LABORATORIO 1: Estructuras de datos y algoritmos
FIC Music
Profesores Laboratorio: Benjamín Alonso - Nicolás Núñez

Estructuras básicas. Pilas, Colas y Listas Enlazadas

Lea el documento completo antes de empezar el desarrollo del laboratorio

Objetivos

- Comprender y aplicar listas enlazas (LinkedList), pilas (Stack) y colas (Queue).
- Implementar operaciones básicas de las estructuras mencionadas.
- Realizar análisis de complejidad y entender su importancia.

Enunciado

La Facultad de Ingeniería y Ciencias (FIC) es muy aburrida y ha bloqueado el acceso a muchas de las plataformas de reproducción de música: YuTub Music, Spotiflain, Pear Music, etc. Por ende, usted y un grupo de amigos, han decidido crear una plataforma que realmente sea capaz de dar vida a la facultad. La idea se llama *FIC Music*.

La plataforma *FIC Music* ha progresado en su desarrollo. No obstante, aún falta el desarrollo de la parte lógica de la aplicación. Dado tu actual compromiso con el curso de "Estructuras de datos y algoritmos", consideras apropiado encargarte de esta sección. Tu intervención permitirá integrar estructuras de datos clave como *Listas enlazadas*, *Colas* y *Pilas*, esenciales para la operatividad óptima de la aplicación.

Se le entregará un código por el cual los estudiantes tenían pensando en controlar la plataforma. Usted debe, creando desde cero las estructuras de datos que conoce.

Este código puede leer todas canciones ingresadas desde un dataset (el archivo .csv entregado) y solo imprimir por terminal los nombres de las canciones, por lo que usted debe modificar este archivo para así completar todos los requerimientos existentes. En el caso que estos requerimientos necesiten algún tipo de input estos se deben realizar sólo por terminal.

Requerimientos

Clases

1. Implementar una clase `Song` que represente una canción con atributos como título, artista y duración.
2. Implementar una LinkedList (Lista Enlazada) para almacenar todas las canciones.
3. Implementar una Queue para crear una cola de reproducción.
4. Implementar un Stack para representar el historial de canciones reproducidas.

Controladores

1. Crear un controlador para gestionar canciones (agregar, eliminar, buscar).
2. Crear un controlador para el reproductor que permita añadir canciones a la lista (cola) de reproducción, reproducir la siguiente canción y ver la próxima canción.
3. Crear un controlador para cambiar el orden entre dos canciones en la lista de reproducción sólo utilizando colas.
4. Crear un controlador para eliminar canciones de la lista o cola de reproducción utilizando una o dos colas.

5. Crear un controlador para el historial que permita ver la última canción reproducida y añadir canciones al historial.
6. Implementar una funcionalidad búsqueda de canciones en base al título de esta.

Análisis de Big O de estructuras

Pruebas Empíricas

Aunque el análisis teórico de algoritmos es esencial para comprender su comportamiento en general, las pruebas empíricas nos permiten observar el rendimiento real de nuestras implementaciones en situaciones específicas. Para este laboratorio, se espera que realicen pruebas empíricas en sus estructuras de datos:

Medición de tiempo

Debe medir el tiempo de su programa al realizar las siguientes acciones:

1. Agregar todas las canciones existentes.
2. Agregar todas las canciones a la cola de reproducción.
3. Buscar una canción en específico después de agregar todas las canciones a la cola de reproducción.
4. Mover la última canción de la cola al primer lugar de esta.
5. Mover la primera canción de la cola al último lugar de esta.
6. Agregar todas las canciones en el historial
7. Buscar canciones.
8. Buscar canciones en el historial.

Reflexión

Hable sobre los resultados obtenidos ¿Estos fueron aceptables? ¿Hay alguna operación que parezca inusualmente lenta? Si es así ¿Como se puede mejorar? ¿Existe alguna manera más óptima de realizar las operaciones?.

Si pudiera agregar muchas playslists ¿Que estrucura de datos utilizaría? ¿Que complejidad tendría buscar cuántas veces se repite una canción entre todas esas playlists?

Evaluación

- Correcta implementación de estructuras de datos: 40%
- Funcionalidad de controladores: 30%
- Interfaz de usuario y experiencia general: 15%
- Informe: 15%

Recursos

- Documentación oficial de Java sobre estructuras de datos
- Calculate execution time
- hrefhttps://drive.google.com/drive/folders/14TOigA097ORX-TpidYhK2PA$_{O}$***knCAzxe?usp = sharingDrivedelcódigoydat***

Aspectos formales de entrega

- **Fecha de entrega: 24** de Septiembre hasta las 23:59 hrs.
- **Número de integrantes:** Individual.
- Las copias de código serán penalizadas con nota mínima. Referencie apropiadamente todo segmento de código que no sea de su autoría.
- Consultas:
  nicolas.nunez2@mail.udp.cl o **naike1** en discord.
  benjamin.alonso1@mail.udp.cl o **papinha1** en discord.