

Maria Camila Luna Velasco – 201920993

Santiago Arciniegas Castro – 202014994

## **USAR ARRAYLIST PARA PROBAR LOS REQUERIMIENTOS.**

### **Requerimiento 1.**

En el caso del requerimiento 1 aunque se utilizan ciclos, ninguno está anidado y el resto son constantes. Por lo que el más costoso es el algoritmo de ordenamiento Merge Sort el cual en promedio tiene una complejidad temporal de promedio de  $O(n \log(n))$ ; por lo que este debería ser el orden de complejidad temporal. Nota Extra: En este requerimiento decidimos calcular los vídeos en tendencia sobre el último registro de likes por cada vídeo, para así poder mostrar aquellos que tuvieron más likes sin repetir vídeos.

### **Requerimiento 2 (Maria Camila Luna Velasco)**

En el caso del requerimiento 2 aunque se utilizan ciclos, ninguno está anidado y el resto son constantes. Por lo que el más costoso es el algoritmo de ordenamiento Merge Sort el cual en promedio tiene una complejidad temporal de promedio de  $O(n \log(n))$ ; por lo que este debería ser el orden de complejidad temporal.

### **Requerimiento 3 (Santiago Arciniegas Castro)**

En el caso del requerimiento 3 aunque se utilizan ciclos, ninguno está anidado y el resto son constantes. Por lo que el más costoso es el algoritmo de ordenamiento Merge Sort el cual en promedio tiene una complejidad temporal de promedio de  $O(n \log(n))$ ; por lo que este debería ser el orden de complejidad temporal.

### **Requerimiento 4**

En el caso del requerimiento 4 aunque se utilizan ciclos, ninguno está anidado y el resto son constantes. Por lo que el más costoso es el algoritmo de ordenamiento Merge Sort el cual en promedio tiene una complejidad temporal de promedio de  $O(n \log(n))$ ; por lo que este debería ser el orden de complejidad temporal.