**OBSERVACIONES DEL LA PRACTICA**

Juan Lago

Juan David Ortiz

Mauricio Martinez

|  |  |  |  |
| --- | --- | --- | --- |
|  | Máquina 1 | Máquina 2 | Máquina 3 |
| Procesadores | Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz | AMD Ryzen 3 4300U with Radeon Graphics 2.70 GHz | Intel Core i7-11370H CPU 4.80 GHz |
| Memoria RAM (GB) | 32.0 GB | 8.0 GB | 16GB |
| Sistema Operativo | Windows 11 - 64 bits | Windows 11-64 bits. | Windows 11 - 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 Real @LP [ms]** |
| 0.1 | **84674.29** | **3240.07** |
| 0.5 | **46228.76** | **1944.21** |
| 0.7 | **43511.36** | **2051.18** |
| 0.9 | **42056.76** | **2352.86** |

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 Real @SC [ms]** |
| 2.00 | 45088.69 | 1809.46 |
| 4.00 | 42468.70 | 1472.59 |
| 6.00 | 42759.87 | 1679.59 |
| 8.00 | 41993.31 | 1563.71 |

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 Real @LP [ms]** |
| 0.1 | 122677.75 | 6277.83 |
| 0.5 | 72963.78 | 4149.90 |
| 0.7 | 69454.85 | 4171.46 |
| 0.9 | 67575.13 | 5233.06 |

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 Real @SC [ms]** |
| 2.00 | 71738.21 | 5156.28 |
| 4.00 | 68217.76 | 4442.18 |
| 6.00 | 68608.48 | 4650.00 |
| 8.00 | 67579.10 | 3974.41 |

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

# **Maquina 3**

## **Resultados**

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo PROBING*** | | |
| **Factor de Carga (PROBING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @LP [ms]** |
| 0.1 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

Tabla 6. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 3.

|  |  |  |
| --- | --- | --- |
| ***Carga de Catálogo CHAINING*** | | |
| **Factor de Carga (CHAINING)** | **Consumo de Datos [kB]** | **Tiempo de Ejecución Real @SC [ms]** |
| 2.00 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

Tabla 7. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 3.

## **Graficas**

La gráfica generada por los resultados de las pruebas de rendimiento en la **Maquina 3.**

* 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 vez de otras funciones como **time.process\_time()**?

Porque proporciona acceso al reloj con la mayor resolución disponible para hacer mediciones a corto plazo más precisas.

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

Porque debe iniciar el proceso de toma de datos antes de ejecutar la función del model.py

1. ¿Por qué no se puede medir paralelamente el **uso de memoria** y el **tiempo de ejecución** de las operaciones?

Porque al medir el uso de memoria tambien se utilizan recursos y tarda mas tiempo y cuando no se hace se puede medir el tiempo solo de las operaciones sin la de medir el uso de memoria

1. Teniendo en cuenta cada uno de los requerimientos del reto ¿Cuántos índices implementaría en el Reto? y ¿Por qué?

Crearia 3, uno para cada grupo de datos sobre resultados, scorers, penalties

1. Según los índices propuestos ¿en qué caso usaría **Linear Probing** o **Separate Chaining** en estos índices? y ¿Por qué?

Usaria Chaining en scorers y panlties porque

1. Dado el número de elementos de los archivos del reto (large), ¿Cuál sería el factor de carga para estos índices según su mecanismo de colisión?
2. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de contenido FIFA?

Entre mayor el factor de carga menor el tiempo en probing y en chaining tambien pero no tanto

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

Igual, entre mayor factor de carga menos memoria necesita.

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.

El tiempo en chaining es normalmente menor al de probing

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.

Igualmente, en el consumo de memoria el de chaining es constantemente menor que el de probing a lo largo del cambio en sus factores de carga

1. ¿Qué configuración de ideal ADT Map escogería para el **índice jugadores FIFA (“scorer”)**?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.

Usaria chaining con factor de carga 8 e iniciando con 800 elementos ya que da los mejores resultados en memoria y tiempo