

Laboratorio N° 1
Estructuras de Datos y Algoritmos

Fecha de inicio: Jueves 22 de Marzo 2022

Fecha de entrega: Jueves 21 Abril 2022 hasta las 23:59h

Enunciado

En este laboratorio procesaremos un dataset con más de 580 mil registros de canciones de Spotify y se utilizarán las estructuras de datos y algoritmos vistos en clase y cursos anteriores. Se debe hacer uso de ArrayLists, Queue y Stack, pero con la condición de que deben implementar ustedes mismos las últimas dos.

Dataset:

<https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks?select=tracks.csv>

1. **(1 punto)**. Escriba un código java que lea el archivo tracks.csv y la copie a una estructura de datos que permita interactuar con estos datos en la memoria del aplicativo.
2. **(1.5 puntos)** Cuente el número de Tracks de cada artista, calcule la popularidad promedio de sus canciones y el promedio de su bailabilidad. Guarde estos resultados junto a la canción más popular de cada Artista.
3. **(2 puntos)** Ordene la información procesada en el punto anterior desde el artista **más popular al menos popular**. Si existen artistas que tengan la misma popularidad promedio estos a su vez se deben ordenar desde el **más al menos bailable**. Una vez ordenados, agregue el resultado a una única playlist "Canciones más populares de cada artista".

La playlist debe tener las siguientes operaciones/restricciones:

- a. La playlist siempre será reproducida empezando con la canción **más popular a la menos popular**.
 - b. Un método llamado *next()* que permita obtener la siguiente canción en la playlist.
 - c. Un método *skipN(int n)* que permita saltar desde la canción actual a la enésima siguiente canción en la lista. Si N es mayor a la cantidad de canciones restantes en la lista de reproducción, entonces la reproducción termina.
4. **(1.5 puntos)** Utilizando el playlist generado en el punto anterior, cree una interfaz por línea de comandos (CLI) que permita al usuario reproducir las canciones utilizando los métodos ya implementados (*next* o *skipN*). El usuario debe tener la

posibilidad de dar *like* a la canción que se reproduce actualmente, almacenando en un historial el ID del track al que se le dió *like*. Además, el CLI debe ser capaz de imprimir dicho historial en orden descendente, empezando por la última canción a la cual el usuario le dió like.

Condiciones de entrega:

- El laboratorio se debe desarrollar en **parejas**.
- El código debe estar escrito en java y debe poder compilarse y ejecutarse **sin cambios**. De no cumplirse esto será evaluado de forma inmediata con la nota mínima
- El laboratorio debe entregarse en un archivo .zip con el nombre del grupo via canvas.
- En caso de copia entre estudiantes se sancionará con la nota mínima.
- Los retrasos en las entregas se sancionarán con la nota mínima.