

Desafío II – UdeATunes Simulación de Plataforma de Streaming Musical

Cristian Florez Monsalve

Thomas Mejia Moncada

Grupo 6

informática II

Aníbal José Guerra Soler

Universidad de Antioquia

Medellín - Colombia

17/10/2025

Análisis del Problema

El desafío consiste en el diseño de una aplicación de consola que emule las funcionalidades básicas de una plataforma de streaming musical digital, va enfocado a la construcción de estructuras de datos que permitan modelar una interacción entre usuarios, artistas, canciones, álbumes, entre otros componentes triviales que hacen parte de la interacción como los anuncios. Cada clase que hace parte del problema cumple un papel esencial según los atributos que se le sean otorgados, para los usuarios se les otorgan ciertas ventajas a demás de las funcionalidades estándar que tienen predefinidas dependiendo del tipo de membresía que se posee, creemos que la dificultad del problema radica en 3 cosas esenciales.

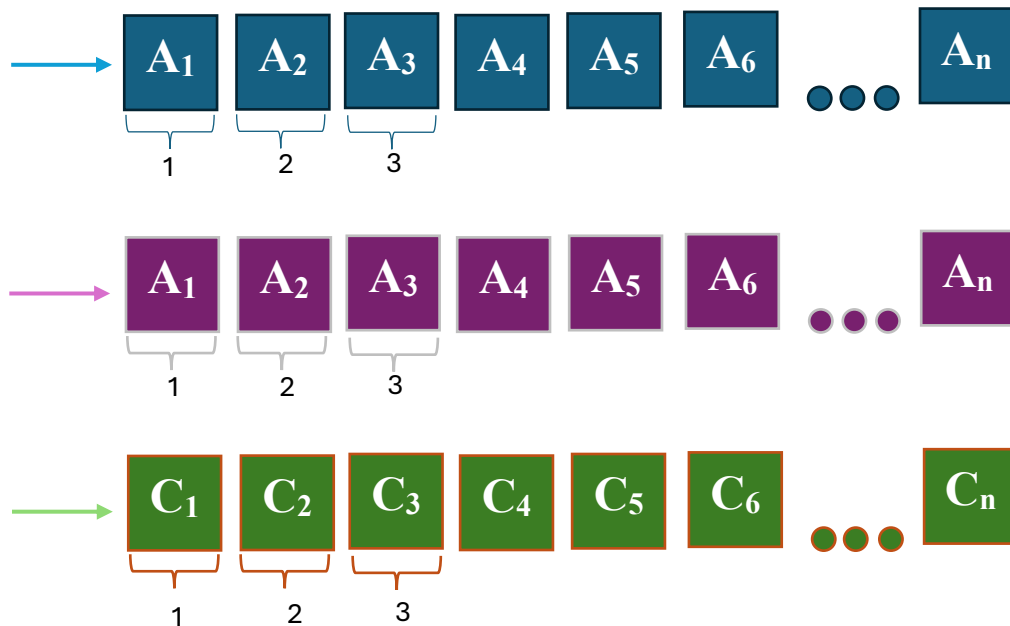
1. El modelado de la estructura de datos ya sea permanente o en tiempo de ejecución ambas estructuras deben guardar una relación entre las interacciones de los objetos en el programa por ende creemos que hay un desafío a la hora de elegir una estructura idónea que no clone datos de forma innecesaria.
2. El diseño de la interacción entre los objetos, si bien parece simple llegamos a la conclusión de que la forma en la cual se relacionan los objetos altera el funcionamiento por ende la forma en la cual sus métodos se comunicaran entre clases y como se comportaran las sobrecargas de operadores añade un nivel más de dificultad.
3. La medición de los recursos, aunque es ajena al problema principal creemos que medir los recursos que nos dan una métrica de la eficiencia del programa también es algo laborioso y complejo pues calcular el espacio disponible en memoria y la cantidad de iteraciones hechas por métodos dentro de clase nos añade un requisito mas a la hora de como definir las y declararlas.

En resumen, pensamos que el desafío puede direccionarse a una problemática general y es evitar el sobredimensionamiento, aplicación innecesaria de los datos y como estos se comporten. Tenemos en cuenta y nos prevenimos de caer en la tentación de asociar el desafío con una plataforma ya existente como Spotify, pues como conscientes de que asociarla con un proyecto verdadero de streaming musical podría llevarnos a un mal desarrollo del desafío con las limitaciones de funcionalidades que ya tiene predefinidas como el hecho de que no hay funcionalidad para registrar usuarios y que la mayoría de los datos ya estarán predefinidos en el data set hecho por nosotros.

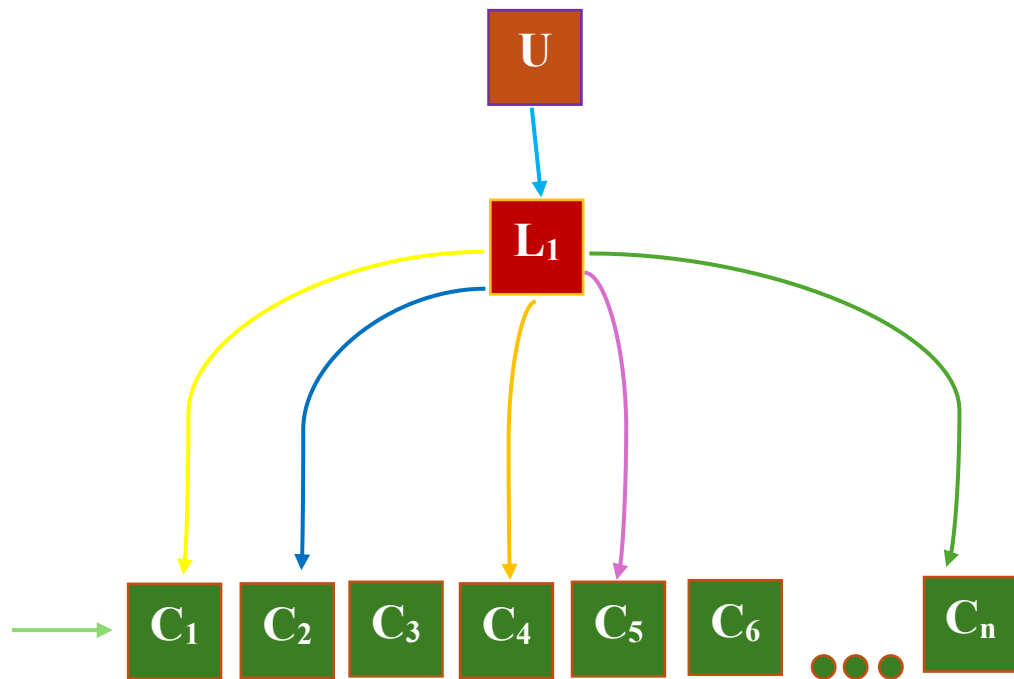
Solución Planteada

Para desarrollar la problemática proponemos un modelo de objetos en conjunto con algunas librerías accesorias que ayudaran a hacer tareas simples o triviales, de tal modo que podamos emular de correcta forma la realidad planteada en el enunciado, la clave de nuestra solución se basa en la correcta utilización y manipulación de las relaciones que existen entre los datos de modo tal que mantener los datos no sea costoso y procesarlos tampoco sea problema.

- La estructura de datos planteada se basa en apuntadores a arreglos de objetos, ya sea de Usuarios, Anuncios, Artistas, Álbumes, etc... y los relacionamos entre si a través de los códigos que estos tengan para el caso de (Artistas, Álbumes, Canciones) y el nickname para el caso de Usuarios.



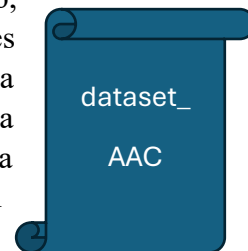
Aunque esta estructura de apuntadores a arreglos que contienen objetos de cada clase sirve para manejar de forma sencilla grupos de objetos, necesitamos que se relacionen de alguna forma, en el caso de Artistas, Álbumes, y Canciones se relacionan únicamente por el código identificador que los une, pero en el caso de las canciones que pertenecen a las listas de los usuarios modelamos algo mas complejo pero mas eficiente a la hora de hacer búsquedas y cuidar la memoria.



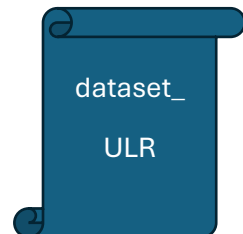
Básicamente las canciones favoritas de cada usuario están relacionadas con el gracias a una clase llamada ListaRP (Lista de reproducción), que nos ayuda a hacer un uso eficiente de la memoria al evitar la clonación de canciones de forma innecesaria para cada usuario por medio de apuntadores que señalan al arreglo de canciones global, cada lista de reproducción únicamente tiene hasta 10.000 apuntadores ósea la cantidad máxima de canciones favoritas que se puede tener.

Para la estructura de la data setproponemos utilizar 3 archivos de texto plano pues ocupan poco espacio para una gran masa de datos al guardarlos en forma de solo texto, y aparte nos permite guardar las relaciones que deben existir entre los datos para que a la hora de leerlo se pueda saber a quién pertenece cada cosa.

El primer archivo se usará para almacenar Artistas, Álbumes, Canciones y lo denotamos como dataset_AAC por la inicial de cada tipo de dato, y se relacionan entre sí por los códigos. Y se divide por bloques de forma tal que hay un bloque para cada clase con una etiqueta que indica cual de todas pertenece dicho bloque en la primera línea o en alguna línea posterior, en cada bloque cada línea representa la información ya sea de un artista, una canción, o un álbum delimitado por el símbolo de |



El segundo archivo se usará para almacenar toda la información referente a los usuarios, las listas de reproducción que se forman con sus canciones favoritas. Esta



dataset_
ULR

dividido por bloques que se separan por un símbolo (-) la primera línea consta de información del usuario y las líneas siguientes de códigos a canciones que pertenecen a su lista de reproducción de modo tal que;

| Nickname | Rol | Ciudad/País | Fecha | registro |
|-------------------|-------------------|-------------------|-------------------|-------------------|
| code ₁ | code ₂ | code ₃ | code ₄ | code ₅ |
| code _n | | | | |

Y denotamos el archivo como dataset_ULR

El tercer archivo se usa para guardar la información de los anuncios donde simplemente cada línea representa la información correspondiente a cada anuncio dividido en 2 partes prioridad | mensaje, así de sencillo. El archivo se denota por dataset_An.

Con estos 3 componentes pretendemos darle una satisfactoria solución a la problemática de como relacionar de forma sencilla todos los datos sin necesidad de crear copias innecesarias de cada uno de ellos.



Dificultades de Desarrollo

En cuanto a problemáticas con el desarrollo tuvimos muchas lo cual nos impidió lastimosamente por esta vez no alcanzar una solución oportuna al problema, pues a pesar de la planeación tuvimos algunos inconvenientes entre los cuales destacan:

1. **Errores con los módulos:** Tuvimos inconvenientes a la hora de intentar implementar funciones de tareas necesarias que aunque no encajaban en los métodos de las clases eran necesarias pues al invocarse desde múltiples clases para su uso descubrimos que generaba un error de multicopia de funciones.
2. **Problema a la hora de diseñar la base de datos:** Sin duda el mayor de los problemas en este desafío fue crear una estructura de datos que no clonara datos de forma innecesaria, y aunque al final lo logramos nos habíamos perdido ya mucho tiempo de desarrollo.
3. **Factores externos:** Sin duda desarrollar una solución para el desafío es una tarea que demanda de mucho tiempo curricular del cual tristemente muchas veces no poseíamos por factores externos.
4. **Problemas de merge:** A la hora de trabajar con ramas tuvimos ciertas confusiones para arreglar problemas con las diferentes versiones entre ramas que eran un dolor de cabeza por resolver, pues requería análisis sobre lo que se necesitaba y lo que no teniendo la delicadeza de no borrar nada importante.
5. **Problemas no previstos:** Aunque somos conscientes de que a la hora de programar requerimos antes un buen análisis y planeación del proyecto también nos dimos cuenta en el desafío que no siempre es suficiente una buena planeación pues en el camino surgen problemáticas ocultas que nunca se consideraron, pero son necesarias para hacer que el programa funcione lo cual añadía capas de dificultad extra.

Conclusión

Aunque en esta ocasión no logramos cumplir completamente con el objetivo propuesto, nos sentimos orgullosos del proyecto realizado y de nosotros mismos. El tiempo y la dedicación invertidos reflejan el compromiso y el cariño que tuvimos por él, a pesar de las dificultades encontradas durante su desarrollo. No solo reforzamos conceptos clave aprendidos en clase, sino que también adquirimos nuevos conocimientos al enfrentarnos a errores y situaciones que nunca habíamos experimentado. Aprendimos a utilizar de manera más adecuada las ramas de trabajo, a resolver errores cuando no te dicen en que línea esta, que la planeación no solo se hace antes sino también durante el proceso y, sobre todo, comprendimos que no siempre se logra el resultado esperado, pero eso está bien, porque lo verdaderamente importante fue todo lo que aprendimos en el proceso.