Preinforme desafío I

Duvan Camilo Aragón Gallego

TI. 1033183697

David Fernando Revelo Morales

C.C 1085909373

Informática II

Aníbal Jose Guerra Soler

Universidad de Antioquia

Medellín

2025

Análisis del problema:

El desafío consiste en crear un sistema de streaming musical llamado UdeATunes, similar a plataformas como Spotify o iTunes. Tenemos que modelar toda la estructura usando programación orientada a objetos en C++, sin poder usar la STL. El sistema debe manejar usuarios con diferentes tipos de membresía, artistas con sus catálogos, álbumes, canciones, y todas las funcionalidades típicas de un servicio de streaming.

Lo interesante del proyecto es que no solo hay que hacer que funcione, sino que también debemos medir qué tan eficiente es nuestro código en términos de iteraciones y memoria consumida. Además, la restricción de no poder usar STL hace que tengamos que implementar nuestras propias estructuras de datos, lo cual es el verdadero reto.

Contexto del Sistema:

UdeATunes busca posicionarse como una plataforma líder en Colombia. El sistema diferencia entre dos tipos de usuarios: los estándar que no pagan y los premium que pagan 19900 pesos mensuales. Esta diferencia no es solo estética, porque cada tipo tiene capacidades completamente distintas en cuanto a calidad de audio, publicidad, y funcionalidades disponibles.

Problemática Principal:

Tenemos que resolver varios problemas simultáneos:

- **Gestión de usuarios:** Diferenciar entre estándar y premium, cada uno con sus propias restricciones y privilegios.
- **Organización del catálogo:** Estructurar correctamente la jerarquía artista, álbum, canción.
- **Sistema de reproducción:** Implementar un reproductor que se comporte diferente según el tipo de usuario.
- **Publicidad inteligente:** Mostrar anuncios a usuarios estándar de forma aleatoria pero ponderada según prioridades.
- **Persistencia de datos:** Diseñar un formato de archivos para guardar y cargar toda la información.
- **Medición de recursos:** Calcular exactamente cuántas iteraciones y cuánta memoria consume cada operación.

Entidades Identificadas:

1. **Usuarios:**

   El sistema maneja dos tipos de usuarios que son bastante diferentes:

   Usuarios Estándar (gratuitos):

   - Solo pueden escuchar música a 128 kbps
   - Tienen que ver publicidad cada dos canciones
   - No pueden retroceder a canciones anteriores
   - No tienen listas de favoritos

   Usuarios Premium (pagan 19900 pesos/mes):

   - Escuchan todo en alta calidad de 320 kbps
   - No ven ninguna publicidad
   - Pueden crear una lista de favoritos con hasta 10000 canciones
   - Pueden seguir las listas de favoritos de otros usuarios premium
   - Pueden retroceder hasta 4 canciones en reproducción aleatoria
   - Pueden retroceder hasta 6 canciones cuando reproducen sus favoritos

   Todos los usuarios tienen: nickname único, ciudad, país, y fecha de inscripción.

2. **Artistas:**

   Los artistas son los que producen el contenido de la plataforma. De ellos guardamos: código identificador, nombre, edad, país de origen, cantidad de seguidores que tienen en la plataforma, su posición en el ranking de tendencias, y su catálogo completo de álbumes.

3. **Álbumes:**

   Cada álbum tiene: código identificador, nombre, fecha de lanzamiento, duración total, sello disquero que lo publicó, ruta donde está guardada la portada (formato PNG), puntuación que le han dado los usuarios (del 1 al 10), y la colección de canciones que contiene.

   Una particularidad es que cada álbum puede clasificarse en hasta cuatro géneros diferentes. Los géneros disponibles son: Pop, Rock, Jazz, Música Clásica, Electrónica, Hip Hop, Reggae, Blues, y Latina.

4. **Canciones:**

Las canciones son la unidad básica del sistema y tienen varias características importantes:

El **identificador** es un número de 9 dígitos que está dividido en tres partes: los primeros 5 dígitos identifican al artista, los siguientes 2 identifican el álbum, y los últimos 2 identifican la canción específica. Por ejemplo, el ID 123450102 significa: artista 12345, álbum 01, canción 02.

Cada canción tiene: nombre, duración, cantidad de veces que ha sido reproducida, los créditos de quienes participaron en su creación, y dos archivos de audio (uno a 128 kbps y otro a 320 kbps, ambos en formato .ogg).

5. **Créditos:**

   Para cada canción guardamos quiénes participaron en su creación, divididos en tres categorías: Productores, Músicos, y Compositores. De cada uno guardamos: nombres, apellidos, y código de afiliación a la sociedad de autores (10 caracteres alfanuméricos). Esta información es importante porque en el futuro se usará para calcular las regalías que debe recibir cada persona cuando se reproduce una canción.

6. **Publicidad:**

   Como los usuarios estándar no pagan, tienen que ver publicidad. El sistema puede almacenar hasta 50 mensajes publicitarios de máximo 500 caracteres cada uno.

   La publicidad tiene tres categorías con diferentes prioridades:

   - Categoría AAA: tiene el triple de probabilidad de aparecer
   - Categoría B: tiene el doble de probabilidad
   - Categoría C: probabilidad normal

   El sistema debe asegurarse de que nunca aparezca el mismo mensaje dos veces seguidas.

Funcionalidad Requerida:

- **Carga y Actualización de Datos:**
  Esta funcionalidad no aparece en el menú del usuario pero es fundamental para el sistema. Debe leer y escribir información desde archivos de almacenamiento permanente. Tenemos que diseñar nosotros mismos el formato de esos archivos considerando: usuarios, artistas, álbumes, canciones, listas de reproducción, y mensajes publicitarios.

- **Ingreso a la Plataforma:**

Un sistema simple de autenticación que verifica el nickname del usuario y muestra el menú apropiado según si es usuario estándar o premium.

- **Reproducción Aleatoria:**

  Esta funcionalidad permite reproducir canciones de forma completamente aleatoria de todo el catálogo disponible. Al reproducir, debe mostrar en pantalla la ruta del archivo que se está reproduciendo y la ruta de la portada del álbum.

  Para usuarios estándar: deben ver publicidad cada dos canciones. Solo tienen opciones para: iniciar reproducción, detener reproducción, y pasar a la siguiente canción.

  Para usuarios premium: además de las opciones estándar, pueden pasar a la canción anterior (hasta 4 canciones hacia atrás) y activar el modo repetir (que reproduce la canción actual indefinidamente hasta desactivarlo).

  Para poder probar esta funcionalidad fácilmente, implementamos un temporizador de 3 segundos que cambia automáticamente a la siguiente canción, y el sistema se detiene después de reproducir 5 canciones.

- **Mi Lista de Favoritos (Solo Premium):**

  Esta funcionalidad tiene tres partes:

  a) Editar favoritos: El usuario puede buscar canciones por su ID y agregarlas o quitarlas de su lista. La lista no puede tener canciones duplicadas y tiene un límite de 10000 canciones.

  b) Seguir otra lista: El usuario puede buscar otro usuario premium por su nickname y copiar toda su lista de favoritos. Si ya tenía una lista propia, ambas se fusionan.

  c) Ejecutar favoritos: Permite reproducir la lista de favoritos, ya sea en el orden original o de forma aleatoria. Aquí el usuario puede retroceder hasta 6 canciones (en lugar de 4 como en modo aleatorio).

- **Medición de Recursos**

  Esta es una funcionalidad crítica que vale 15% de la nota. Al terminar cualquier funcionalidad, el sistema debe mostrar dos métricas:

  1. La cantidad total de iteraciones que se ejecutaron (directa e indirectamente) para completar esa funcionalidad.
  2. El total de memoria que se está consumiendo en ese momento, considerando TODAS las estructuras de datos, objetos, variables locales y parámetros por valor.

Estructuras de Datos Necesarias:

Se emplean estructuras de datos basados en arreglos dinámicos y punteros.

Cada clase gestionará internamente sus elementos mediantes arreglos dinámicos de punteros, junto con variables que controlan la cantidad y capacidad de elementos. Permitiendo administrar eficientemente la memoria y facilitar las operaciones de inserción, búsqueda y eliminación.

Además, se hará uso de la clase string para manejar cadenas de texto, ya que facilita la manipulación de datos leídos desde archivos externos.

El almacenamiento permanente de la información se realiza mediante archivos externos en formato CSV. Al iniciar estos datos serán leídos y cargados dentro de la estructura dinámica correspondiente, al finalizar estos datos serán guardados nuevamente en los archivos para mantener la persistencia en el sistema.

Decisiones de Arquitectura

Sistema Centralizado: Vamos a crear una clase “SistemaUdeATunes” que actúe como controlador principal. Esta clase va a gestionar todas las colecciones (usuarios, artistas, etc.) y va a coordinar las operaciones del sistema.

Manejo de Tipos de Usuario: Decidimos usar un solo tipo de clase Usuario con un atributo que indica si es estándar o premium. En cada método verificamos el tipo para determinar qué puede hacer el usuario.

Lógica de Reproducción: Toda la lógica de reproducción va a estar en la clase Usuario. Esto incluye mantener un historial de las canciones reproducidas (para poder retroceder), controlar cuándo mostrar publicidad, y manejar los diferentes controles según el tipo de usuario.

Conclusiones

Por ahora, nos concentramos en conectar las clases y mejorar el diagrama de clases actual, debido a que consideramos que hacen falta más cosas, una vez logremos eso, iniciaremos a programar nuestro sistema.