

Observaciones Reto 3

• Análisis de complejidad

Requerimiento 1:

Para este requerimiento, se halló primero la altura del árbol RBT que se utilizó, así como la cantidad de elementos en el mismo. La complejidad de esto es $O(2 \log n)$ para cada método, donde n es la cantidad de elementos del árbol (este n depende de la característica especificada por el usuario y de la cantidad de datos que se estén cargando del archivo csv). Adicionalmente, para este requerimiento se halló la cantidad de reproducciones y el número de artistas únicos, dado un rango para la característica que deseara el usuario. En este caso, la complejidad para la cantidad de reproducciones tuvo una complejidad de $O(2 \log n)$ (se utilizó *om.values()* y *om.size()*), y la complejidad para hallar el número de artistas únicos fue de $O(\log n)$ pues se utilizó la función *om.size()* de nuevo, pero para un mapa distinto que en el caso anterior.

Requerimiento 2:

Para este requerimiento se utilizó de igual manera la función *om.values()*, sin embargo, se tuvo que ejecutar la función *ListReps_to_HashPistasUnicas(lista_listas_reps)*, la cual accede a cada reproducción de la lista de listas, lo cual incrementó la complejidad del requerimiento en $O(n)$. Posteriormente, la función *om.valueSet()* también se utilizó, lo cual genera una complejidad de $O(n)$. Además, se utilizó la función *getValue()* la cual tiene una complejidad de $O(1)$.

Requerimiento 3:

Este requerimiento es bastante similar al Requerimiento 2, en términos de funcionamiento y de complejidad. También se tuvo que hacer uso de una función llamada *Lista_unicas_Instrumentalness(Reproducciones_Rango_Instrumentalness)*, de manera que se pudiera acceder a cada reproducción que pertenecía al rango de Instrumentalness determinado. Esta función tuvo una complejidad de $O(n)$ donde n es el número de datos de la lista. Adicionalmente, para completar el requerimiento se utilizó la función *om.valueSet()*, la cual tiene una complejidad $O(n)$. Luego se hicieron algunas operaciones con complejidad $O(1)$, las cuales no incrementaron mucho la complejidad del requerimiento.

Requerimiento 4:

La primera parte de este requerimiento consiste en agregar un género que desee el usuario. Esta primera parte tiene una complejidad $O(1)$, dado que se trata de ponerle un nombre al nuevo género, y especificar el rango de tempo para el mismo. Posteriormente, para hacer la consulta de los géneros que desee el usuario, se generó una lista que va agregando en la última posición ($O(1)$) los géneros de consulta. A partir de esta lista, se ejecuta la función *req4()*, la cual genera un mapa de respuesta con los géneros como llave y las reproducciones que pertenecen a cada género como valor, con una complejidad $O(n)$ donde n es la cantidad de reproducciones a recorrer. Al tener la lista de reproducciones para cada género, se le calcula el *size()*, para así obtener el número total de

reproducciones por género. Esto tiene una complejidad de $O(n)$ donde n es el número de reproducciones para el género determinado.

Requerimiento 5:

Para este requerimiento, se utilizan diversos mapas ordenados, tablas de hash y estructuras, de manera que se pueda calcular lo que se pide en este caso. Podemos ver que se ejecutan recorridos dobles, e incluso triples, lo cual incrementa significativamente la complejidad del requerimiento. Para la segunda parte, se utilizan funciones como *om.values()* para obtener los valores de las respuestas, así como el uso de la función *size()*, la cual tiene una alta complejidad. Con esto, la complejidad de este requerimiento es lineal.

• Análisis de tiempo de ejecución y consumo de memoria

Carga del catálogo:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
144893.145	665273.1875

Requerimiento 1:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
4912.759	20.099

Requerimiento 2:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
816.0156	26.1376

Requerimiento 3:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
662.945	24.25

Requerimiento 4:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
10273.899	6588.7841

Requerimiento 5:

Tiempo de Ejecución [ms]	Consumo de Datos [kB]
21776.422	1078.123

Como podemos ver, el tiempo de ejecución es relativamente bajo para todos los requerimientos; sin embargo, el requerimiento 4 sube un poco con respecto a los demás. Esto se puede deber a la cantidad de operaciones que se ejecutan en dicho requerimiento, como se explica en el literal anterior. Además, podemos ver cómo el consumo de datos para los requerimientos no es muy alto, excepto para el requerimiento 4 y el 5, los cuales consumen mucha más memoria dadas las operaciones mencionadas.