**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

Estudiante 3 Cod XXXX

|  |  |  |  |
| --- | --- | --- | --- |
|  | Máquina 1 | Máquina 2 | Máquina 3 |
| Procesadores | AMD Ryzen 5 5500U with Radeon Graphics, 2100 Mhz, 6 procesadores principales, 12 procesadores lógicos |  |  |
| Memoria RAM (GB) | 8.00 GB |  |  |
| Sistema Operativo | Microsoft Windows 11 Home Single Language |  |  |

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 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

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 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

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 |  |  |
| 0.5 |  |  |
| 0.7 |  |  |
| 0.9 |  |  |

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 |  |  |
| 4.00 |  |  |
| 6.00 |  |  |
| 8.00 |  |  |

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()**?

* En la función “getTime()”, se utiliza porque “time.perf\_counter()” proporciona una medida más precisa del tiempo de procesamiento en milisegundos, especialmente en entornos donde la precisión del tiempo es crítica, como en mediciones de rendimiento. “time.perf\_counter()” devuelve una medida de tiempo que es independiente del tiempo del sistema y que tiene una mayor resolución que “time.process\_time()”.

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

* Las funciones “start()” y “stop()” son importantes porque permiten realizar un seguimiento detallado del uso de la memoria. La función start(), inicia el rastreo de asignaciones de memoria, lo que significa que a partir de este punto, “tracemalloc” comenzará a registrar información sobre las asignaciones de memoria realizadas por el programa. La función stop(), detiene el rastreo de asignaciones de memoria, después de llamar a stop(), “tracemalloc” deja de registrar la información sobre las asignaciones de memoria.

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

* No es posible medir de manera simultánea y directa el uso de memoria y el tiempo de ejecución de las operaciones debido a sus diferentes métricas y unidades de medida, así como el impacto en el rendimiento y la posible interferencia entre las mediciones.

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

* Para el reto sería apropiado implementar varios índices para optimizar las consultas como los siguientes:

1. Índice por país: Un índice que permita buscar ofertas de trabajo por país. Esto facilitará las consultas relacionadas con países específicos, como en los requerimientos 1,4,5 y 8.
2. Índice por nivel de experticia: Otro índice para buscar ofertas de tabajo según el nivel de experticia. Esto será útil para el requerimiento 1 y posiblemente para otros que requieran filtrar por nivel de experiencia.
3. Índice por empresa: Un índice que vincule las ofertas de trabajo con la empresa que las publicó, esto serviría para el requerimiento 3, donde se busca consultar ofertas por empresa durante un periodo de tiempo específico.
4. Índice por ciudad: Un índice por ciudad sería útil para para consultas que involucren ciudades, como el requerimiento 5.
5. Índice por fecha de publicación: Un índice que ordene las ofertas por fecha de publicación será útil para el requerimiento 3 y cualquier otra consulta que necesite filtrar por periodos de tiempo.
6. Índice para clasificación: si se requiere clasificar ciudades o países según el número de ofertas de trabajo o experticia, se podría implementar un índice adicional que facilite esta clasificación sin tener que realizar cálculos complejos en tiempo real.
7. Según los índices propuestos ¿en qué caso usaría **Linear Probing** o **Separate Chaining** en estos índices? y ¿Por qué?
8. Índice por país y ciudad: Linear Probing, dado que se espera un número limitado de colisiones y un tamaño de índice relativamente pequeño para los países y ciudades, Linear Probing podría ser el mas adecuado. Proporciona una implementación mas simple y eficiente en este contexto.
9. Índice por nivel de experticia y fecha de publicación: Separate Chaining, considerando la diversidad de niveles de experticia y la posibilidad de muchas ofertas de trabajo publicadas en diferentes fechas, es probable que haya colisiones frecuentes. Por lo tanto, Separate Chaining es el más apropiado para este índice.
10. Índice por empresa y clasificación: Linear Probing, dado que el número de empresas y clasificaciones puede no ser tan grande como otros índices, y las colisiones podrían ser menos frecuentes, Linear Probing sería una elección adecuada aquí debido a su simplicidad y menor uso de memoria.
11. 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?
12. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar las ofertas de trabajo?
13. ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar las ofertas de trabajo?
14. ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
15. ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
16. ¿Qué configuración de ideal ADT Map escogería para el **índice de ofertas de trabajo por “id”**?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.