

Tune Trace

Juan David Ardila Diaz
Juan Esteban Cardenas Huertas
Juan Esteban Peña Burgos
Julian Andres Gomez Niño
Nestor Steven Piraquive Garzón

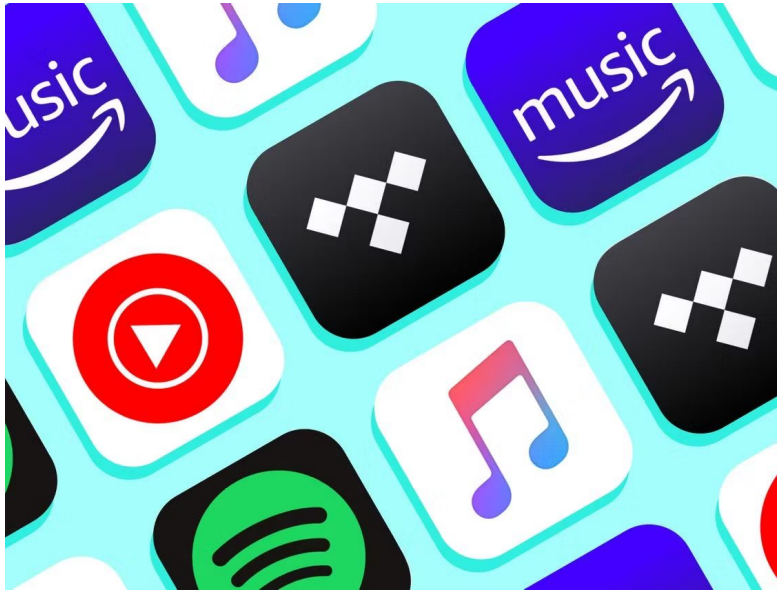


UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE BOGOTÁ



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.





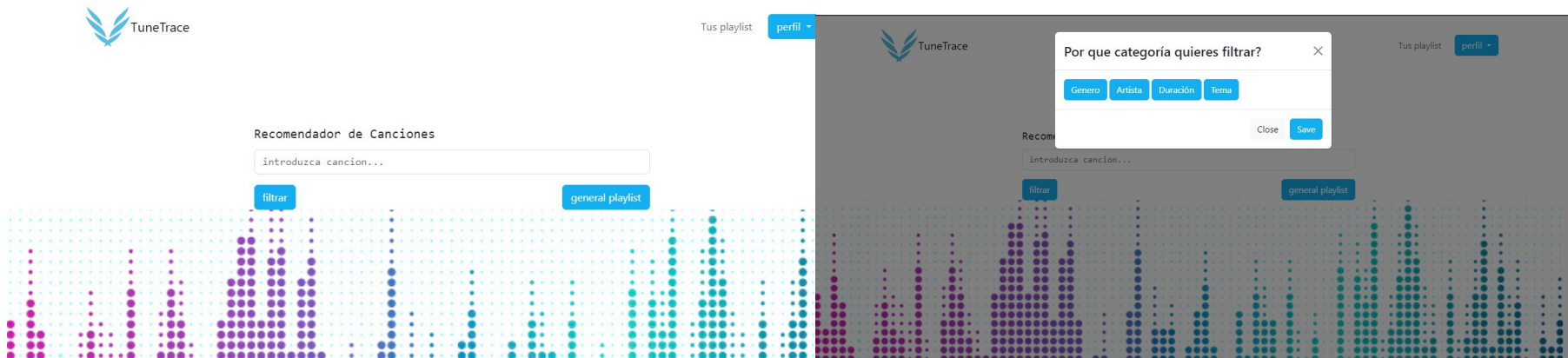
Requerimientos funcionales

- **Ingreso de Canciones por Usuario:**

La aplicación en base a la(s) canción(es) ingresada(s) por el usuario genera una playlist en base a los metadatos de esta y demás filtros que el usuario ingrese.

- **Filtración de canciones personalizado:**

La aplicación almacena los criterios de filtrado que el usuario quiera seleccionar para generar una playlist.





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

Árbol binario de búsqueda (BST) y AVL: Se implementaron los árboles binarios con el fin de clasificar las canciones por un parámetro dado (cada árbol representa un parámetro) y mantenerlas ordenadas, ya sea por identificador o nombre de la canción.

Lista doblemente enlazada: Se implementa para el guardado de los árboles que corresponden a cada posibilidad del parámetro dado.

Heaps: Se implementa para el guardado de los árboles que corresponden a cada posibilidad del parámetro dado. En este caso, el objetivo es buscar canciones de menor duración o parámetros del estilo.

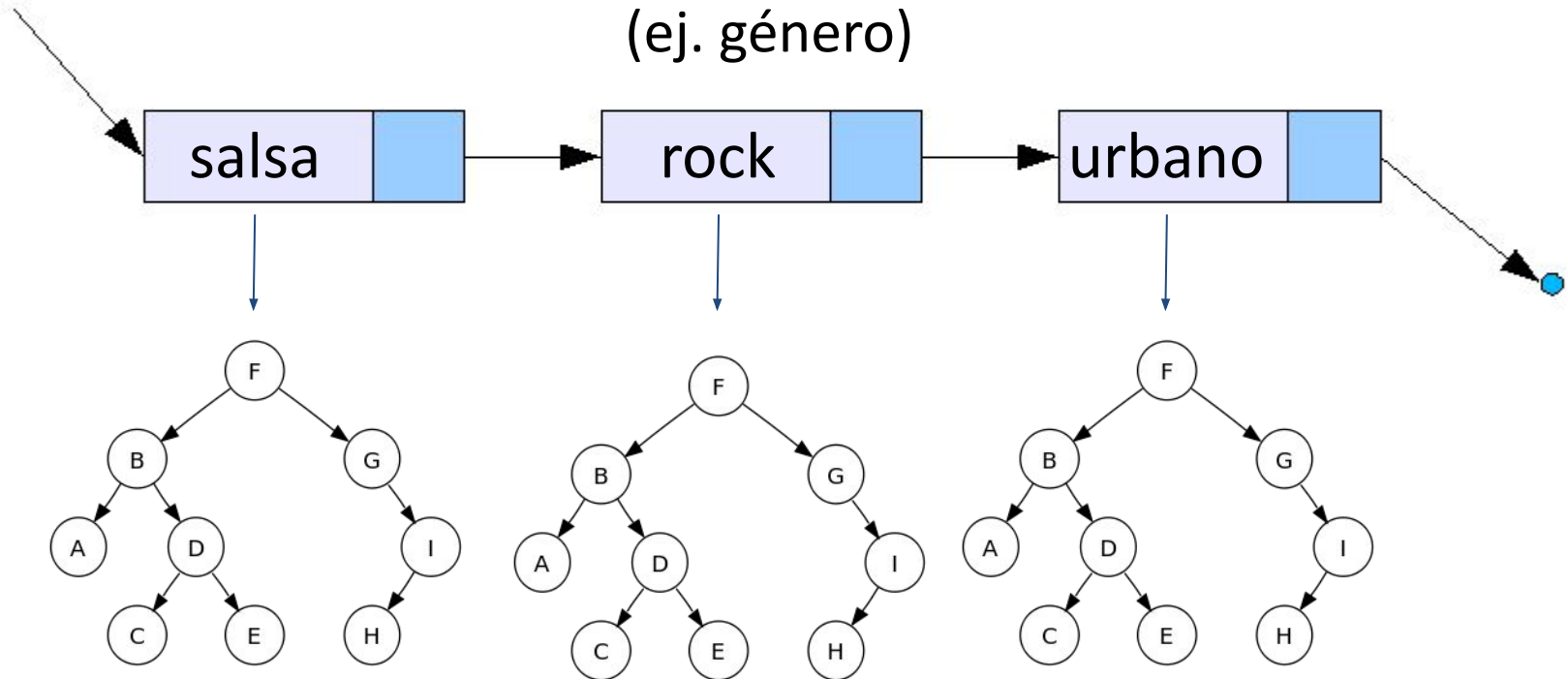
Conjuntos disjuntos: Se implementaron con el fin de mantener las canciones agrupadas y tener sus relaciones según un parámetro dado.

Arreglos dinámicos: Se implementaron de manera complementaria a los heaps y a los conjuntos disjuntos.



Idea general para la clasificación:

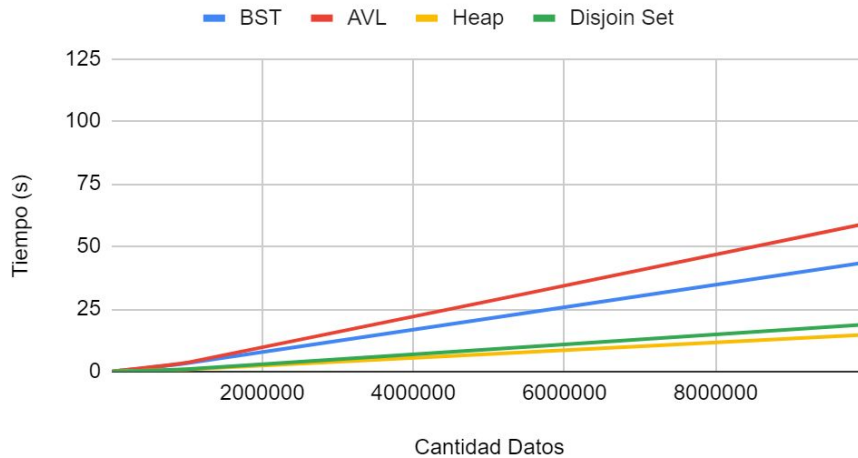
LISTA DEL PARÁMETRO DADO (ej. género)



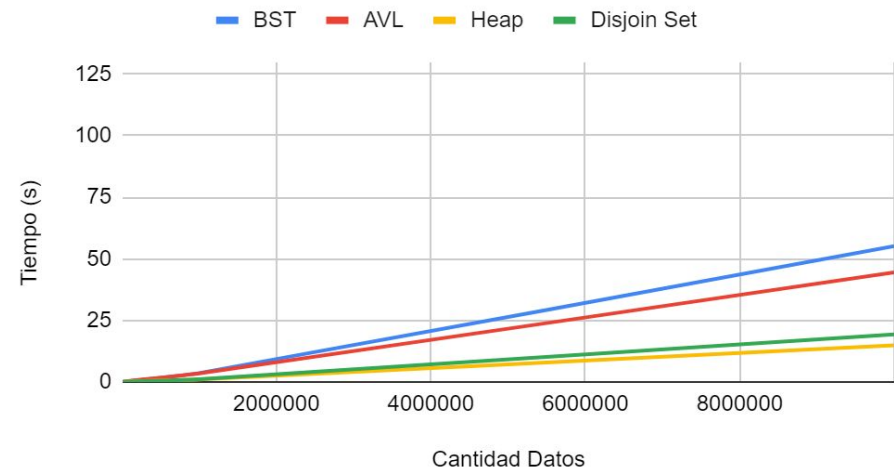


Tiempos de ejecución (Inserción y clasificación):

Inserción y Clasificación (Género)



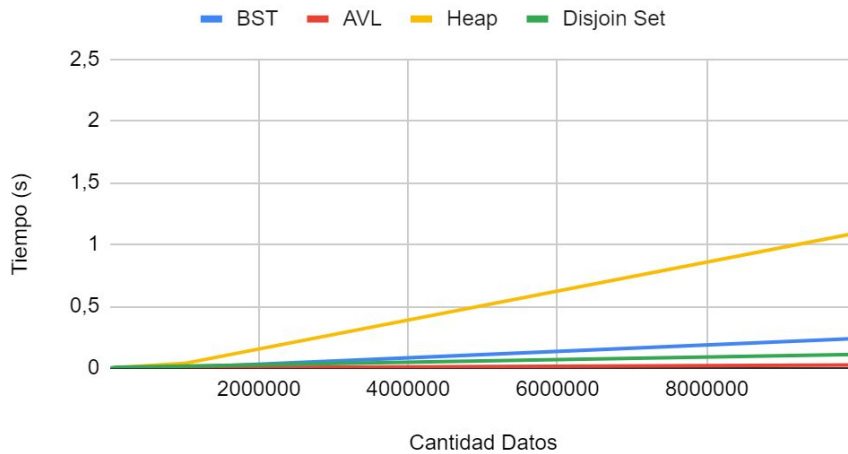
Inserción y Clasificación (Autor)



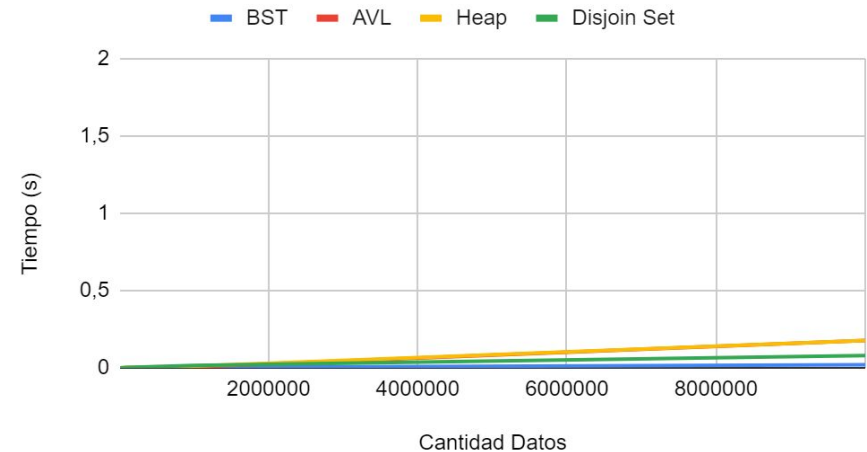


Tiempos de ejecución (Consulta por parámetro dado):

Consulta por Género



Consulta por Autor





Complejidades:





Que se observó?

- Pudimos evidenciar que, aunque no nos permita almacenar los datos de manera ordenada, los conjuntos disjuntos resultan ser bastante eficiente para agrupar y filtrar canciones por parámetros. Al igual que consultarlos por filtros.
- Los árboles BST y AVL tienen un comportamiento similar, sin embargo, el BST podría generar inconvenientes en caso de una mala inserción de datos. Por lo que es mejor hacer uso de AVL para este proyecto ya que no tenemos control de la inserción de canciones.
- Según las necesidades de ordenamiento del algoritmo final, podría usarse el método de filtro que usa AVL o los conjuntos disjuntos.

[...]