

Documento de análisis

Reto 1-EDA-2021-10-Grupo 12

Estudiante A: Julian Castro del Valle - j.castrod@uniandes.edu.co - 202020847

Estudiante B: Tomás Otero - t.otero@uniandes.edu.co – 202021733

Análisis de complejidad por requerimiento:

- Requerimiento 1: $O(N)$ ya que cuenta con un ciclo for para recorrer la lista.
- Requerimiento 2 (Estudiante A): $O(N^2)$ ya que cuenta con dos ciclos for, uno para recorrer la lista principal y otro para recorrer la sub-lista que finalmente sacará el video que fue trending más días (en países).
- Requerimiento 3 (Estudiante B): $O(N^2)$ ya que cuenta con dos ciclos for, uno para recorrer la lista principal y otro para recorrer la sub-lista que sacará el video que fue trending más días (en categorías).
- Requerimiento 4: $O(N)$ ya que cuenta con un ciclo for para recorrer la lista.

Para los requerimientos 1 y 4 se utiliza el algoritmo merge sort que tiene complejidad $O(N \log N)$ en promedio, siendo el algoritmo más rápido de los vistos en clase.

Se utiliza la estructura de datos tipo arreglo (ARRAY_LIST), al ser la más óptima en este caso en cuanto a velocidad y eficacia para almacenar datos.

Observaciones importantes:

- La función `videos_a_dias_trending` cuenta con complejidad $O(N)$ pero usa merge sort, que tiene, en promedio, complejidad $O(N \log N)$. Se usa para contar los días que fue trending un video, o sea que se usa para el requerimiento 2 y 3.
- Las funciones `printResults($O(N)$)`, `printResultsLikes($O(N)$)`, `printResultPais ($O(1)$)` y `printResultCateg ($O(1)$)` son auxiliares para que se muestre la información requerida en el view, no influyen en cómo se organizan/filtran los datos.