

# OBSERVACIONES DEL LA PRACTICA

JOSE VICENTE RINCON CELIS Cod 202013321

JUAN ESTEBAN ARBOLEDA RESTREPO Cod 201921578

	Máquina 1	Máquina 2
Procesadores	Intel Core I5 – 8300H CPU @ 2.30 GHz	1,4 GHz Intel Core i5 de cuatro núcleos
Memoria RAM (GB)	8 GB	8 GB 2133 MHz
Sistema Operativo	Windows 10 Home Single Language (64 - bits)	MacOS BigSour

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	1388748,470	24878,704
0.50	1388711,540	22966,672
0.80	1388711,540	23074,679

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

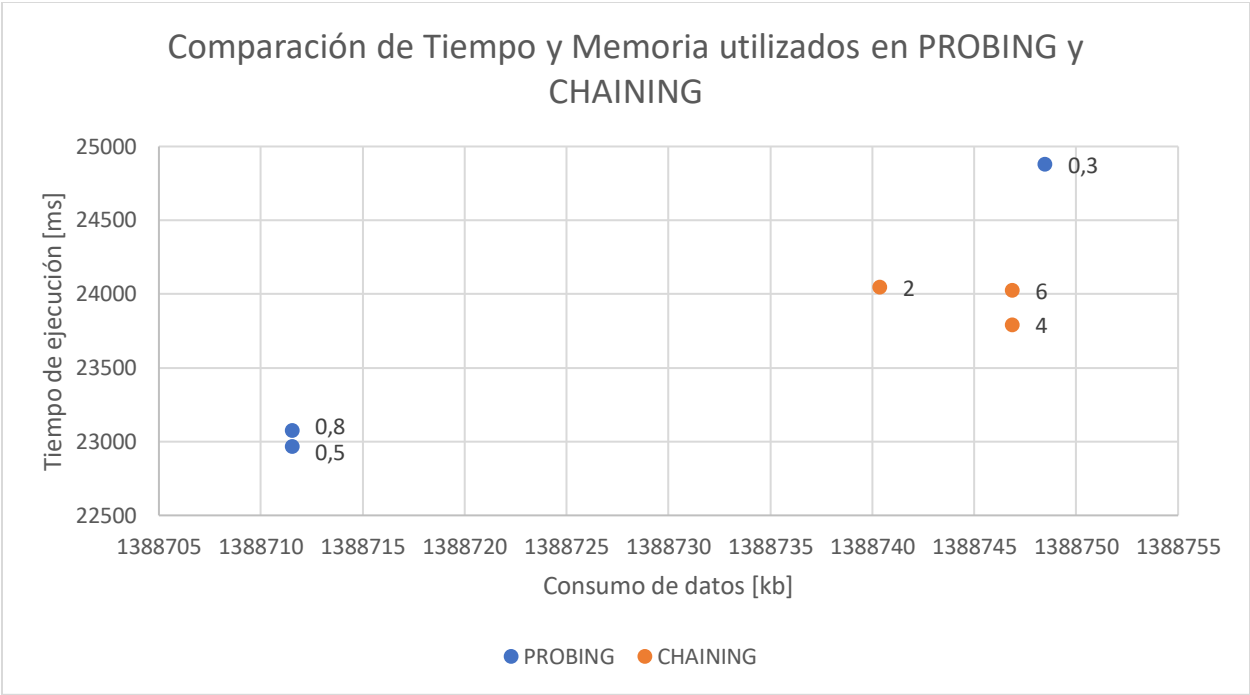
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	1388740,371	24047,596
4.00	1388746,859	23791,848
6.00	1388746,859	24026,378

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



\* El factor de carga de cada dato se presenta como etiqueta de cada punto.

Maquina 2

Resultados

Carga de Catálogo PROBING		
Factor de Carga (PROBING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
0.30	1406112.284	29688.01
0.50	1406048.535	25471.671
0.80	1406048.535	24147.626

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	1406048.535	24261.3
4.00	1406048.535	24333.803
6.00	1406048.535	24203.522

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 vez de la previamente conocida **time.process\_time()**?
  - Se utiliza la función `perf_counter()` ya que esta tiene mayor resolución a la hora de contar el tiempo.
- 2) ¿Por qué son importantes las funciones **start()** y **stop()** de la librería **tracemalloc**?
  - Las funciones `start()` y `stop()` de la librería `tracemalloc` son importantes para que Python sepa cuando empezar y cuando terminar el seguimiento a la memoria. Esto es necesario, ya que por definición, la librería `tracemalloc` “es una herramienta de depuración para rastrear bloques de memoria asignados por Python”<sup>1</sup>. Esto quiere decir, que Python no hace seguimiento a los bloques de memoria por defecto, por lo que es importante explicitar cuando empezar y finalizar el seguimiento a la memoria.
- 3) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el factor de carga máximo para cargar el catálogo de videos?
  - Es difícil encontrar una correlación entre el tiempo de ejecución y el factor de carga de los datos. Esto se debe a dos factores. El primero es que en todos los casos, se están cargando los mismos datos, por lo que los tiempos de carga deben ser similares (la

<sup>1</sup> <https://docs.python.org/3/library/tracemalloc.html>

complejidad temporal de cargar datos no depende del manejo de colisiones ni del factor de carga). En ese orden de ideas, el tiempo debería permanecer constante en todos los escenarios. En todo caso, las otras tareas que corren en el computador causan las variaciones.

- 4) ¿Qué cambios percibe en el **consumo de memoria** al modificar el factor de carga máximo para cargar el catálogo de videos?
  - Aparentemente al aumentar el factor de carga, el consumo de memoria disminuye. Esto tiene sentido ya que muestra que el array que hay detrás de la tabla de HASH es de menor tamaño, y por consiguiente ocupa menos espacio.
- 5) ¿Qué cambios percibe en el **tiempo de ejecución** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
  - En promedio, el tiempo de ejecución parece ser el mismo para los dos mecanismos de colisión. Esto tiene sentido ya que en ambos casos se carga la misma cantidad de información, y la complejidad temporal de la carga de información, no depende en gran medida del mecanismo de colisión.
- 6) ¿Qué cambios percibe en el **consumo de memoria** al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.
  - Las dos maquinas muestran resultados opuestos en términos de consumo de memoria y esquema de colisiones. En general, los dos métodos almacenan la misma información. Sin embargo, CHAINING permite factores de carga mayores, lo que en teoría reduce el uso de memoria. En ese orden de ideas, si se ven diferencias en el consumo de memoria al modificar el esquema de colisiones y (ignorando la primera máquina), CHAINING consume menos memoria que PROVING.