

OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Andrés Felipe Gómez García Cod 202021189
Estudiante 2 Tomás Bedoya Calixto Cod 202020689

	Máquina 1	Máquina 2
Procesadores	Intel® Core™ i7-5820k CPU @ 3.30GHz 3.30GHz	Intel® Core™ i7-4710HQ CPU @ 2.50GHz 2.50 GHz
Memoria RAM (GB)	16.0 GB	12.0 GB
Sistema Operativo	Windows 10 Pro	Windows 10 Home Single

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	1965558.746	279696.248
0.50	2229675.457	304517.272
0.80	2185181.402	335464.447

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

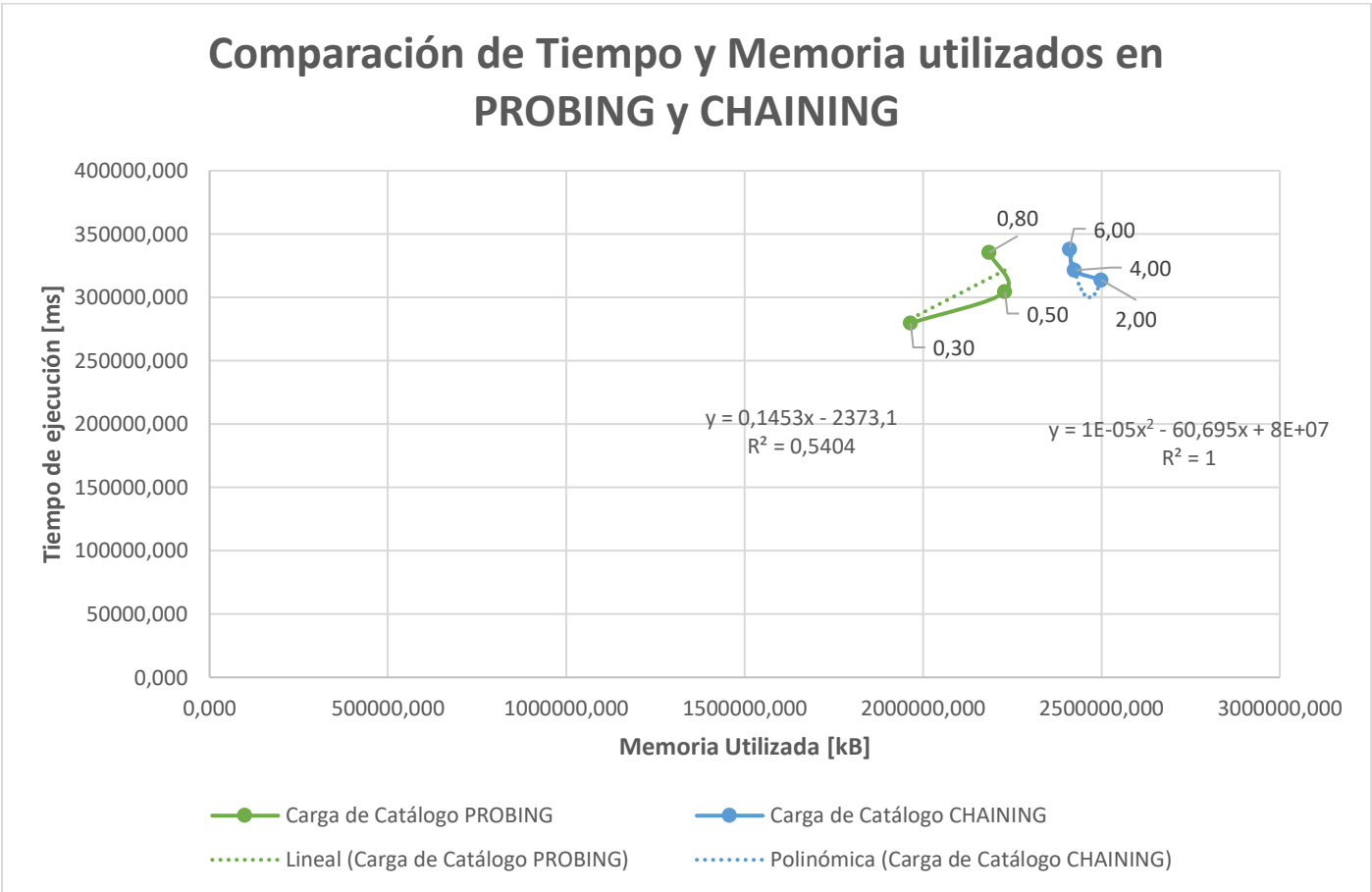
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	2499459.670	313853.392
4.00	2424273.896	321398.762
6.00	2410373.928	338032.991

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]
0.30	2163680.416	286156.669
0.50	2436845.424	303855.106
0.80	2390829.580	345910.998

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

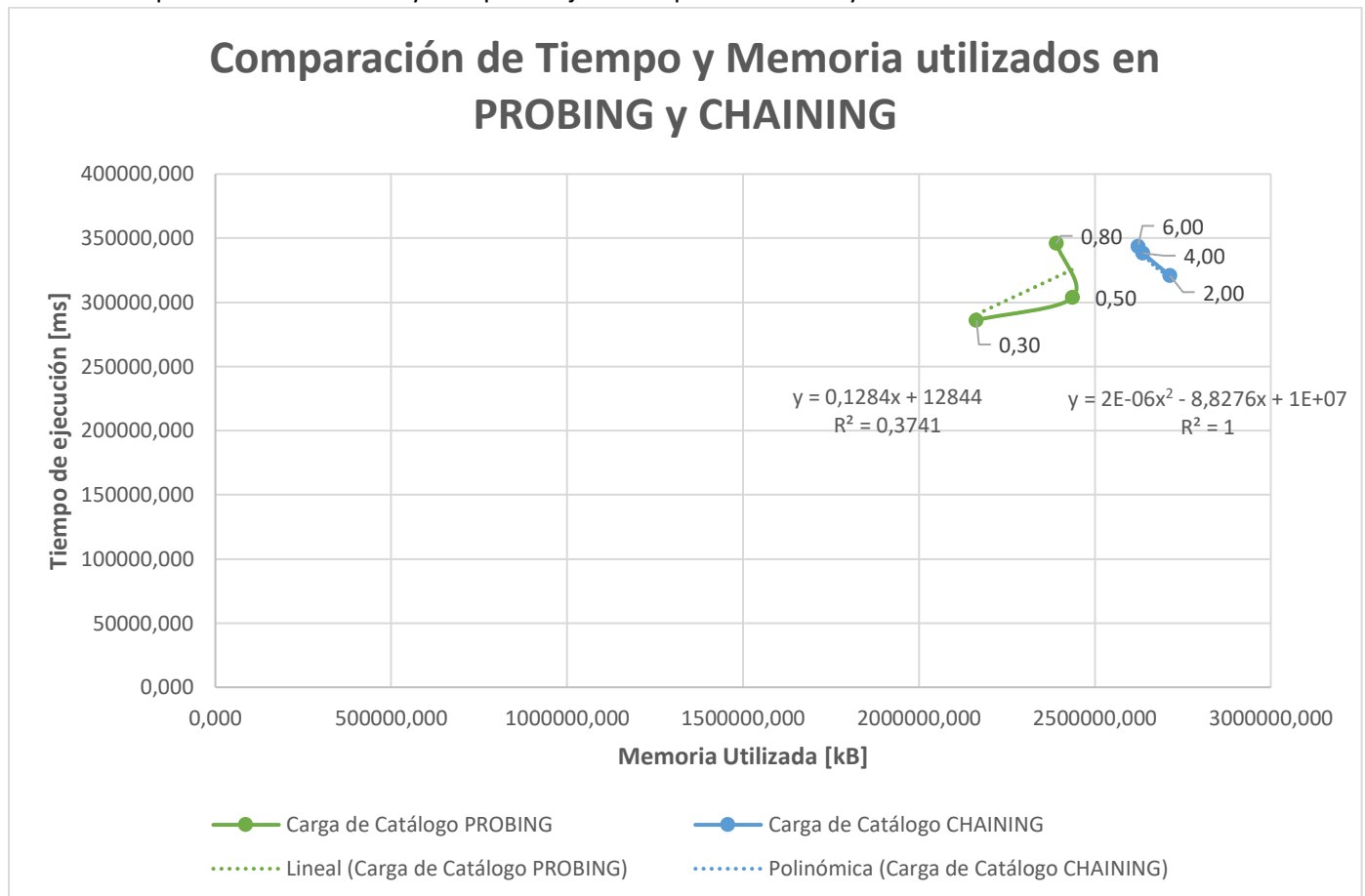
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	2713601.299	320847.158
4.00	2637097.197	338239.582
6.00	2622888.525	343541.468

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 ves de la previamente conocida **time.process_time()**?

Además de que **time.perf_counter()** utiliza un reloj de mayor precisión que **time.process_time()**, **time.perf_counter()** contabiliza el tiempo real en el que un programa transcurre, es decir, toma en cuenta los momentos en los que el programa está durmiendo o no ejecutando una línea de código. Consecuentemente, **time.perf_counter()** nos proporciona una medida temporal más próxima a aquella percibida por el usuario que **time.process_time()**, el cual únicamente contabiliza el tiempo en el que el programa se encuentra corriendo líneas de código.

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

La función **tracemalloc**, a diferencia de las funciones de registro temporal, no monitorea el uso de memoria permanentemente, sino que debe inicializarse para obtener datos de la asignación de memoria de los procesos subsecuentes; para esto se necesita **start()**. Luego de que se utiliza, la

función deja de registrar por medio de stop()), pues no es necesario que recopile información mientras no se necesita.

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

Al analizar los datos registrados, tanto en la maquina 1 como en la maquina 2, es evidente que, conforme aumenta el factor de carga, el tiempo de ejecución del código también muestra un incremento.

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

La variabilidad del consumo de memoria de ninguno de los dispositivos presenta un patrón de cambio aparente al modificarse los factores de carga. De hecho, podría decirse que las variaciones son insignificantes o no relacionadas.

- 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.

Es manifiesto que el tiempo de ejecución resulta de mayor brevedad al utilizarse el sistema de Linear Probing, pues este resulta ser más rápido que Separate Chaining para factores de carga proporcionales.

- 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.

Se aprecia como, al emplearse Separate Chaining, la memoria utilizada es considerablemente mayor que al utilizar Linear Probing para todos los factores de carga experimentados. Además, se reitera que en ninguno de los 2 esquemas de colisiones se observa un patrón de cambio con respecto al uso de memoria cuando se cambia el factor de carga.