OBSERVACIONES DE LA PRACTICA 7

Natali Mercado Solórzano 202012682
Gabriel Santiago Villabón Linares 202013898

	Máquina 1	Máquina 2
Procesadores	Intel(R) Core(TM) i5-7200U CPU @2.50GHz 2.20GHz	Intel(R) Core (TM) i7-8750H CPU @2.20GHz
Memoria RAM (GB)	8	16
Sistema Operativo	Windows 10 Pro 64-bits	Windows 10 Pro 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 [ms]
0.30	4712,843	185,869
0.50	4712,861	190,59
0.80	4712,916	205,272

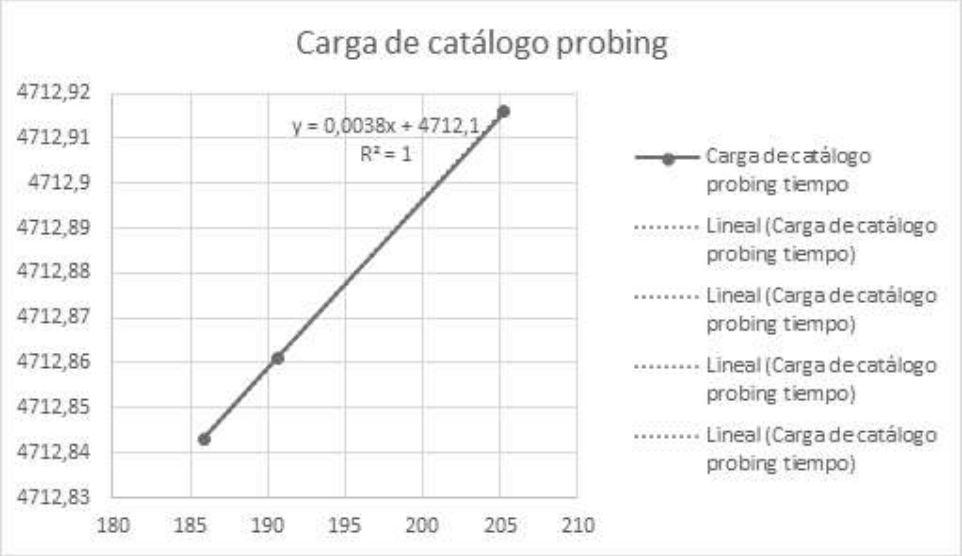
Tabla 2. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando PROBING en la Maquina 1.ñ

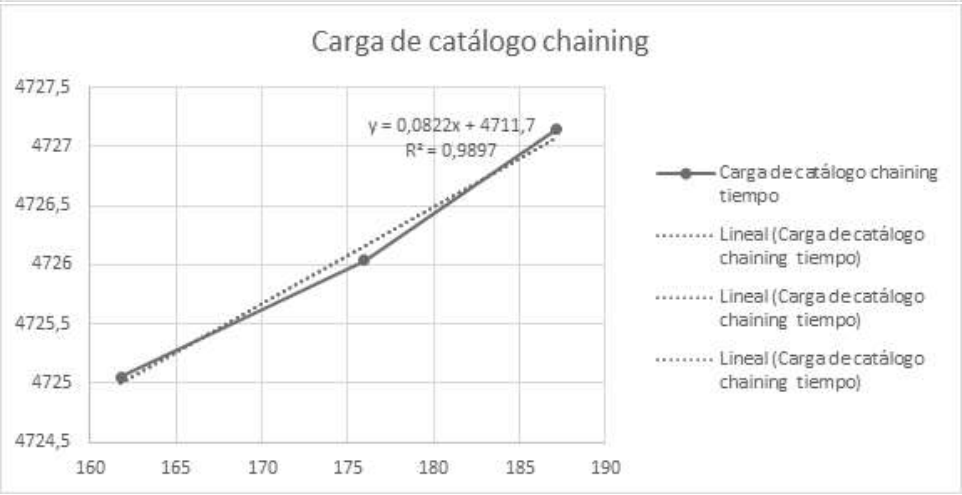
Carga de Catálogo CHAINING		
Factor de Carga (CHAINING)	**Consumo de Datos [kB]**	**Tiempo de Ejecución [ms]**
2.00	4725,049	161,776
4.00	4726,038	175,953
6.00	4727,145	187,088

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	4752.434	109.919
0.50	4752.434	107
0.80	4753.863	117.033

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

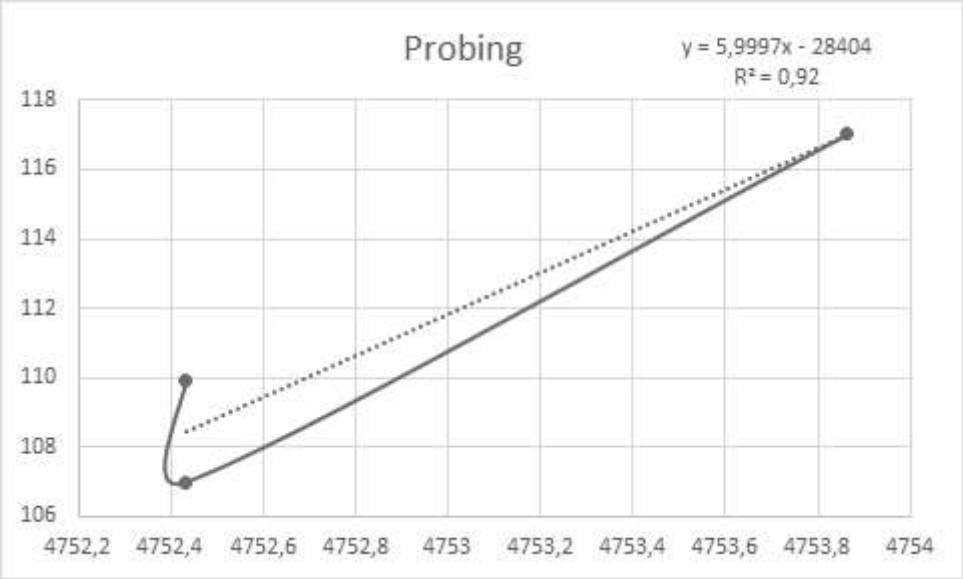
Factor de Carga (CHAINING)	Consumo de Datos [kB]	Tiempo de Ejecución [ms]
2.00	4766.238	102.087
4.00	4766.238	100.883
6.00	4766.238	117.417

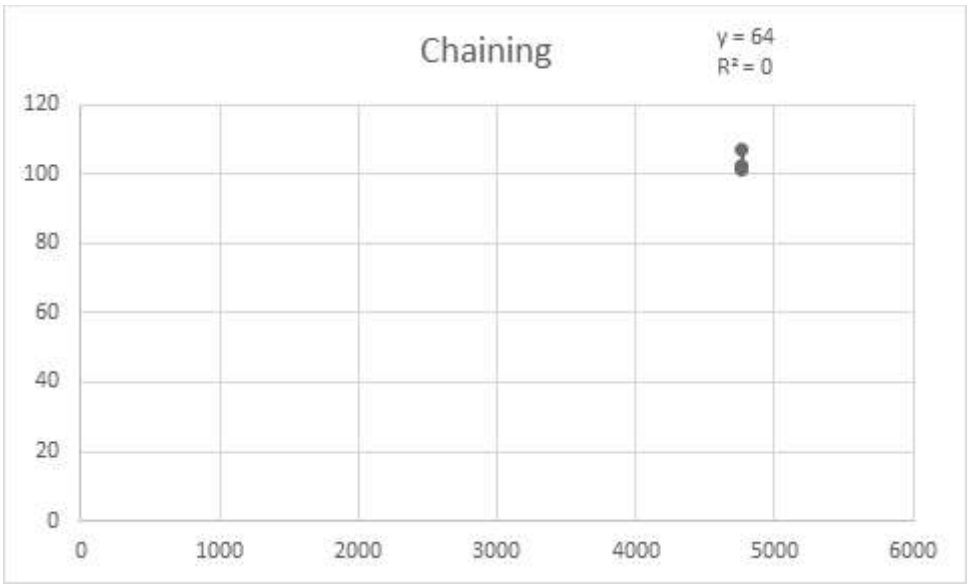
Tabla 5. Comparación de consumo de datos y tiempo de ejecución para carga de catálogo con el índice por categorías utilizando CHAINING en la Maquina 2.

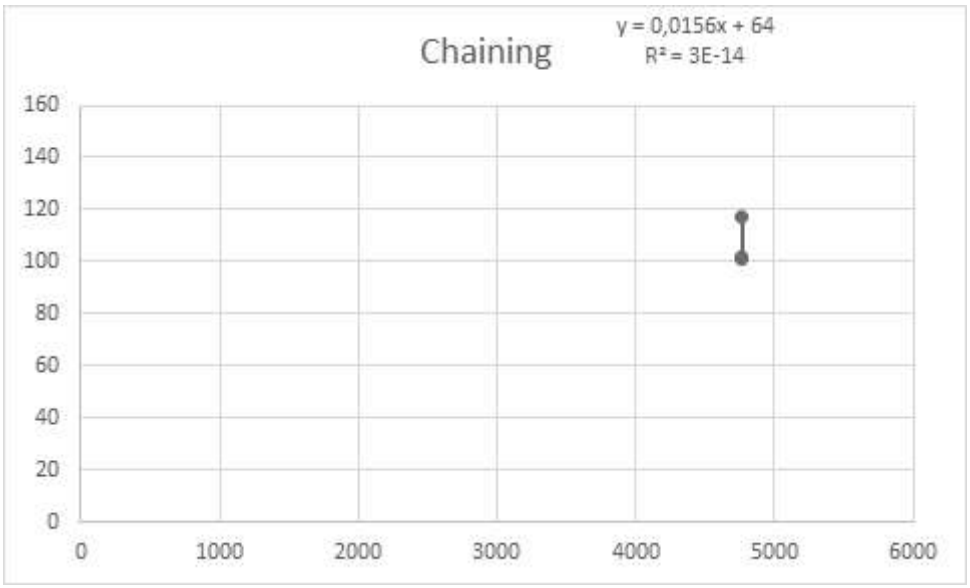
Graficas

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()**?

Time.perf_counter() es más exacto que time.process_time() puesto que funciona como un cronometro que mide el tiempo real que se demora un proceso de duración corta en completarse, en este caso la carga del catálogo. Su exactitud se debe a que tiene la resolución más alta disponible. Por otro lado, time.process_time() retorna la suma del tiempo de CPU del sistema y del usuario durante el proceso, es decir, retorna el tiempo empleado por la computadora para completar el proceso.

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

Son esenciales para determinar los límites dónde se desea empezar y terminar de rastrear las asignaciones de memoria en python, en este caso se limita la prueba al proceso de carga de catalogo.

3) ¿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 aumenta a medida que aumenta el factor de carga, sin embargo, no son cambios significativos.

4) ¿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 aumenta cuando se aumenta el factor de carga pues la relación entre el número de datos y el tamaño de la tabla aumenta, por ende, se ocupa más espacio y hay mayor complejidad espacial.

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.

Los tiempos de ejecucion son más altos con probing que con chaining, sin embargo, la diferencia no es significativa. Lo anterior se explica porque el tiempo de ejecución con linear probing se mantiene medianamente constante si el factor de carga es menor a 1 y con chaining si tenemos una cadena muy larga, la complejidad puede llegar a ser de O(N).

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.

Se ocupa más memoria con linear probing debido a que cada pareja clave-valor ocupa una celda, por ende, hay más complejidad espacial que con chaining. Es importante resaltar que con chaining la complejidad espacial disminuye puesto que en cada celda encontramos listas y por ende tenemos menos celdas. Sin embargo, en los datos reportados se observa lo contario, es decir que están asociados a errores y no concuerdan con la teoría.