

## **Análisis de complejidad Reto #2**

Para el desarrollo de todos los requerimientos en este reto, se decidió utilizar el TAD map, con una estructura de datos perteneciente a las tablas de Hash, y utilizando un mecanismo de colisiones linear probing. Sin embargo, a la hora de utilizar algunos ordenamientos, se optó por implementarlos con el TAD lista, y una estructura de datos “SINGLE\_LINKED”.

### **Requerimiento 1:**

El análisis de complejidad para este requerimiento se basa en el map y ordenamiento utilizado, donde al entrar al map 'VideosByCategory', se evidencia una complejidad  $O(k)$ . Luego, al utilizar el ordenamiento que en este caso corresponde al de Merge Sort, su complejidad viene dada por  $O(n \log(n))$ , teniendo en cuenta que este  $n$  corresponde al número de videos en la lista del país que se pide. Por lo tanto, la complejidad de este requerimiento es  $O(k)$  junto con  $O(n \log(n))$ .

### **Requerimiento 4:**

La complejidad para este requerimiento se basa en el map, el ordenamiento utilizado y además en los dos recorridos que se hacen tanto en la lista de videos como en la lista de tags. En ese sentido, se conoce que la complejidad de entrar a un map es  $O(k)$  y la complejidad del ordenamiento utilizado es  $O(n \log(n))$ , correspondiente a Merge Sort. Además, la complejidad de los recorridos viene dada por  $O(n)$ , donde  $n$  corresponde a la lista de videos, y  $O(n_1)$  donde  $n_1$  representa la lista que contiene los tags. De ese modo, se tiene que la complejidad temporal de este requerimiento es  $O(n) * O(n_1) * O(k)$  y se le añade la complejidad del ordenamiento, es decir,  $O(n \log(n))$ .

### **Requerimiento 2:**

Ivan Camilo Ballén Méndez

En este caso para el requerimiento 2 se tiene en cuenta la entrada al map 'VideosByCountry', donde su complejidad es de  $O(k)$ . Luego, se recorre la lista de videos perteneciente al país elegido, por lo que su complejidad viene dada por  $O(n_1)$ , teniendo en cuenta que esta  $n_1$ , no corresponde al numero total de datos en el mapa, sino al numero de datos en la lista de videos. Así mismo, la complejidad del ordenamiento debe ser añadida, la cual es de  $O(n \log(n))$ , evidenciando que  $n$  es el número de datos que tiene la lista de trabajo. De acuerdo con lo anterior, la complejidad del requerimiento viene dada por  $O(k) * O(n_1)$  junto a  $O(n \log(n))$ .

### **Requerimiento 3:**

María José Sáenz Rodríguez

El análisis de complejidad para el requerimiento es muy similar al anterior, puesto que al entrar en el map 'VideosByCat\_id', la complejidad de esto sigue siendo  $O(k)$  y al realizar el

recorrido la complejidad viene dada por  $O(n_1)$ . Sin embargo, este  $n_1$  corresponde al número de elementos que se encuentran en la lista de videos según el id de la categoría predeterminada. Por último, se tiene en cuenta la complejidad del ordenamiento la cual corresponde a  $O(n \log(n))$ , donde  $n$ , pertenece al número de elementos en la lista de trabajo. Finalmente, la complejidad del requerimiento 3 viene dada por  $O(k) * O(n_1)$  junto a la complejidad del ordenamiento  $O(n \log(n))$ .

### Análisis de rendimiento (Tiempo de ejecución)

Requerimiento	Solo listas [ms]	Listas con mapas [ms]	Diferencias [ms]
1	0,048	0,036	0,012
2	194305,175	27395,236	166909,939
3	88939,541	12764,056	76175,485
4	0,0033	0,0031	0,0002

Para concluir nuestro análisis, se utilizó la función `perfcounter()` de la librería `time` para realizar unas pruebas rápidas sobre el reto 1 y el reto 2 con el 20% de los datos. Se probaron cada uno de los requerimientos y se puede ver con claridad en la tabulación que el orden de complejidad temporal presenta una disminución en el reto 2 en comparación con el reto 1. Esta diferencia es mucho más evidente en aquellos requerimientos donde se usaron varios mapas para ahorrar los recorridos de filtros que se hacían en el reto 1. Con esto, ratificamos los órdenes de complejidad teóricos. En cuanto a la complejidad temporal, consideramos que no es necesaria realizar pruebas debido a que es claro que las listas del reto 1 junto con más estructuras de datos y más listas. Por ende, es fácil ver que el orden de complejidad espacial aumentó, pero trajo como beneficio una disminución en el orden de complejidad temporal.