**Documento de análisis Reto 2**

Estudiante 1: Pablo Alejandro Guatibonza Briceño| Requerimiento 2

p.guatibonza@uniandes.edu.co

202014393

Estudiante 2: Kaled Andres Correal | Requerimiento 3

k.correal@uniandes.edu.co

201920829

Requerimiento 1(Grupal)

1. Se obtiene la lista correspondiente al país especificado mediante la tabla de hash implementada en la carga de datos. Complejidad O (1)
2. La lista obtenida en el punto anterior es usada dentro de un bucle, el cual compara si cada elemento de la lista tiene la categoría pasada por parámetro. Si el elemento no tiene esa categoría, se deja al final de la lista, de tal manera que al inicio de la lista quedan los elementos con la categoría especificada. Dicho esto, la complejidad es O(n) debido a que se hace un solo bucle que recorre toda la lista
3. Con la lista ordenada, se reducen los datos de ella, de tal manera que la lista quede con solo los videos que tengan la categoría especificada. Complejidad O (1)
4. La complejidad del algoritmo es: O(1) +O(n)+O(1)

Requerimiento2(Estudiante1)

1. Se obtiene la lista correspondiente al país especificado mediante la tabla de hash implementada en la carga de datos. Complejidad O(1)
2. La lista obtenida anteriormente se ordena por el id mediante un merge sort, de tal manera que los videos que tengan el mismo id queden consecutivos. Complejidad O(n log n)
3. La lista ordenada es recorrida mediante un bucle en donde se compara cuantos días de tendencia tiene cada video. Complejidad O(n)
4. La complejidad del algoritmo es : O(1) + O(n log n) + o(n)

Requerimiento 4 (Grupal)

1. Se obtiene la lista correspondiente al país especificado mediante la tabla de hash implementada en la carga de datos. Complejidad O(1)
2. Se recorren los datos de la lista anterior mediante un bucle, el cual compara si cada video tiene un tag especifico. En caso de que el video tenga el tag, es añadido a una nueva lista. Complejidad O(n)
3. La nueva lista es ordenada por Likes mediante un merge Sort. Complejidad O(m log m)
4. La lista se ordena ahora por id mediante un merge sort, de tal manera que los videos con mismo id queden consecutivos, y que a su vez el primero de esos mismos videos sea el que tenga más likes. Complejidad O (m log m)
5. Los videos de la lista se recorren mediante un bucle donde se añade a la lista cada video, el cual es único en ella, es decir, si hay varios videos con el mismo id, solo se añade uno a la nueva lista. Complejidad O( m)
6. La nueva lista es ordenada por likes mediante un merge Sort . Complejidad O(x log x)
7. Complejidad : O(1) + O(n) + O(m log m) + O(m log m)+ O(m) + O(x log x)

Donde : X<= M <= N