

OBSERVACIONES DEL RETO 2

Integrantes individuales:

Cristian Camilo Cortes Moreno Cod. 202011908

(cc.cortesm1@uniandes.edu.co)

Natalia Villegas Calderón Cod. 202113370

(n.villegasc@uniandes.edu.co)

Requerimientos individuales:

Requerimiento 2: Natalia Villegas Calderón

Requerimiento 3: Cristian Camilo Cortes Moreno

Analisis de complejidad → $O()$ + Tiempo de ejecución:

Porcentaje de la muestra [pct]	Carga
5.00%	187854,48
10.00%	1454620,57
50.00%	2478352,39
80.00%	95366854,59
100.00%	187854,48

Espacio de la carga [KB]

Ambientes de pruebas:

	Máquina 1	Máquina 2
Procesadores	Chip M1	Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz
Memoria RAM (GB)	8.0 GB	16 GB
Sistema Operativo	Mac OS Big Sur	Windows 10 Home Single Language 21H2

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

Máquina 1:

Porcentaje de la muestra [pct]	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6
5.00%	5,59	4,59	4,08	3,02	3,16	1,52
10.00%	6,76	4,52	4,96	3,56	3,28	1,55
50.00%	10,68	4,62	12,57	4,1	4,44	2,63

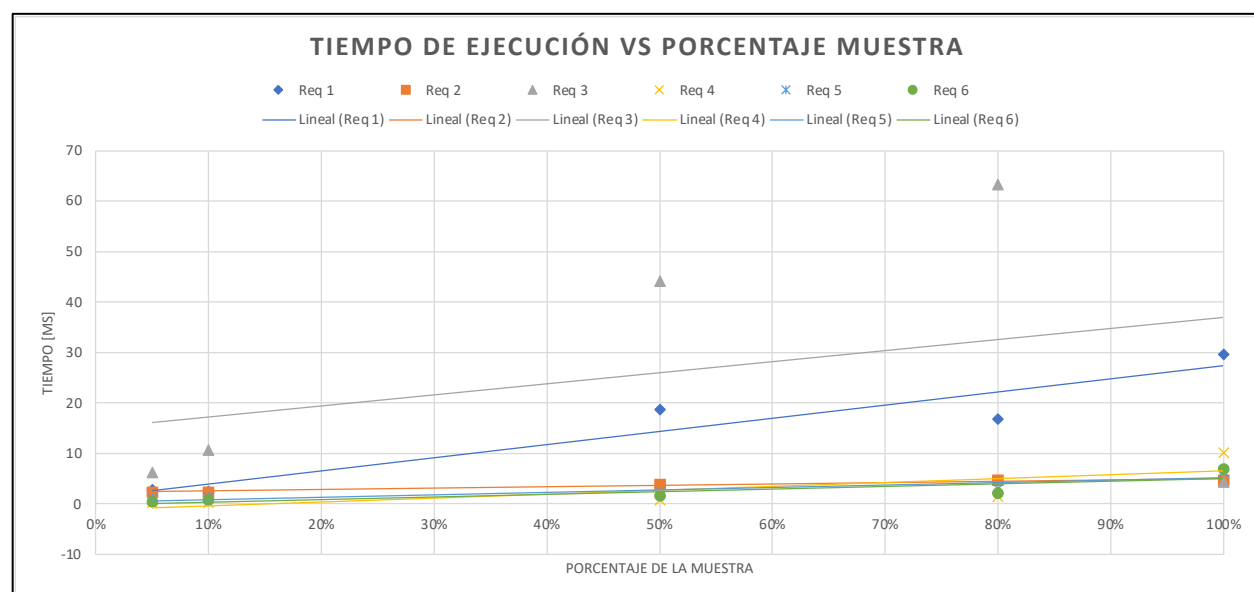
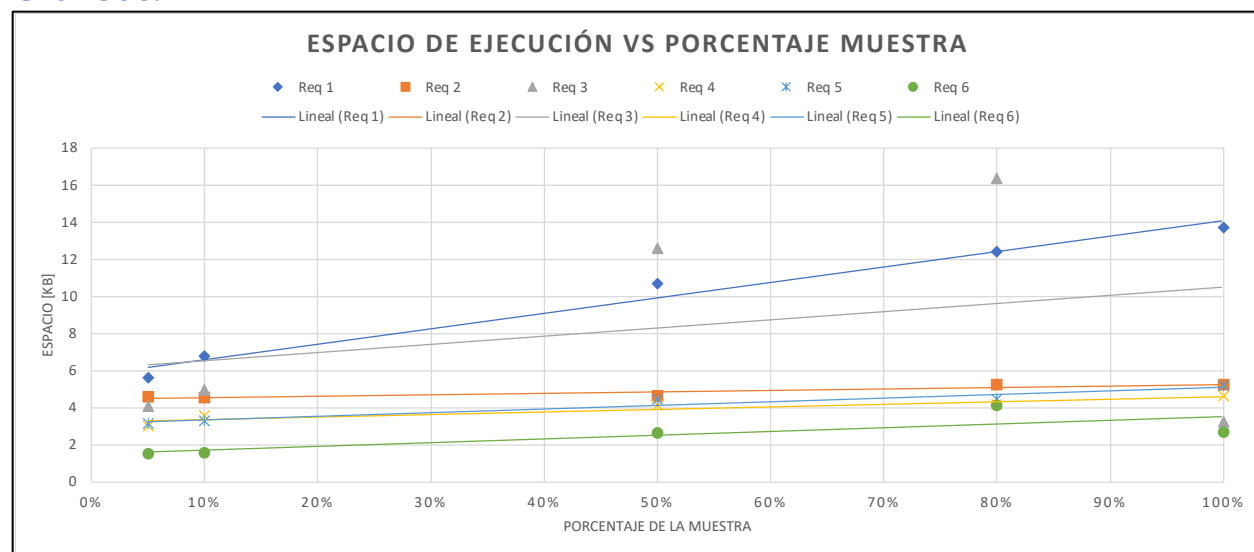
80.00%	12,4	5,25	16,34	4,15	4,45	4,14
100.00%	13,71	5,25	3,23	4,64	5,17	2,68

Memoria consumida vs porcentaje muestra Máquina 1

Porcentaje de la muestra [pct]	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6
5.00%	2,81	2,38	6,27	0,18	0,86	0,42
10.00%	2,46	2,35	10,72	0,27	1,28	0,69
50.00%	18,72	3,9	44,14	0,77	2,27	1,6
80.00%	16,85	4,64	63,35	1,43	3,04	2,11
100.00%	29,65	4,67	4,34	10,12	6,34	6,89

Tiempo de ejecución vs porcentaje muestra Máquina 1

Gráficas:



Máquina 2:

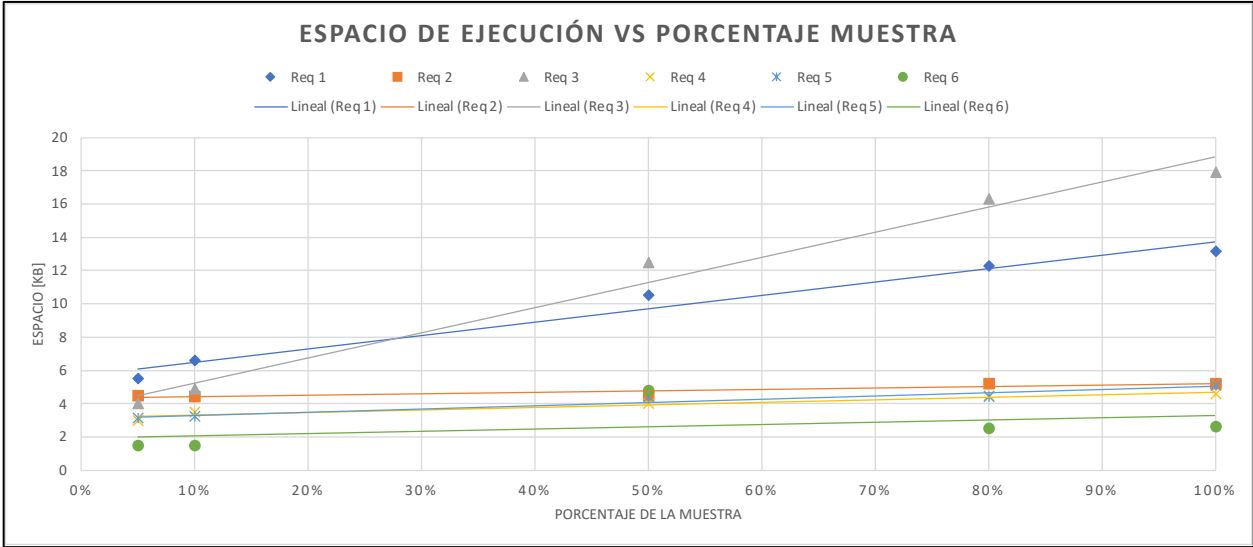
Porcentaje de la muestra [pct]	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6
5.00%	5,52	4,5	4	2,98	3,12	1,5
10.00%	6,6	4,4	4,9	3,5	3,25	1,5
50.00%	10,5	4,5	12,5	4	4,4	4,8
80.00%	12,3	5,18	16,3	4,5	4,4	2,53
100.00%	13,16	5,18	17,89	4,57	5,12	2,64

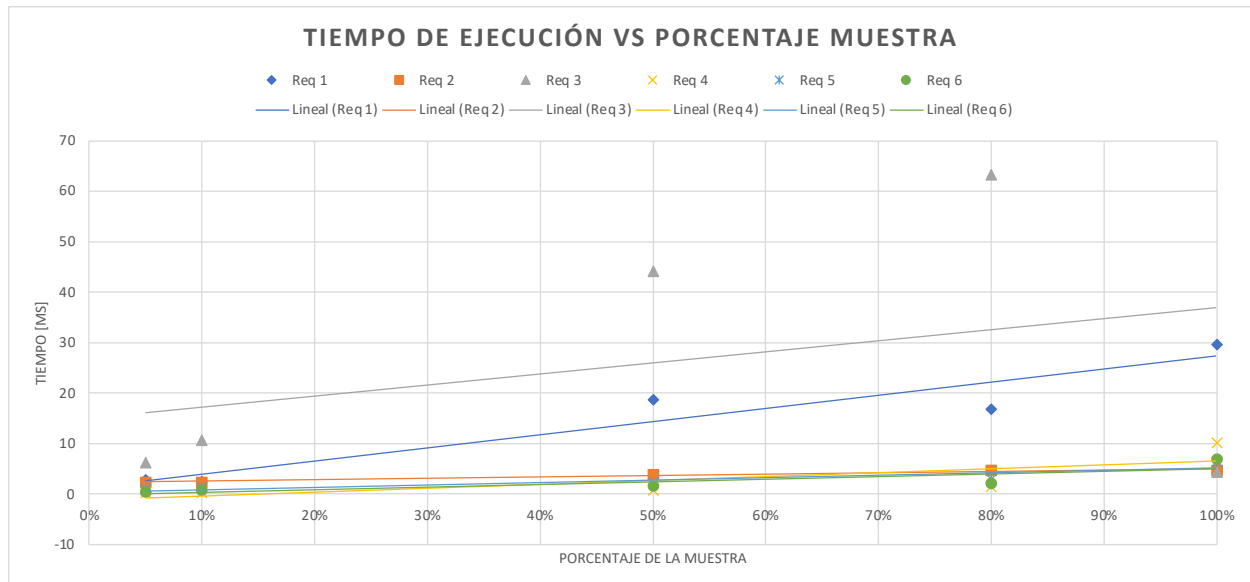
Memoria consumida vs porcentaje muestra Máquina 2

Porcentaje de la muestra [pct]	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6
5.00%	5,62	1,41	4,37	0,25	0,4	0,47
10.00%	6,26	1,73	8,1	0,4	1,66	0,61
50.00%	14,3	1,63	45,19	0,94	1,84	0,77
80.00%	19,73	2,14	61,59	1,35	1,72	1
100.00%	22,55	1,56	77,5	2,51	2,19	1,23

Tiempo de ejecución vs porcentaje muestra Máquina 2

Gráficas:





Requerimiento 1:

M = Tamaño Lista de álbumsEncontrados

$O(M \log(M))$

En el requerimiento 1 consultamos en un mapa ('albumsByYear') donde cada llave es un año y su valor tiene un array con todos los álbumes que pertenecen a ese año. La consulta tiene una complejidad $O(1)$ posteriormente se organizan los álbumes encontrados alfabéticamente, proceso que le corresponde una complejidad de $\log(\text{álbumsEncontrados})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar los álbumes encontrados (m tamaño de los álbumes encontrados).

Requerimiento 2:

K = Tamaño Lista de artistasEncontrados

$O(K \log(K))$

En el requerimiento 2 consultamos en un mapa ('artistsByPopularity') donde cada llave corresponde a un nivel de popularidad y su valor tiene un array con todos los artistas que tienen ese nivel de popularidad. La consulta tiene una complejidad $O(1)$ posteriormente se organizan los artistas encontrados por su cantidad de followers, proceso que le corresponde una complejidad de $\log(\text{artistasEncontrados})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar los artistas encontrados.

Requerimiento 3:

C = Tamaño Lista de cancionesEncontradas

$O(C \log(C))$

En el requerimiento 3 consultamos en un mapa ('tracksByPopularity') donde cada llave corresponde a un nivel de popularidad y su valor tiene un array con todas las canciones que tienen ese nivel de popularidad. La consulta tiene una complejidad $O(1)$ posteriormente se organizan las canciones encontradas por su duracion, proceso que le corresponde una complejidad de cancionesEncontradas $\log(\text{cancionesEncontradas})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar las canciones encontradas.

Requerimiento 4:

C = Tamaño Lista de cancionesEncontradas

$O(\text{Clog}(C))$

En el requerimiento 4 consultamos en un mapa ('artistsByName') donde cada llave corresponde al nombre de un artista y su valor es un diccionario con la información del artista. La consulta tiene una complejidad $O(1)$ posteriormente se consulta en un mapa interno ('tracksByCountry') donde cada llave corresponde al código de un país y su valor es una lista con todas las canciones en ese país. Esta consulta también tiene una complejidad $O(1)$. Después organizamos las canciones por su popularidad y en su defecto la duración. Este proceso le corresponde una complejidad de cancionesEncontradas $\log(\text{cancionesEncontradas})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar las canciones encontradas.

Requerimiento 5:

M = Tamaño Lista de álbumsEncontrados

$O(\text{Mlog}(M))$

En el requerimiento 5 consultamos en un mapa ('artistsByName') donde cada llave corresponde al nombre de un artista y su valor tiene un array con todos los álbumes que pertenecen a ese artista. La consulta tiene una complejidad $O(1)$ posteriormente se extrae diversa información del diccionario acciones con complejidades $O(1)$. Después organizamos los álbumes por su año de salida. Este proceso le corresponde una complejidad de albumesEncontrados $\log(\text{albumesEncontrados})$. Después se recorren los álbumes del artista extrayendo sus canciones más populares con complejidad $O(\text{albumesEncontrados})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar los álbumes encontrados.

Requerimiento 6:

C = Tamaño Lista de cancionesEncontradas

$O(\text{Clog}(C))$

En el requerimiento 6 consultamos en un mapa ('artistsByName') donde cada llave corresponde al nombre de un artista y su valor es un diccionario con la información del artista. La consulta tiene una complejidad $O(1)$ posteriormente se consulta en un mapa

interno ('tracksByCountry') donde cada llave corresponde al código de un país y su valor es una lista con todas las canciones en ese país. Esta consulta también tiene una complejidad $O(1)$. Después organizamos las canciones por su popularidad y en su defecto la duración. Este proceso le corresponde una complejidad de cancionesEncontradas $\log(\text{cancionesEncontradas})$. Por lo que la complejidad del requerimiento se resume en complejidad de organizar las canciones encontradas.

Análisis con Reto 1:

Requerimiento	Tiempo (Large)		Complejidad	
	Reto 1	Reto 2	Reto 1	Reto 2
Req 1	68,72		$O(n\log(n))$	M = Tamaño Lista de álbumsEncontrados $O(M\log(M))$
Req 2	0,3		$O(n)$	K = Tamaño Lista de artistasEncontrados $O(K\log(K))$
Req 3	0,68		$O(n)$	C = Tamaño Lista de cancionesEncontradas $O(C\log(C))$
Req 4	0,97		N = Tamaño Lista de Artistas $O(\log(n))$	C = Tamaño Lista de cancionesEncontradas $O(C\log(C))$
Req 5	5,25		$O(\log(n))$	M = Tamaño Lista de álbumsEncontrados $O(M\log(M))$
Req 6 (Bono)	538,37		$O(n\log(n))$	C = Tamaño Lista de cancionesEncontradas $O(C\log(C))$

En el requerimiento 1 del Reto 1 se buscaba en un rango de años la consulta era binaria para la búsqueda de la posición de la posición 1 y lineal, por diversos problemas en la implementación para la búsqueda de la posición del final del rango después se hacía

una sublista de esos datos. Por otra parte, en el reto 2 en este requerimiento, solo hay la consulta de un año con complejidad $O(1)$ y el Sort de los álbumes que allí pertenecen, siendo en este caso un requerimiento con menor complejidad.

En el caso del requerimiento 2 del Reto 1 la complejidad era $O(n)$ o en su defecto la cantidad de artistas que deseábamos listar, esto gracias a que en la carga los datos ya se encontraban organizados y solo toca extraerlos. Mientras que en el Reto 2 hacemos una consulta $O(1)$ y un Sort de los Artistas encontrados, encontrando en esta casi allí mayor complejidad.

En el requerimiento 3 del Reto 1 la complejidad es similar al requerimiento anterior, puesto que la complejidad se basa en la cantidad de datos a listar. De igual manera, para el caso del reto 2 la complejidad se encuentra en el Sort de los datos encontrados dentro de la consulta $O(1)$ en el mapa.

En el requerimiento 4 del Reto 1 la complejidad está en la búsqueda con complejidad $\log(n)$ del artista y posteriormente la búsqueda lineal con complejidad $O(\text{cancionesDelArtista})$ de las canciones de este artista para encontrar las que pertenezcan al país de distribución deseado. En el caso de requerimiento 2, tan la búsqueda de las canciones y del artista es $O(1)$ gracias a la complejidad de las tablas de hash, pero se hace un Sort de este pequeño grupo de datos, siendo este el que le aumenta la complejidad.

En el requerimiento 5 del Reto 1 la complejidad está en la búsqueda con complejidad $\log(n)$ del artista para después extraer sus datos internos de los álbumes, lo que corresponde a $O(1)$. Para el caso del reto 2 los álbumes se encuentran a una complejidad de $O(1)$ y posteriormente se recorren linealmente para extraer la información de interés, siendo esta la complejidad se define este requerimiento.

En el requerimiento 6 del Reto 1 la complejidad es similar a la del requerimiento 1 donde se tiene que buscar un rango con complejidad logarítmica para una posición y lineal para la otra. En el caso del reto 2 para este requerimiento el comportamiento es similar a la del requerimiento 4 con dos consultas $O(1)$ y finalmente un Sort de Los datos encontrados.

NOTA: En el caso del reto 2 los Sorts no implican un aumento considerable, puesto que los datos que se encuentran son considerablemente pequeños, por lo que no afectan la velocidad de las consultas. Por otra parte, el uso de tracemalloc para revisar el uso de la memoria implica un aumento relevante en la ejecución de los requerimientos, por lo que los datos de tiempos recogidos no son fieles a la realidad.