**OBSERVACIONES DEL LA PRACTICA**



Adriana Sofía Rozo Cepeda 1 Cod 202211498

Diego Fernando Galván Cruz 2 Cod 202213709

|  |  |  |  |
| --- | --- | --- | --- |
|  | Máquina 1 | Máquina 2 | Máquina 3 |
| Procesadores | 11th Gen Intel(R) Core(TM) i7-1195G7 @ 2.90GHz 2.92 GHz |  |  |
| Memoria RAM (GB) | 16,0 GB (15,7 GB usable) |  |  |
| Sistema Operativo | Sistema operativo de 64 bits, procesador basado en x64 |  |  |

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 | 34929.78 | 251.05 |
| 0.5 | 34929.78 | 225.16 |
| 0.7 | 34929.78 | 194.11 |
| 0.9 | 35114.09 | 189.27 |

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 | 35114.82 | 214.91 |
| 4.00 | 35114.82 | 203.05 |
| 6.00 | 35282.56 | 203.16 |
| 8.00 | 45348.3 | 206.98 |

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()**?

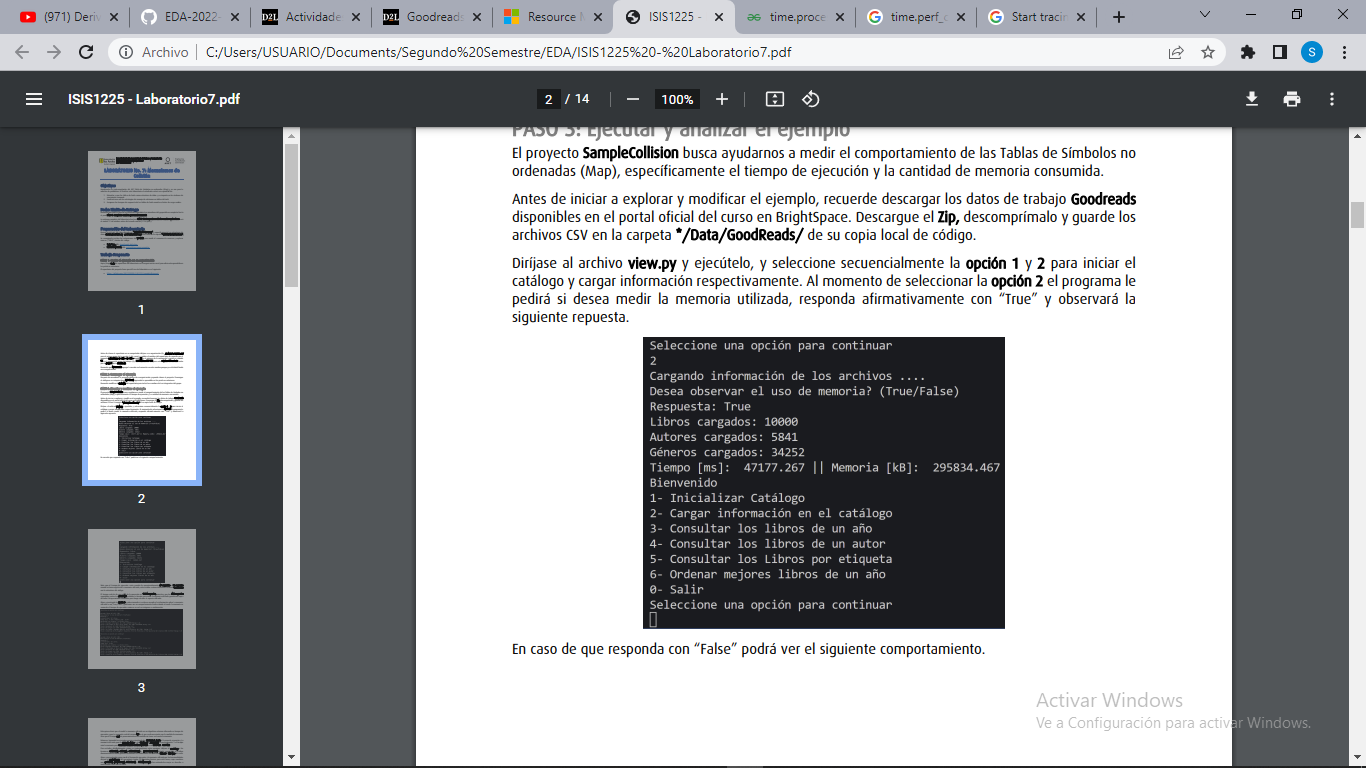
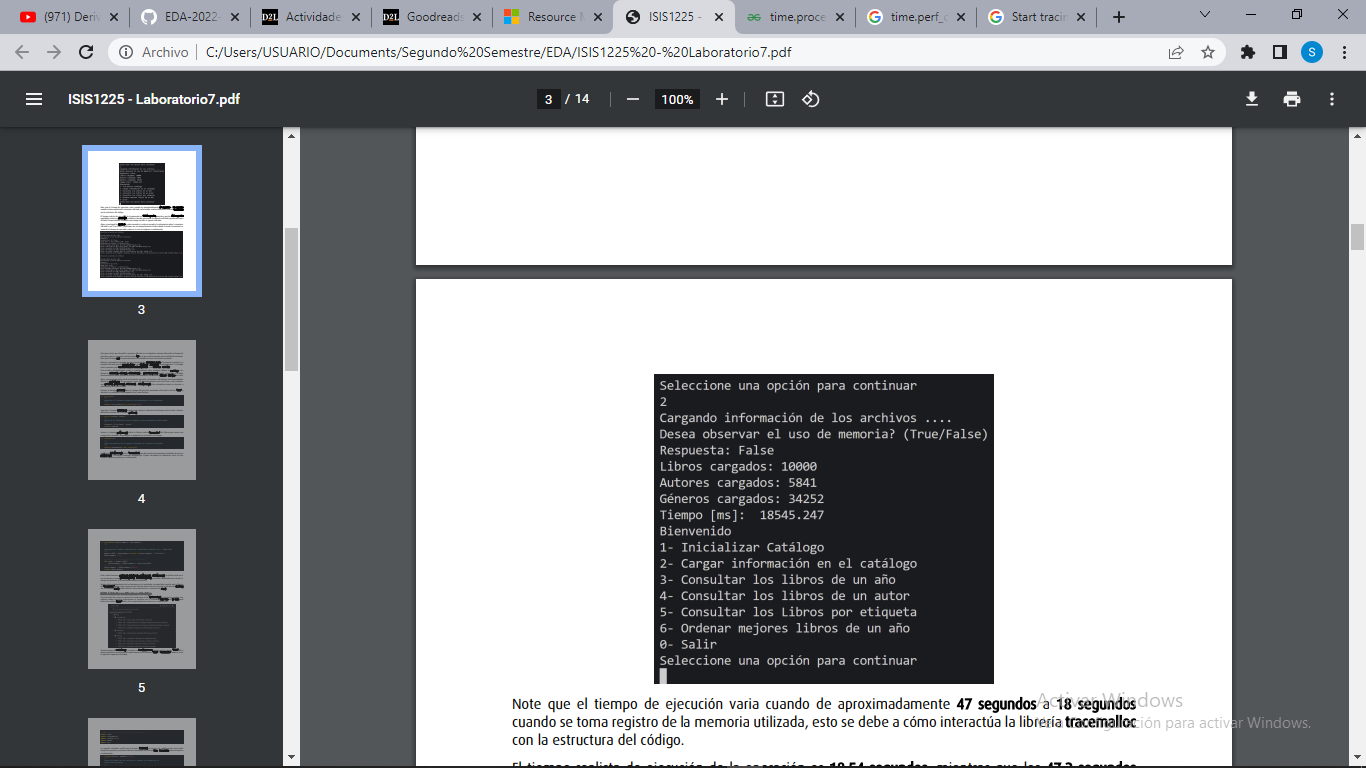
Porque time.perf\_counter() mide el tiempo real que se demora en hacer el proceso, (como si se midiera con un cronometro),aparte muestar una mayor cantidad de veces en comparación con otras funciones, lo que le da una mayor presición, mientras que otras funciones como time.process\_time() son más imprecisas.

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

Porque dichas funciones son las que comienzan o finalizan el trazo o rastreo de las asignaciones de memoria de Phyton. Son necesarias para permitir la toma de estos datos.

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

El tiempo de ejecución aumenta cuando se mide el uso de memoria, por ende, no se estaría midiendo correctamente el tiempo de ejecución. El tiempo real de ejecución se ve distorsionado al pedir la memoria, esto se debe a cómo interactúa la librería tracemalloc con la estructura del código.





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

Se implementarían aproximadamente 10 índices (pueden ser menos) debido a que a cada requerimiento se le asignaría unos (uno por cada requerimiento y adicionales para búsqueda).



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

Elegimos usar Linear Probing en todos los casos con loadfactor=0.4 porque tiene un tiempo considerablemente estable considerando que la memoria tiene menos consumo que con separate chaining.



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?



0.4 con Linear Probing.

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

Entre más pequeño el factor de carga el tiempo aumenta. Con factores de carga más grandes el tiempo disminuye (comportamiento Linear Probing y Chaining).

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

No varía mucho el consumo de memoria, se mantiene estable exceptuando en los últimos factores de carga. Por ejemplo, linear probing y chaining.

Ultimo factor carga probing:

|  |  |  |
| --- | --- | --- |
| 0.9 | 35114.09 | 189.27 |

Ultimo factor carga chaining:

|  |  |  |
| --- | --- | --- |
| 8.00 | 45348.3 | 206.98 |

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.

Para chaining el tiempo de ejecución era más rápido que el linear probing aunque no variaba mucho. Sin embargo, probing (comparando completamente todos los factores de carga) tenía el mejor tiempo de carga.

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.

Chaining consume más memoria que linear probing (podemos evienciar esto en las tablas de las pruebas).

1. ¿Qué configuración de ideal ADT Map escogería para el **índice géneros de contenido (“listed\_in”)**?, especifique el mecanismo de colisión, el factor de carga y el numero inicial de elementos.

