

# OBSERVACIONES DEL LA PRACTICA 7

Estudiante 1: Ehimar Andres Vargas Malaver. 202014902. e.vargasm()uniandes.edu.co

Máquina 1	
Procesadores	AMD Ryzen 7 4800H 4.2GHz
Memoria RAM (GB)	16,0 GB
Sistema Operativo	Windows 10 Pro

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	2273014,855	52959,524
0.50	2273115,124	52959,175
0.80	2273189,60	54227,855

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	2273040,630	54729,29
4.00	2273114,855	55160,00
6.00	2273174,811	55659,37

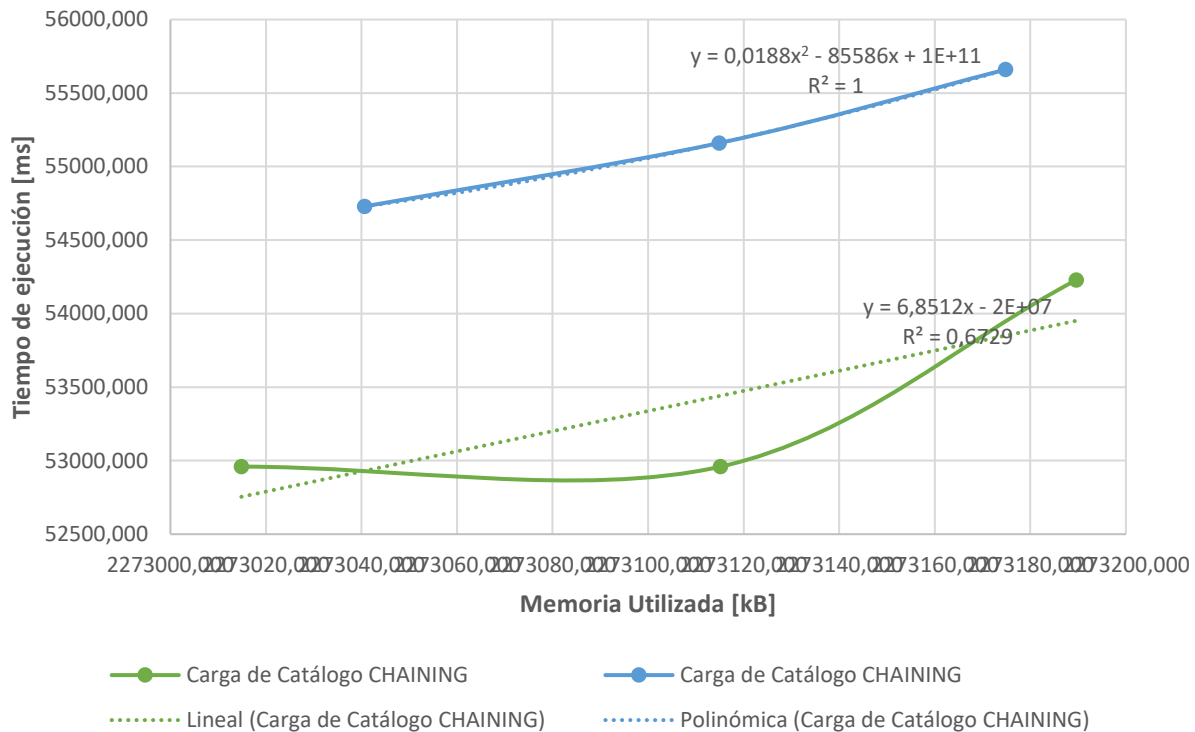
Tabla 3. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 1.

### 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

### Preguntas de análisis

- 1) ¿Por qué en la función **getTime ()** se utiliza **time? perf\_counter ()** en ves de la previamente conocida **time. process\_time ()**?

Se utiliza **time. perf\_counter ()** ya que este mide la cantidad de tiempo real que tarda un proceso en realizarse usando dos intervalos de tiempo, logrando así una alta precisión en los tiempos medidos, contrario a **time. process\_time ()** que mide el tiempo de manera continua.

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

son importantes ya que “**start()**” indica que desde ese momento se debe analizar cómo se están organizando los datos entrantes a la memoria dentro de esta, para así, poder medir la cantidad de memoria usada y hasta el momento en que se use la función “**stop()**” la cual borra todos lo registros de memoria asignados para que no afecte las mediciones en otros procesos.

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

Los cambios en los tiempos de carga son mínimos por no decir nulos, no hay mucha diferencia entre usar un factor de carga pequeño o alto, tal y como se puede observar en la gráfica, esto se podría deber a que es muy poca la cantidad de datos que se están guardando.

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

El programa retorna valores muy similares de memoria ocupada al variar el factor de carga, este cambio tan pequeño podría deberse a la manera en cómo se pensó guardar los datos, debido a que estos son muy pocos, los factores de carga dejan de ser relevantes.

- 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.

Debido a que la cantidad de datos es muy poca y las casillas disponibles son muchas, las colisiones son muy pocas, no importa si se usa "PROBING" o "CHAINING" los tiempos de carga son muy similares.

- 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.

Debido a que la cantidad de datos es muy poca y las casillas disponibles son muchas, las colisiones son muy pocas, no importa si se usa "PROBING" o "CHAINING" la cantidad de memoria ocupada es muy similar.