**OBSERVACIONES DEL LA PRACTICA**

Santiago Martínez Delgadillo Cod 202012611

Estudiante Kevin David Alvarez Cod 202022834

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel Core i5-9300H |  |
| Memoria RAM (GB) | 8 GB |  |
| Sistema Operativo | Windows 10 Home 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 | 1729157.476 | 43643.18 |
| 0.50 | 1729157.524 | 43395.05 |
| 0.80 | 1729157.582 | 45815.64 |

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 | 1729179.818 | 47989.19 |
| 4.00 | 1729179.834 | 45605.87 |
| 6.00 | 1729179.856 | 44752.82 |

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 | 1406082.896 | 27223.609 |
| 0.50 | 1406082.896 | 27059.735 |
| 0.80 | 1406082.896 | 27221.51 |

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 | 1406082.896 | 28745.101 |
| 4.00 | 1406082.896 | 28306.245 |
| 6.00 | 1406082.896 | 28456.738 |

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()**?

La función time.process\_time() es la encargada de calcular la suma del tiempo del cpu del sistema y del usuario del proceso actual. Mientras que time.perf\_counter() mide la cantidad real de tiempo que tarda un proceso como si se utilizara un cronometro, sin tomar en cuenta el cpu del sistema.

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

Porque indican los intervalos desde donde se va a realizar el calculo de la memoria utilizada por las funciones, en este caso, mas pesadas del programa, ya que la unción start() empieza a rastrear asignaciones de memoria.

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

El tiempo de ejecución a medida que aumenta el factor de carga tiene ligeras variaciones, pero siempre se mantiene dentro de un mismo rango. El tiempo de ejecución tiene cambios más significativos al momento de hacer búsquedas de datos en vez de cargar los mismos.

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

No se percibieron cambios en memoria en los dos dispositivos, esto esta relacionado con la carga de los archivos ya que al realizar las pruebas se uso siempre el mismo archivo sin un dato más ni uno menos.

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.

Al cambiar el esquema de colisiones los tiempos de ejecución tuvieron un aumento ligero sin que influya mucho el factor de carga. En ambas maquinas el esquema ‘PROBING’ tuvo el menor tiempo de ejecucion en general. Esto se puede explicar con la definicón teorica del mismo esquema, ya que este le da una unica posición a un unico elemento, y comparandolo con el esquema ‘CHAINING’ el cual al poseer listas enlazadas tiene que sumar el tiempo de agregar un elemento al final de las mismas.

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.

No se detectaron cambios por la razón explicada en la respuesta 4.