

Laboratorio N°2

Estructura de Datos y Algoritmos

Fecha de inicio 17 de Abril 2023 hasta las 23:59h

Fecha de entrega 23 de Abril 2023 hasta las 23:59h

Enunciado

Como ingenieros/as en informática y telecomunicaciones les llega un trabajo desde una plataforma de vídeos, en el cual le entregan una API con las clases y métodos actualmente implementados en la empresa. A su vez, se tiene una CLI (Command-Line Interface) llamada Client.py que consume la API según los datos ingresados por el usuario.

Dataset: <https://www.kaggle.com/datasets/danielaamoreno/youtubevideosdts>

Codigos: <https://drive.google.com/drive/folders/1z96wcYGaWPIk1NeNiPvnYxtLAJjWlkvF?usp=sharing>

Descripción de la API (Aplication Plataform Interface)

Clase Video

Atributos

Clase *Video* que contiene los siguientes atributos:

- **videoID:** ID del vídeo (campo único).
- **videoTitle:** título del vídeo.
- **channelID:** ID del canal que subió el vídeo.
- **channelTitle:** título del canal.
- **publishedAt:** fecha de publicación del vídeo.
- **viewCount:** cantidad de visitas.
- **likeCount:** cantidad de me gusta.
- **commentCount:** cantidad de comentarios.
- **next:** siguiente vídeo a reproducir.

Métodos

- **play:** imprime en consola el ID y el título del vídeo.

Clase Platform

Atributos

La clase *platform* es una **lista enlazada simple** con el siguiente atributo:

- **head:** inicio de la lista enlazada.

Métodos

- **isNumericInt:** recibe un *String* y verifica si es posible convertirlo en *int*. Si es posible lo convierte, en caso contrario retorna 0.
- **isNumericLong:** recibe un *String* y verifica si es posible convertirlo en *long*. Si es posible lo convierte, en caso contrario retorna 0.
- **arrayToVideo:** crea y retorna un vídeo con los datos de un *arrayList*.
- **InsertFromFile:** inserta todos los datos de un archivo .csv en la lista enlazada.
- **iterativeLast:** recorre la lista de forma iterativa hasta el último nodo y lo retorna.
- **recursiveLast:** recorre la lista de forma recursiva hasta el último nodo y lo retorna.
- **insertAtEnd:** inserta un vídeo al final de la lista enlazada.
- **iterativePrint:** imprime todos los vídeos de la plataforma accediendo a ellos de forma iterativa.
- **recursivePrint:** imprime todos los vídeos de la plataforma accediendo a ellos de forma recursiva.

Actividad 1

Como primer trabajo, le solicitan que incluya el parámetro de popularidad en cada vídeo recopilado desde el archivo .csv. La popularidad se calcula como la cantidad de vistas dividida en la cantidad de *likes*. En caso de que la cantidad de *likes* sea 0, entonces la popularidad será de 0.

Como puede darse cuenta, no existe el atributo popularidad en la clase *Video*, por lo que tendrá que calcularla e incluirla a la clase.

Por otro lado, mientras analiza la API se da cuenta de que hay errores e implementaciones poco óptimas en los métodos actuales. Realice **depuración** y **mejoras** a las clases y métodos que le entregaron.

Específicamente en esta actividad debe:

1. **Depurar las clases y métodos** para que funcionen correctamente.
2. **Calcular e incluir** la popularidad de cada vídeo de la plataforma.
3. En caso de ser posible **reducir la complejidad temporal** de algún método, mejórelo.
4. **Incluir el método** para buscar un vídeo por el campo *videoID*.
5. **Incluir el método** para invertir la lista de orden, es decir, el *head* ahora será el último nodo y debe invertir cada uno de los punteros hacia el nodo anterior.
6. Probar el funcionamiento de la CLI (command-line interface) ubicado en el archivo Client.java.

Actividad 2

Realice un informe que contemple los siguientes aspectos:

- Comente acerca del proceso de depuración realizado.
- De acuerdo con los métodos mejorados e incluidos de las actividades anteriores. ¿Cuál es la complejidad temporal en su peor caso? En caso de mejora de un método ¿En cuánto mejoró la complejidad temporal?
- La API contaba con métodos con implementaciones iterativas y recursivas. ¿Cuáles son las principales diferencias entre estas?

Condiciones de entrega

- La tarea se desarrolla en **parejas**. Los códigos serán sometidos a comprobación automática de plagio. En caso de copia entre estudiantes se sancionará con la nota mínima.
- El código debe estar escrito en Java y se debe poder compilar y ejecutar **sin cambios**. De no cumplir esto será evaluado de forma inmediata con la nota mínima.
- El informe debe ser entregado en formato **PDF**. Realizar el informe es requerimiento, si el informe está vacío será calificada con la nota mínima.
- **Se deben entregar todos los programas y el informe en un archivo .zip** con el nombre del grupo vía canvas. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato.