

OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod 202013371

Estudiante 2 Cod 202022217

	Máquina 1	Máquina 2
Procesadores	I5-9300H CPU 2.40GHZ	AMD Ryzen 5 3500U
Memoria RAM (GB)	16.0 GB	12.0 GB
Sistema Operativo	Windows 10	Windows 10, 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]
0.30	1488314.086	28732.028
0.50	1488317.582	28510.490
0.80	1488301.354	29334.666

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

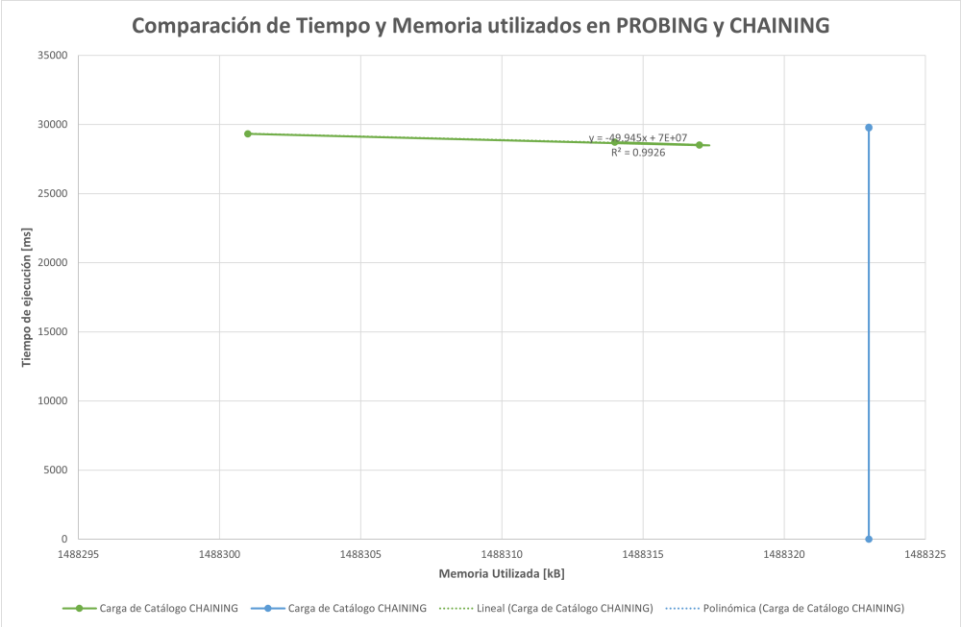
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	1488323.860	29752.515
4.00	1488323.860	29803.548
6.00	1488323.860	30721.251

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	1488359.764	43228.611
0.50	1488344.046	43889.892
0.80	1488326.0	43607.681

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.

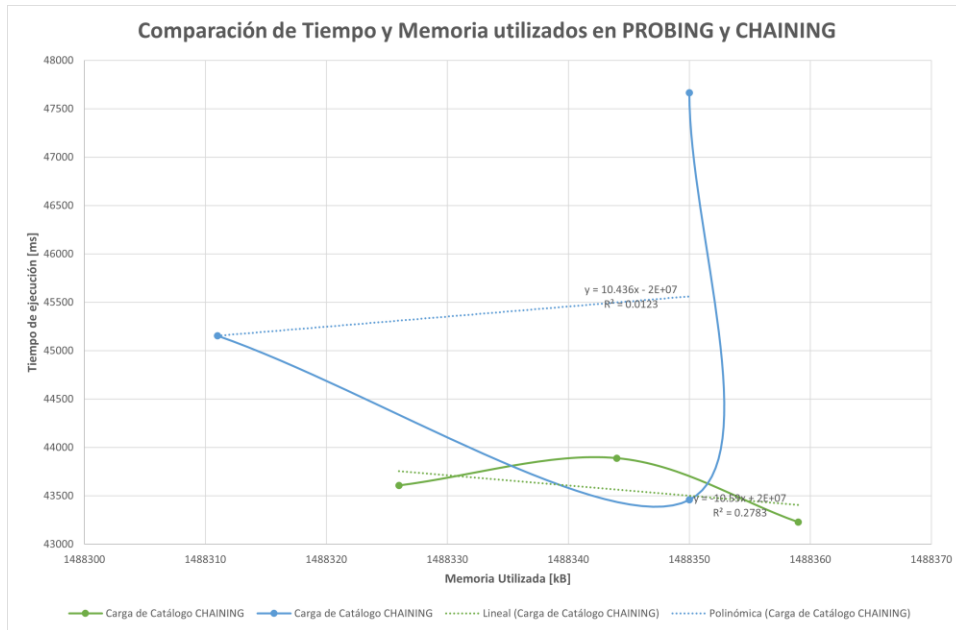
Carga de Catálogo CHAINING		
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	1488350.029	47664.553
4.00	1488350.029	43458.452
6.00	1488311.262	45154.465

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()**?

Aunque `process_time` es mucho mas rápido(4.67 veces mas rapido que `perf_counter`), `time.process_time` solo mide la suma del tiempo de CPU del sistema y del usuario del proceso actual, mientras que `perf_counter` funciona como un cronómetro normal.

Ya que además de la eficiencia del código se evalúa los componentes de la maquina en la cual se corre, `perf counter` da tiempos mas precisos pues no se limita al código si no que toma en cuenta todos los aspectos. Esta precisión se evidencia al comparar las diferencias de tiempos medidos en un `for in range(1000000)`:

Clock	Num Changes	Time to Complete
<code>perf_counter</code>	500,982	0.24273775100482453
<code>process_time</code>	22	0.328125

[Python Clocks Explained - Webucator Blog](#)

`Process_time` solo evidenció 22 cambios de tiempo del millón de ciclos completados y cada ciclo le tomó 0.32 segundos, mientras que `perf_counter` percibió 500,982 cambios de tiempo del millón de ciclos completados y le tomó 0.24 segundos por ciclo.

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

Por que la función `start()` es la que inicia el rastreo de la asignación de memoria por frame, mientras que la función `stop()` para dicho rastreo y remueve los "ganchos" que hacen el rastreo de la asignación

de memoria en python. Su importancia radica en cuando y donde se ejecutan start() y stop(). pues, por ejemplo, si se inicia start() en el controller.py y se ejecuta stop() después de un print es probable que se incluya el uso de memoria para el repr del print, lo cual dependiendo de la cantidad de datos, puede ser un número considerable y altera la precisión de la cantidad medida.

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

A medida que se aumentaba el factor de carga, los tiempos de ejecución aumentaban de manera ligera, siendo la mayor brecha entre tiempos de dos segundos. Su valor general se mantuvo constante (no hubo saltos grandes entre las mediciones de tiempo).

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

El consumo de memoria en general se mantuvo constante. La mayor diferencia entre tiempos no pasaba de un megabyte y la menor diferencia era inferior a 300 bytes.

- 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 de probing a chaining, los tiempos de ejecución aumentaron en general, pero se mantuvieron cercanos a los valores correspondientes en probing (la mayor diferencia entre tiempos en probing-chaining fue de 10 segundos.). Los tiempos en chaining siguieron el comportamiento de probing.

- 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 de probing a chaining, el uso de memoria promedio fue mayor en cantidades despreciables (menor a 50kb) por lo que es válido afirmar que no se percibió un cambio de memoria considerable (se mantuvo constante).