

OBSERVACIONES DEL RETO 2

Juan José Osorio (202017020)
Thais Tamaio Ramírez (202022213)

	Máquina 1	Máquina 2
Procesadores	3,1 GHz Intel Core i5 de dos núcleos	Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz 2.50 GHz
Memoria RAM (GB)	8 GB 2133 MHz LPDDR3	4.00 GB (3.90 GB usable)
Sistema Operativo	macOS Big Sur versión: 11.2.1	Windows 10 Home 64-bits

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

Para realizar el análisis de los resultados en términos de memoria y tiempos de ejecución, decidimos medir el tiempo de ejecución y la memoria consumida para el reto 1 y el reto 2, en dos máquinas diferentes. En las siguientes tablas se muestran los resultados al ejecutar la carga de datos y los cuatro requerimientos de los dos retos.

Máquina 1/Reto 1:

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4
Tiempo de ejecución [ms]	44590,65	553,72	2146,94	2458,69	323,83
Memoria consumida [kB]	550848,33	83,41	359,00	707,63	142,41

Máquina 1/Reto 2:

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4
Tiempo de ejecución [ms]	105152,66	539,34	2134,51	2250,07	1603,30
Memoria consumida [kB]	533981,81	76,34	358,59	354,26	238,00

Máquina 2/Reto 1:

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4
Tiempo de ejecución [ms]	66552,93	814,98	3402,35	3873,79	448,99
Memoria consumida [kB]	547903,73	82,79	358,80	707,57	142,39

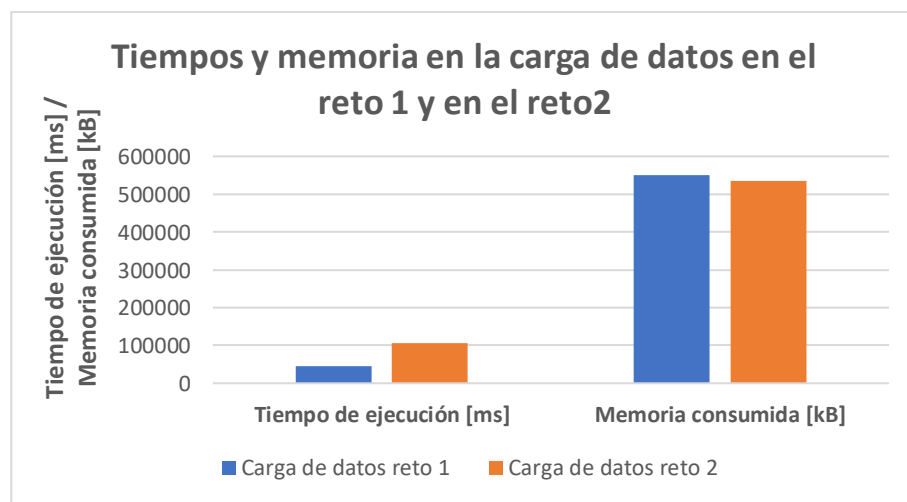
Máquina 2/Reto 2:

	Carga de datos	Req. 1	Req. 2	Req. 3	Req. 4
Tiempo de ejecución [ms]	165261,33	283,73	3249,75	3401,64	2364,84
Memoria consumida [kB]	524410,29	61,17	356,66	353,13	236,19

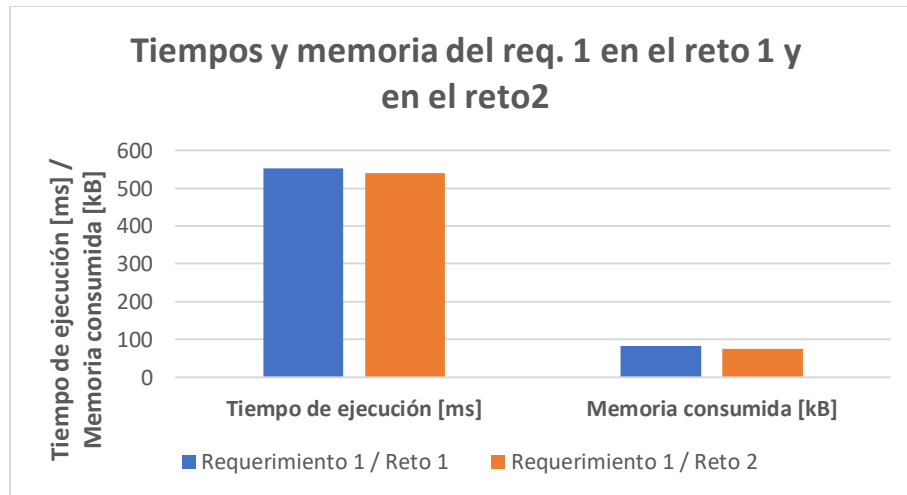
Por medio de las tablas que se mostraron anteriormente, se obtuvieron las siguientes gráficas para la máquina 1 y la máquina 2:

1. Gráficas máquina 1:

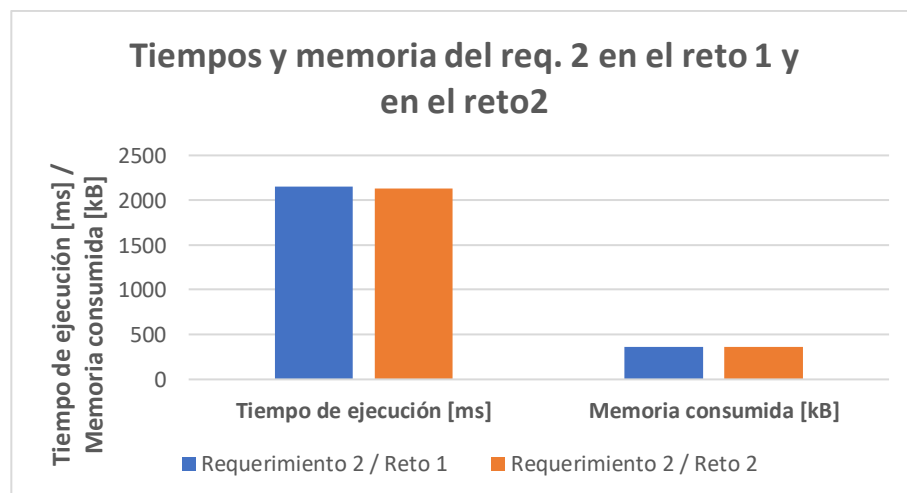
- Carga de datos:



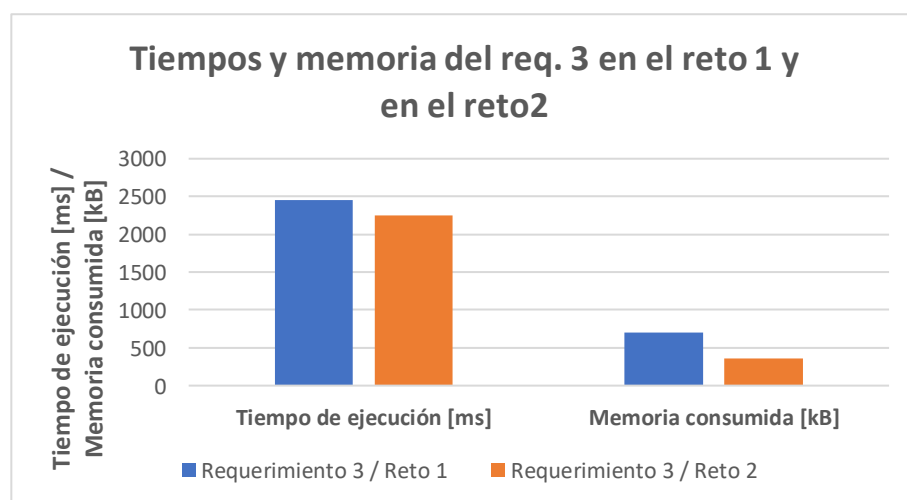
- Requerimiento 1:



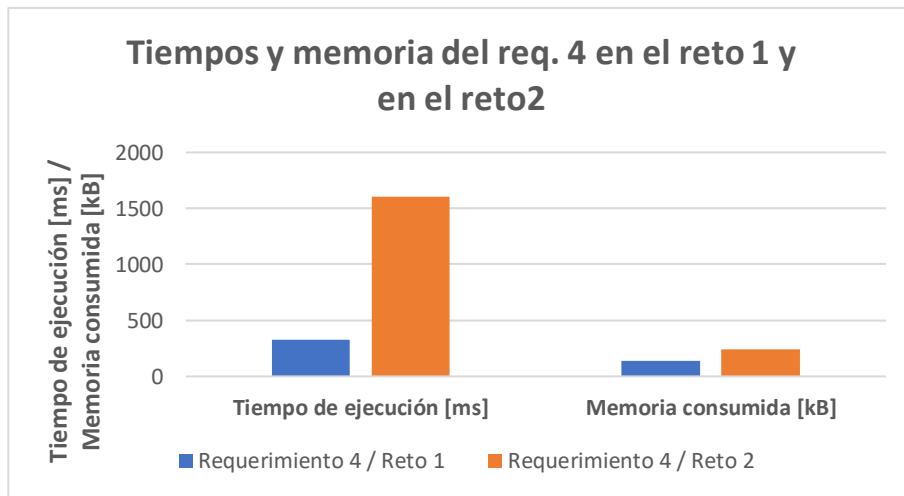
- Requerimiento 2:



- Requerimiento 3:

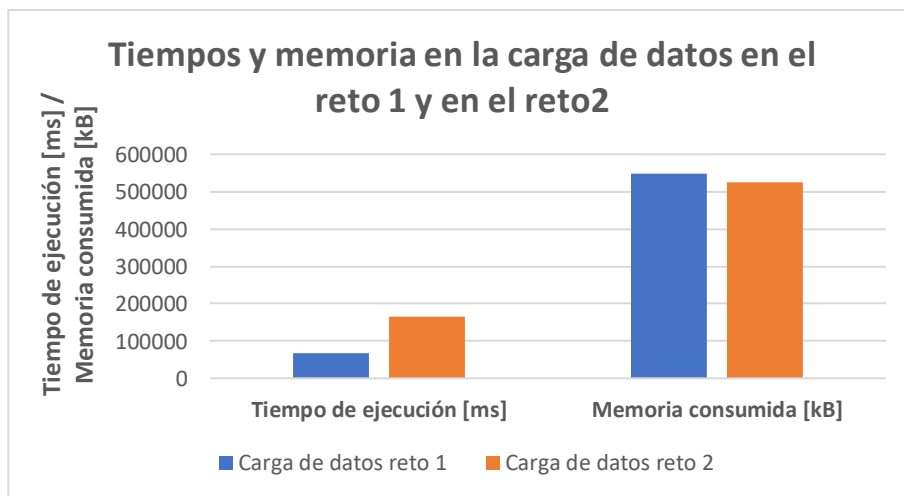


- Requerimiento 4:

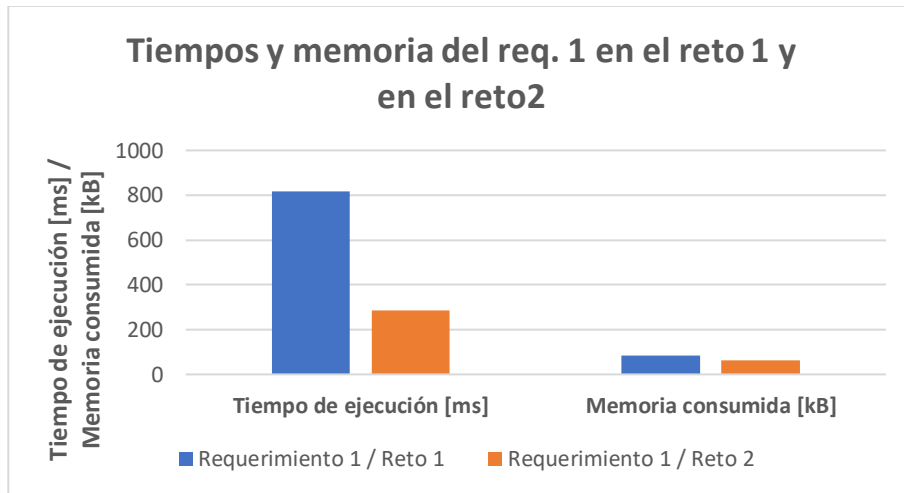


2. Gráficas máquina 2:

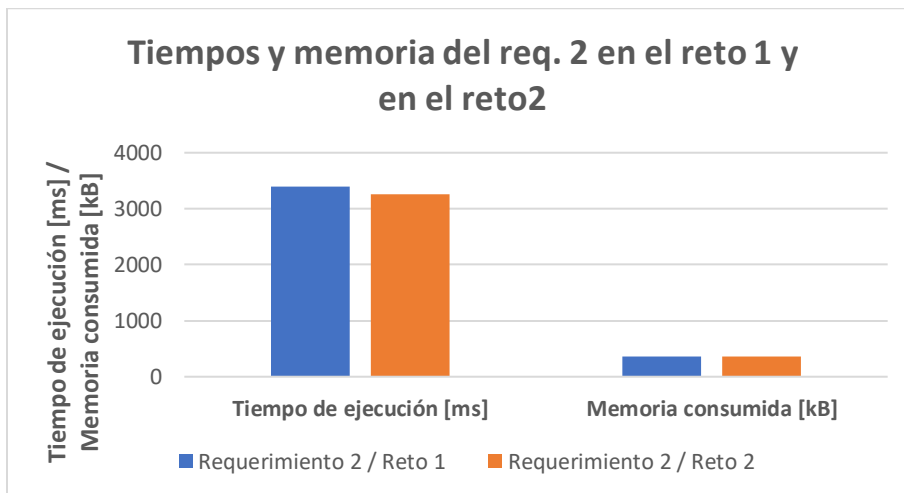
- Carga de datos:



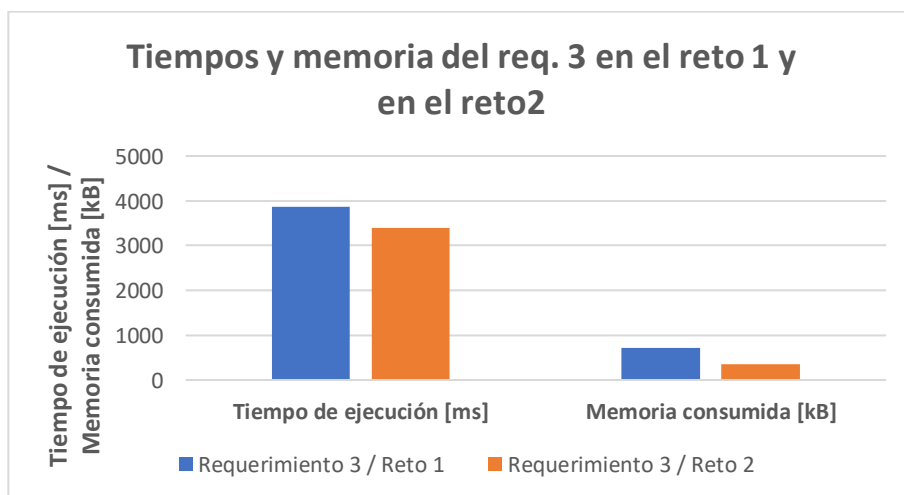
- Requerimiento 1:



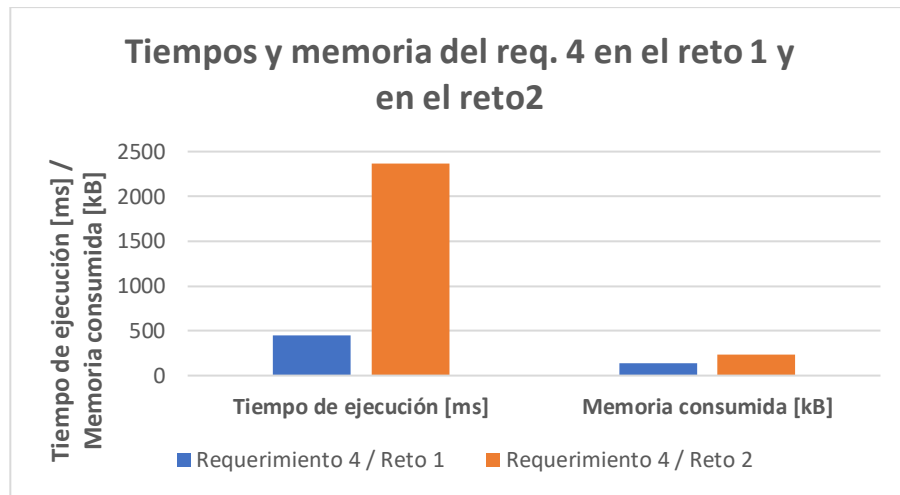
- Requerimiento 2:



- Requerimiento 3:



- Requerimiento 4:



De manera general se puede determinar lo siguiente:

Aunque la diferencia sea mínima, para la mayoría de los requerimientos y en ambas máquinas, se obtuvieron menores resultados en el consumo de memoria para el reto 2, en comparación con el reto 1. Adicionalmente, se puede apreciar que para la carga de los datos los tiempos de ejecución en el reto 2 fueron mucho mayores a los del reto 1. Esto se debe principalmente a la implementación de los mapas, ya que el almacenar todos los datos dentro de éstos es un proceso que tarda más tiempo que almacenarlos dentro de un arreglo.

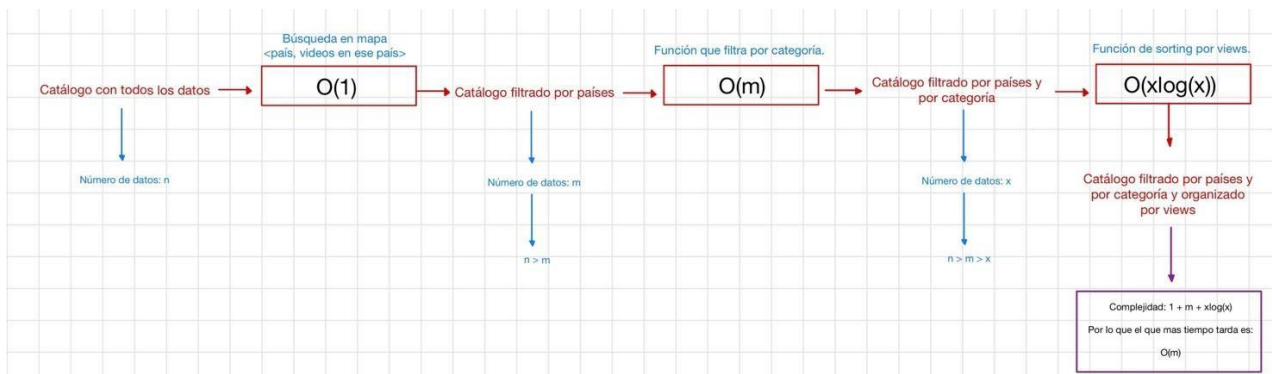
De igual manera, para la implementación de algunos requerimientos, los tiempos de ejecución fueron menores en el reto 2 a comparación del reto 1, esto es producto de que los mapas sean más rápidos para las búsquedas dentro de éstos. No obstante, en el requerimiento 4 se presentó lo contrario, ya que los tiempos de ejecución y el consumo de memoria del requerimiento 4 fueron mayores en el reto 2. Esto puede deberse a que, para este requerimiento, seguimos la recomendación de un profesor al implementar un mapa que tiene como llave un tag y como valor todos los videos asociados a este. Por consiguiente, la creación de este mapa llega a tardar más tiempo y a consumir más memoria que la implementación con listas que se utilizó en el reto 1.

Por otro lado, un aspecto a resaltar es que las diferencias entre una implementación con listas y otra con mapas pueden llegar a ser mínimas debido a la cantidad de datos que contiene el archivo CSV. Si se hubieran utilizado millones de videos, los tiempos de búsqueda serían menores para los mapas y mayores para las listas y los tiempos de carga sería mayores para los mapas y menores para las búsquedas. Finalmente, sobre la variedad de estructuras de datos en nuestro programa, se escogieron únicamente tablas

de hash de tipo PROBING ya que consumen menos memoria puesto que, a diferencia de las tablas de tipo CHAINING, estas no crean una lista encadenada cuando ocurre una colisión, sino que buscan un espacio disponible. Además, se eligió factor de carga 0.5 para mantener la complejidad lo más constante posible. Por otro lado, cuando hablamos de listas utilizamos ARRAY's ya que consumen menos memoria (no guardan el siguiente nodo) y además es más fácil llegar a un elemento puesto que guardan los elementos en posiciones contiguas en memoria. Para cada arreglo la función de comparación varía dependiendo del requerimiento.

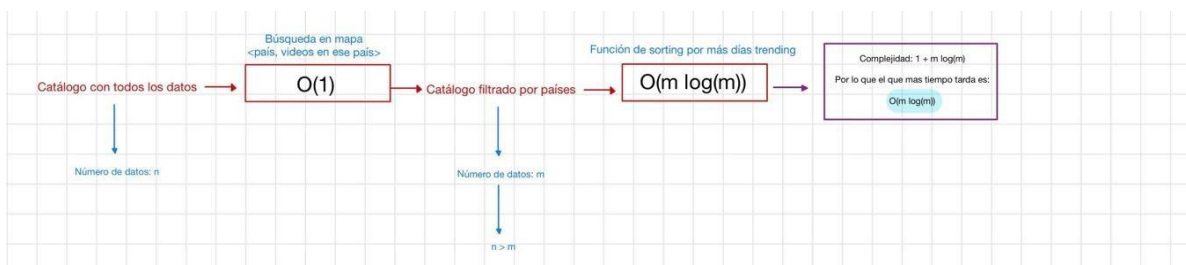
De manera singular se puede determinar lo siguiente:

1. **Carga de datos:** Se obtuvo un mayor tiempo de ejecución y un mayor consumo de memoria para el reto 2, en comparación con los resultados del reto 1.
 2. **Requerimiento 1:** Se obtuvo un menor tiempo de ejecución y un menor consumo de memoria para el reto 2, en comparación con los resultados del reto 1.
- Análisis de complejidad temporal:



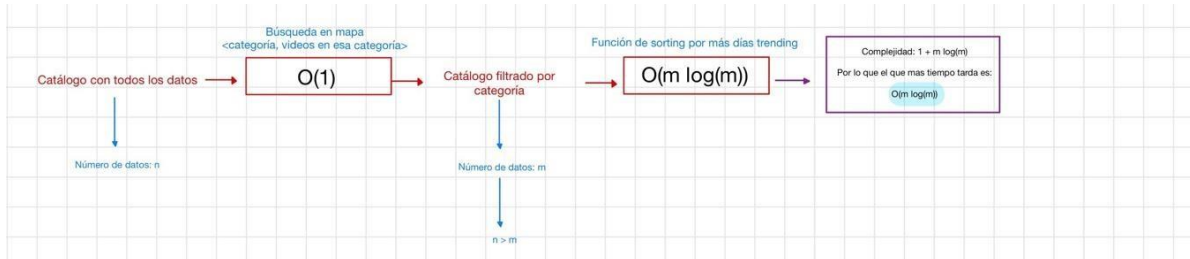
3. **Requerimiento 2:** Se obtuvo un menor tiempo de ejecución para el reto 2, en comparación con los resultados del reto 1. No obstante, el consumo de memoria entre ambos retos fue muy similar.

- Análisis de complejidad temporal:



4. **Requerimiento 3:** Se obtuvo un menor tiempo de ejecución y un menor consumo de memoria para el reto 2, en comparación con los resultados del reto 1.

- Análisis de complejidad temporal:



5. **Requerimiento 4:** Se obtuvo un menor tiempo de ejecución y un menor consumo de memoria para el reto 1, en comparación con los resultados del reto 2.

- Análisis de complejidad temporal:

