**OBSERVACIONES DEL LA PRACTICA**

Sergio Arango Cod 201921814

Helena Vegalara Cod 201823328

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel Core i7 | AMD Ryzen 5 |
| Memoria RAM (GB) | 12 | 8 |
| Sistema Operativo | Windows 10 | Windows 10 |

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 | 296210.054 | 19716.255 |
| 0.50 | 296210.024 | 20971.662 |
| 0.80 | 296210.002 | 22287.743 |

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.

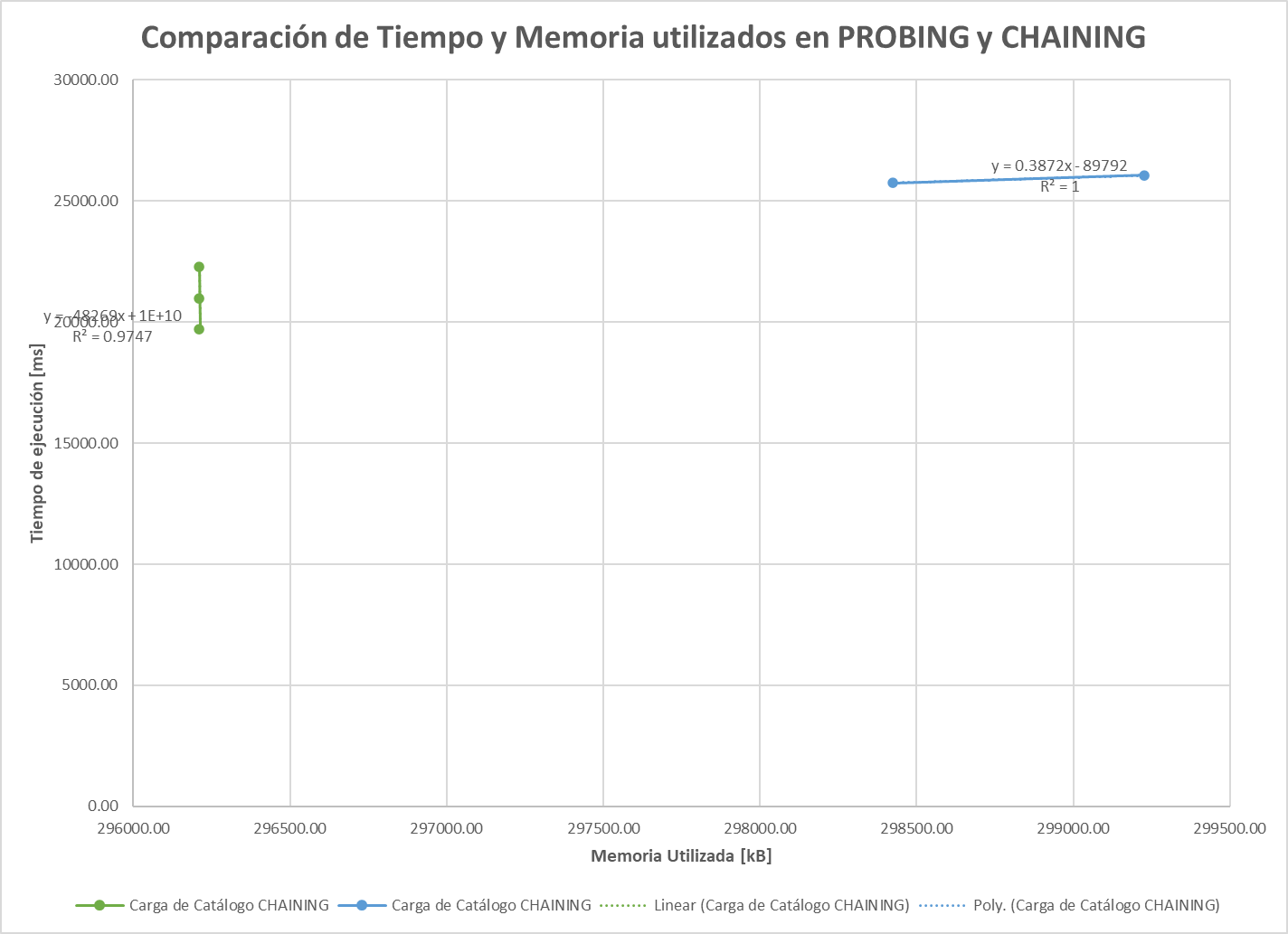
|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 298424.123 | 25345.106 |
| 4.00 | 299226.439 | 25749.111 |
| 6.00 | 30061.998 | 26059.745 |

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 | 297826.804 | 29268.899 |
| 0.50 | 297826.742 | 32252.491 |
| 0.80 | 297826.684 | 33803.605 |

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.

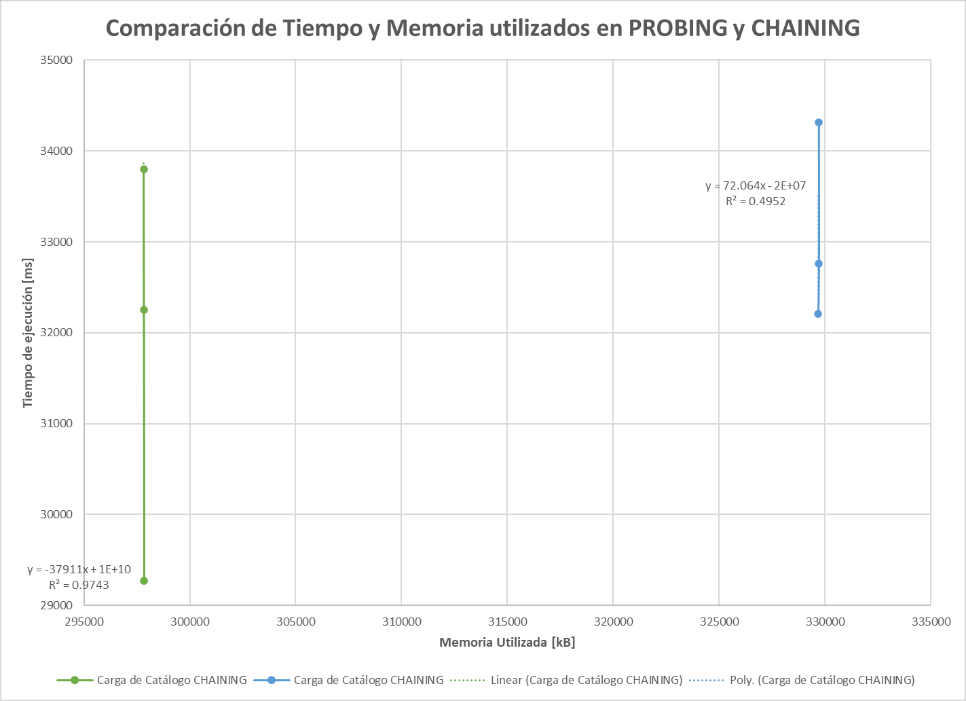
|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución [ms]** |
| 2.00 | 329702.293 | 34319.298 |
| 4.00 | 329702.293 | 32762.421 |
| 6.00 | 329683.761 | 32205.361 |

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()**?

Es más precisa porque mide el tiempo más veces por segundo.

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

Permiten fijar los puntos con los cuáles se define el intervalo de tiempo que se quiere medir.

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

Se puede ver que con el aumento del valor para el factor de carga, el tiempo de ejecución tiende a aumentar.

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

En este caso, el consumo de memoria se mantuvo aproximadamente constante, sin embargo, por lo general, entre mayor sea el factor de carga, más consumo de memoria va a haber con el programa.

1. ¿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 general, los tiempos de ejecución al utilizar Chaining fueron mayores que al usar Probing. Esto se debe a que Probing “no sabe” dónde se encuentra una llave que no está en el primer lugar que debería estar según su valor de hash entonces debe hacer un recorrido lineal sobre todos los lugares para encontrar la llave que busca. Esto no suena tan mal hasta que la estructura intenta encontrar una llave la cuál no está, deberá hacer recorridos poco eficientes.

1. ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

En cuanto al consumo de memoria, se pudo ver que el esquema de colisiones Chaining tuvo un mayor consumo de memoria que el esquema de colisiones Probing. Esto se debe a que en cada valor correspondiente a un valor de hash se tiene una lista de llaves correspondientes. Estas listas son las que hace que este método consuma más memoria.