

# Análisis de Resultados

Jose David Florez Ruiz Cod 202121507

Santiago Castro 202122158

Req. 2	Req. 3
Jose Florez	Santiago Castro

	Máquina 1	Máquina 2
Procesadores	AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz	Intel(R) Core(TM) i3-7100U @ 2.40GHz
Memoria RAM (GB)	16.0 GB	8.0 GB
Sistema Operativo	Windows 11 Home	Windows 10 Home

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

## Maquina 1

### Resultados

#### Tiempo de ejecución

% Carga	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5	Req. 6
80	30.75	1.92	145.87	1.99	5392.89	30.75
100	52586.22	4.08	242.02	44.42	0.46	26.65

#### Consumo de memoria

% Carga	Carga
80	1161076.18
100	1417488.85

## Maquina 2

### Resultados

#### Tiempo de ejecución

% Carga	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5	Req. 6
80	173424.7	13.91	483.05	62.92	11776.70	61128.79
100	95857.20	29.69	127.77	15.23	111.36	76178.46

#### Consumo de memoria

% Carga	Carga
80	1211453.61
100	1417488.85

## Análisis de complejidad:

Carga:

Se va se tiene que recorrer cada archivo una vez, mientras que se recorren los archivos los datos se van agregando a distintos maps, dependiendo de la necesidad de cada requerimiento. La complejidad es de  $O(n)$  por cada archivo.

Requerimiento 1:

Con un map con llaves de años, que almacena una lista, se recorre toda la lista por lo que la complejidad es de  $O(m)$ .

Requerimiento 2:

En un map con la llave siendo popularidad del artista y el valor una lista con los ID de artistas. Se recorre toda la lista para conseguir el nombre de cada artista, por lo que la complejidad es de  $O(m)$

### Requerimiento 3

En un map con la llave siendo popularidad de la canción y el valor una lista con los ID de canción. Se recorre toda la lista para conseguir el nombre de cada canción, por lo que la complejidad es de  $O(m)$

### Requerimiento 4:

En un map con la llave del artista y el valor el ID de la canción se recorre todas por todas las llaves, para crear una lista filtrada, luego se ordena con un ordenamiento shell, además se buscan los nombres de los álbumes, por lo que la complejidad es de  $O(3m \log m)$

### Requerimiento 5:

En un map que las llaves son los álbumes y los valores son los id de artista, se recorre todas las llaves para hacer el conteo de tipos de álbumes, por lo que la complejidad es  $O(m)$ . Los loops de primeros tres y últimos tres no se tienen en cuentas porque su complejidad es de solo  $O(6)$ .

### Requerimiento 6:

En un map donde las llaves son los artistas y el valor una lista con los ID de canción. Se crea una lista filtrada ( $O(m)$ ), luego se ordena con un ordenamiento shell, además se buscan los nombres de los álbumes( $O(m)$ ), por lo que la complejidad es de  $O(3m \log m)$

## Comparación Análisis de complejidad con el Reto 1:

### Complejidades Reto 1:

Carga de datos -  $O(n)$

Requerimiento 1 -  $O(n \log(n))$

Requerimiento 2 -  $O(n \log(n))$

Requerimiento 3 -  $O(n \log(n))$

Requerimiento 4 -  $O(n)$

Requerimiento 5 -  $O(n)$

Requerimiento 6 -  $O(n \log(n))$

La mayor diferencia entre ambos retos, es que gracias a los mapas se puede almacenar información en base a un criterio por lo que buscar información es mucho más rápido, es sin mencionar que la búsqueda de una llave es constante. Mientras que en el mejor caso la complejidad del reto 1 es de  $O(n \log n)$ .