

Tune Trace

Juan David Ardila Diaz

Juan Esteban Cardenas Huertas

Juan Esteban Peña Burgos

Julian Andres Gomez Niño

Nestor Steven Piraquive Garzon

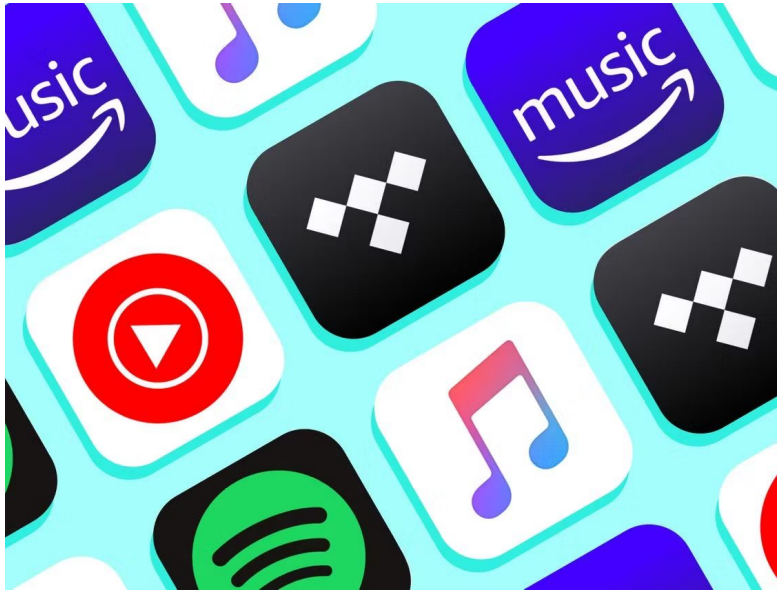


UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE BOGOTÁ



Problema a resolver

El dilema que enfrentan los amantes de la música es la limitación de las plataformas de streaming actuales para ofrecer recomendaciones que a veces no tienen coherencia con lo solicitado.



Esto resulta en una experiencia de búsqueda y descubrimiento limitada.

Por lo tanto la solución es TuneTrace.





Requerimientos funcionales

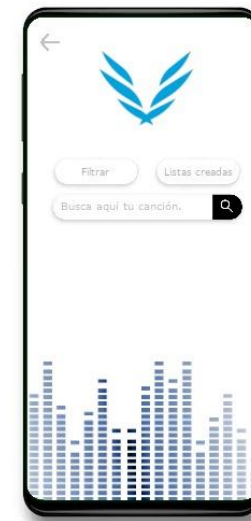
- **Ingreso de Canciones por Usuario:**

La aplicación almacena la lista de canciones introducidas por el usuario, junto con sus características específicas, como la duración, el género, el autor, entre otros.



- **Filtración de canciones personalizado:**

La aplicación almacena la lista de preferencias seleccionadas por el usuario (puede estar vacía).





Requerimientos funcionales

- **Recomendaciones de canciones:**

búsqueda parcial en la biblioteca de canciones de canciones que compartan dos o más características con la lista de canciones ingresadas por el usuario, además de cumplir con las preferencias seleccionadas.



- **Guardado de lista de canciones escogidas por el usuario:**

La aplicación almacena la lista de canciones generada durante la búsqueda parcial y la muestra en la sección correspondiente.





Uso de estructuras de datos en la solución del problema a resolver

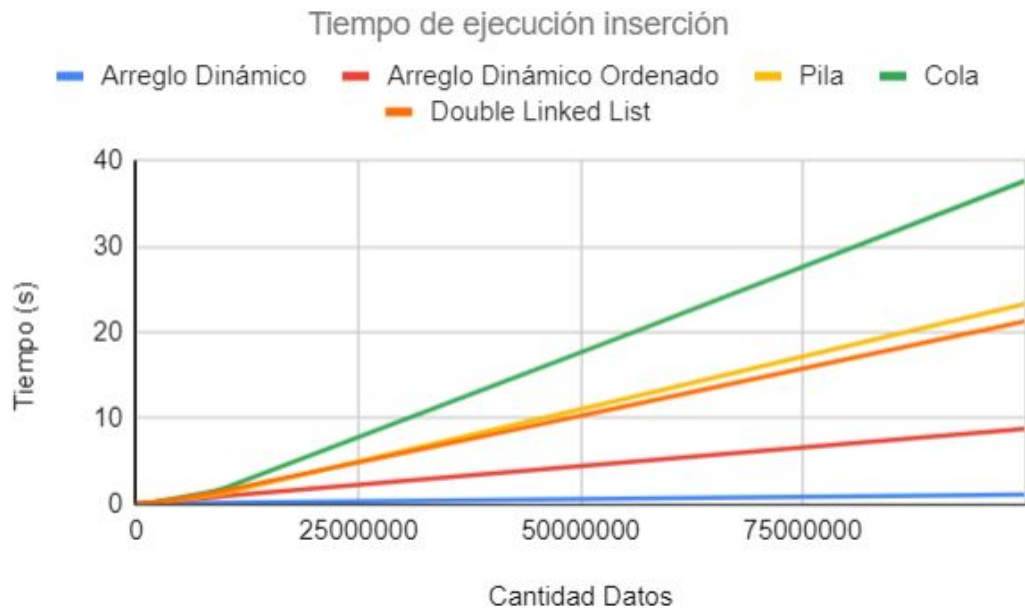
Arreglos dinámicos: Se implementan arreglos dinámicos tanto ordenados como no ordenados, para el almacenamiento de las canciones existentes como la base con la que el usuario interactúa. (Guarda los objetos canciones, y permite realizar filtros de acuerdo a los atributos)

Pilas y Colas: Se implementan de la clase LinkedList, para permitir al usuario almacenar las canciones de interés que le ingresa al programa, desde las cuales este toma las decisiones para realizar recomendaciones.

Lista doblemente enlazada: Se implementa para el retorno de recomendaciones, de manera que el usuario pueda realizar modificaciones y guardar las canciones que le interesan, además de aplicar filtros de acuerdo a sus intereses personales.

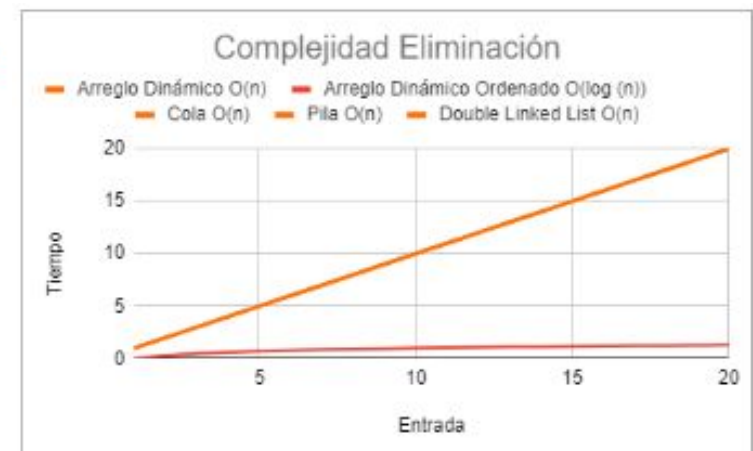
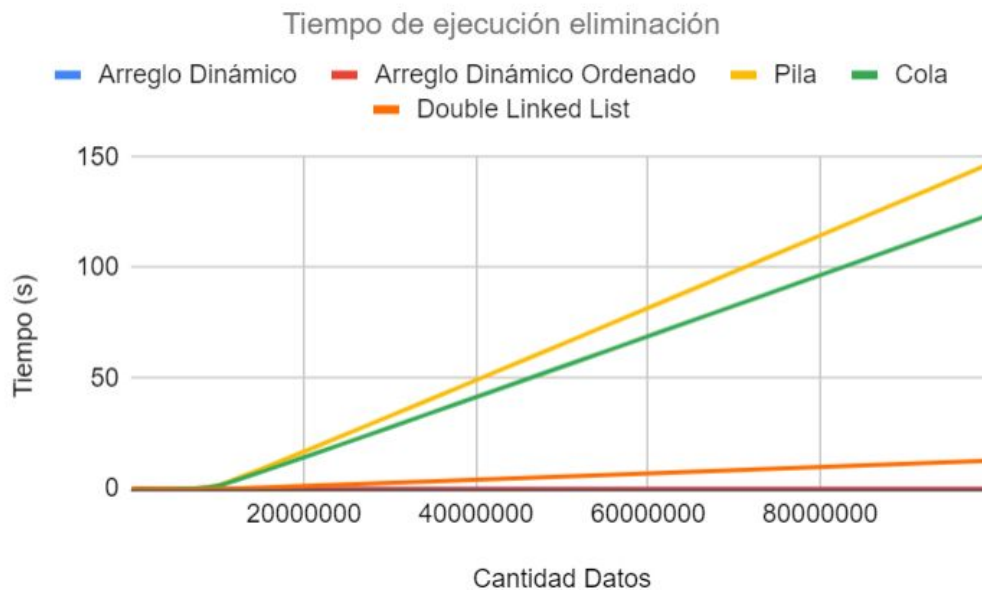


Tiempos de ejecución (Inserción):





Tiempos de ejecución (Búsqueda):





Tiempos de ejecución (Eliminación):





Tiempos de ejecución (Consulta):





Que se observó?



- Para el funcionamiento del proyecto se ve necesario la implementación de estructuras dinámicas que puedan escalar al nivel que lo requiere el aplicativo.
- De las estructuras analizadas las colas y pilas tienen los peores tiempos tanto de inserción como de búsqueda y eliminación de un elemento.
- Las estructuras que mejor cumplen estos requerimientos son las Double Linked-List y los arreglos Dinámicos.

[...]

