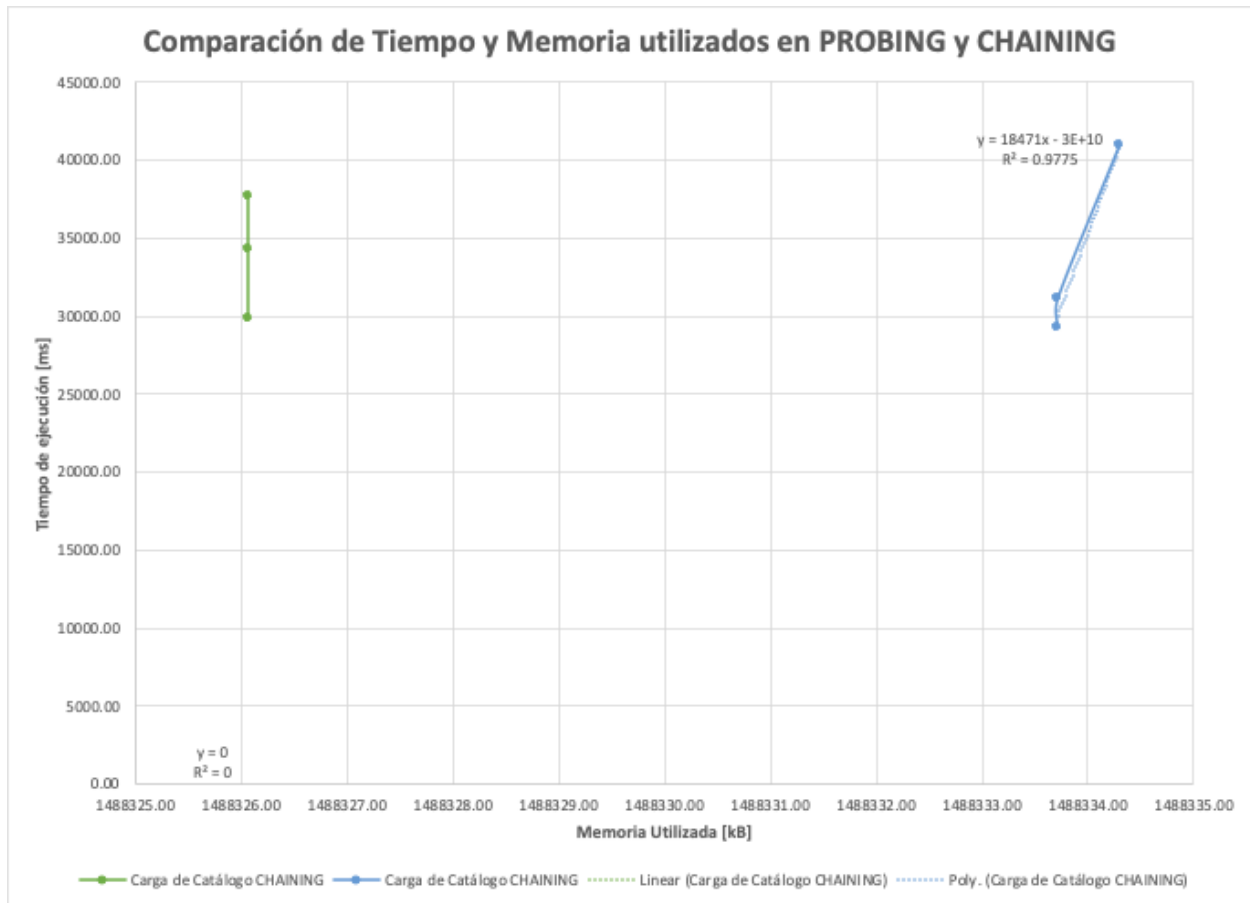
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	1488334.30	40928.03
4.00	1488333.72	31106.44
6.00	148833.72	29265.77

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 la previamente conocida **time.process_time()**?
 - En esencia, se utiliza **time.perf_counter()** en lugar de **time.process_time()** por el tiempo promedio que toman las pruebas en este caso. Para el reto anterior, el cual involucraba únicamente **ARRAY_LIST** y **SINGLE_LINKED**, los tiempos de carga, ordenamiento, ejecución y demás eran mucho mayores. Según la documentación de Python, **time.perf_counter()** es mucho más eficiente para tiempos cortos que **time.process_time()**, pues emplea un performance counter. En otras palabras, cuenta el tiempo con el reloj de mejor definición disponible. Además, incluye en la medición del tiempo los instantes en los que la máquina se encuentra en reposo.
- 2) ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?
 - La función **start()** permite inicializar el proceso para medir la memoria, por otro lado la función **stop()** permite finalizar el proceso de medir la memoria.

- 3) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de videos?
- Cuando se carga con un PROBING a medida que se aumenta el factor de carga máximo se puede ver como el tiempo de ejecución disminuye, por otro lado, cuando se carga con CHAINING, a medida que aumenta el factor de carga máximo también se ve que el tiempo de ejecución disminuye.
- 4) ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar el catálogo de videos?
- En PROBING el factor de carga de 0.5 y 0.8 tienen el mismo que consumo de memoria, pero el factor de carga de 0.3 presenta un consumo menor de memoria, en CHAINING el factor de carga de 4 y 6 tienen el mismo consumo de memoria, pero el factor de carga de 2 presenta un consumo menor de memoria. Esto podría indicar que a medida que se aumenta el factor de carga el consumo de memoria tiende a ser más estable.
- 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.
- Al cambiar el esquema de colisiones se puede ver como la variación en el tiempo no es tan significativa, además cuando se cambia entre ambos esquemas de colisión, se puede apreciar que el factor de conversión juega un papel crucial ya que este influye en el tiempo de ejecución de ambos esquemas.
- 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.
- Al cambiar el esquema de colisiones se puede apreciar que PROBING consume menos memoria que el CHAINING, pero algo que ambos tienen en común es que cuando el factor de conversión se aumenta, el consumo de memoria se vuelve estable.