**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod 202013610

Estudiante 2 Cod 202020706

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | Intel(R)Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz | Intel(R)Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz |
| Memoria RAM (GB) | 16.0 GB 2666 Mhz | 16.0 GB 2666 Mhz |
| Sistema Operativo | Windows 10 Home - 64bits | Windows 10 Home - 64bits |

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 | 1623637.474 | 27223.032 |
| 0.50 | 1623637.532 | 27516.480 |
| 0.80 | 1623637.478 | 28231.180 |

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 | 1623662.165 | 28631.702 |
| 4.00 | 1623662.165 | 28796.579 |
| 6.00 | 1623662.165 | 28930.514 |

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 | 1623637.475 | 23189.949 |
| 0.50 | 1623637.532 | 23508.850 |
| 0.80 | 1623637.479 | 24039.130 |

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 | 1623662.104 | 24528.151 |
| 4.00 | 1623662.102 | 24563.897 |
| 6.00 | 1623662.165 | 24696.098 |

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.perf\_counter()** nos permite tener una visión más exacta de cuanto duro el proceso. Esto debido a que la función **time.process\_time()** evalúa el tiempo que tomó el proceso en función del computador, cosa que no sucede con **time.perf\_counter()** ya que este funciona como un cronómetro y mide cuanto tardó el proceso exactamente. Por ejemplo, el **time.process\_time()**  cuenta el tiempo que tardo el computador en hacer los prints de una función, el **time.perf\_counter()** no lo hace sino que sólo tomá el tiempo que tardo en realizar los procesos.

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

Estas dos funciones son importantes para que **time.perf\_counter()**  funcione correctamente. Estos funcionan, como su nombre lo indica, como el inicio y el final del cronometro.

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 no se ve casi afectado, sólo hay unas ligeras variaciones, pero en desde una vista muy general son bastante parecidos

1. ¿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 presenta un comportamiento interesante ya que las variaciones son mínimas, cosa que no vimos en las pruebas en el código de books que presentaba una clara tendencia a usar menos memoria con un mayor factor de carga.

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 modificar el esquema de colisiones observamos que en probing tenemos un menor tiempo de respuesta. Sin embargo, la diferencia entre los dos esquemas es muy poca y encontramos que son bastante 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.

Sorprendentemente la memoria que utilizó para que se pudiera realizar el proceso en los dos esquemas de colisiones es prácticamente idéntica, cosa que no vimos en las pruebas en el código de books que tendía a consumir menos datos a mayor tiempo de carga y en el esquema probing.