**INFORME PRELIMINAR DESAFÍO 2**

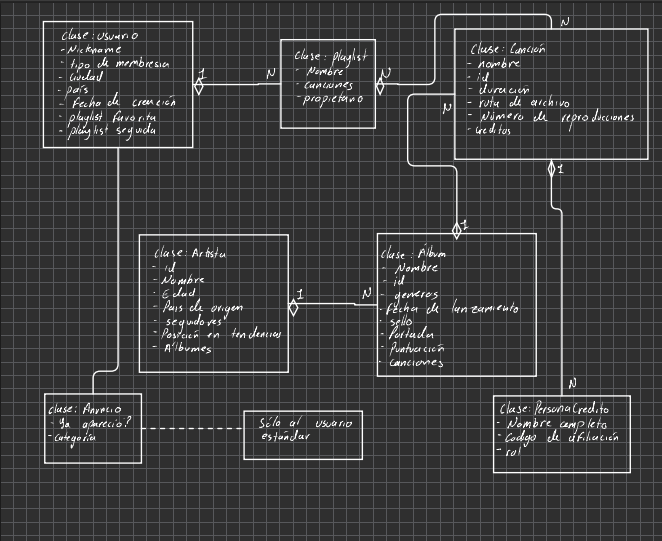
**INFORMÁTICA II**

**Integrantes:** Sara Echeverri Giraldo**,** Luis Miguel Alzate Rios.

**Introducción**

Este documento presenta el informe preliminar del proceso de análisis y diseño para la solución del Desafío II del curso Informática II. Esta etapa presenta una comprensión detallada del problema, los fundamentos del diseño orientado a objetos adoptado y la forma en que se gestionará la memoria mediante estructuras de datos dinámicas sin usar STL. La propuesta se basa en el diagrama de clases que se mostrará a continuación, que define la arquitectura general del sistema de reproducción de música UdeATunes.

**Diagrama de clases:**

****

**Análisis del problema**

El sistema UdeATunes representa una plataforma de reproducción musical con funcionalidades similares a las de servicios reales de streaming. El objetivo es modelar un entorno que permita la gestión de usuarios, artistas, álbumes, canciones, listas de favoritos y anuncios publicitarios, considerando diferencias funcionales entre tipos de usuarios (estándar y premium).

Durante el siguiente análisis se identifican los procesos fundamentales:

* Reproducción de canciones, tanto de forma secuencial como aleatoria.
* Gestión de listas de favoritos y la posibilidad de seguir favoritos de otros usuarios (uno a la vez).
* Aparición de anuncios publicitarios con ponderación por categoría y sin repetición.
* Registro de estadísticas como cantidad de reproducciones, iteraciones y consumo de memoria.
* Control de las reglas de membresía (calidad de audio, límite de retrocesos, presencia de anuncios).

El comportamiento de cada proceso está definido en el modelo de clases presentado en la imagen anterior, el cual estructura el sistema en módulos funcionales coherentes, garantizando la encapsulación y la claridad del flujo de datos.

**Diseño de la solución**

El diseño del sistema se basa en el diagrama de clases aprobado, compuesto por las clases Usuario, Playlist, Canción, Álbum, Artista, PersonaCrédito y Anuncio. Cada una cumple una función específica dentro de la plataforma musical:

* ***Usuario:*** contiene la información personal, tipo de membresía y ubicación del usuario. Además, mantiene dos listas de reproducción: una propia (favoritos personales) y otra que apunta a la lista de reproducción de otro usuario al que sigue. Estas dos listas de reproducción se guardan en un arreglo de dos posiciones, lo que permite cambiar fácilmente la lista de reproducción seguida sin modificar la estructura completa.
* ***Lista de reproducción:*** almacena las canciones asociadas a un usuario. Las listas de reproducción se guardan en un arreglo semidinámico, donde la primera dimensión (estática) contiene punteros, y cada puntero apunta a un arreglo pequeño donde se guardan las canciones. Cuando uno de esos arreglos se llena, se empieza a usar el siguiente puntero disponible. Las posiciones vacías apuntan a nulo.
* ***Canción:*** representa una pista musical con atributos como nombre, duración, ID, ruta de archivo, número de reproducciones y créditos asociados. Estas canciones son los elementos que se almacenan en las listas de reproducción y en los álbumes de los artistas.
* ***Álbum:*** agrupa un conjunto de canciones y puede incluir hasta cuatro géneros musicales. Además, contiene información como fecha de lanzamiento, sello discográfico, portada y calificación.
* ***Artista:*** contiene la información del creador, como nombre, país, edad, cantidad de seguidores y álbumes publicados. Desde esta clase se puede acceder a todas las canciones de sus álbumes para reproducir su repertorio completo o de forma aleatoria.
* ***PersonaCrédito:*** guarda la información de las personas que participaron en la producción de una canción (por ejemplo: productores o compositores).
* ***Anuncio:***representa los mensajes publicitarios que se mostrarán a los usuarios estándar. Todos los anuncios se almacenan juntos en un único arreglo, donde los de categoría AAA están al inicio, los de tipo B en el centro y los C al final. Los anuncios aparecerán de forma aleatoria, pero los de tipo AAA tendrán más probabilidad de mostrarse y los de tipo C, menos.

**Gestión de memoria y estructuras de datos**

El manejo de memoria se realiza sin usar STL, empleando arreglos de punteros y estructuras semi-dinámicas que permiten controlar directamente el espacio utilizado:

* Las listas de reproducción creadas por los usuarios se guardan en un arreglo semidinámico con una primera dimensión estática de punteros. Cada puntero apunta a un arreglo pequeño de canciones; cuando un bloque se llena, el siguiente puntero apunta a un nuevo bloque, mientras las posiciones libres apuntan a nulo.
* Las listas de reproducción del usuario (propia y seguida) se guardan en un arreglo de dos posiciones. La primera posición apunta a la lista de reproducción del propio usuario y la segunda a la del usuario seguido. Si el usuario deja de seguir a alguien y comienza a seguir a otro, solo se actualiza ese puntero, sin necesidad de modificar toda la estructura.
* Los anuncios se almacenan en un único arreglo, donde el orden físico de las categorías facilita su acceso. Sin embargo, la selección del anuncio será aleatoria, con mayor probabilidad de aparición para los AAA, intermedia para los B y menor para los C.

Este enfoque permite manejar la memoria de forma controlada y eficiente, cumpliendo con las restricciones del proyecto y garantizando independencia entre los distintos elementos del sistema.

**Conclusiones**

El proceso de análisis y diseño permitió definir una estructura lógica clara, basada en clases bien delimitadas y coherentes entre sí. La separación de responsabilidades entre las clases Usuario, Lista de reproducción, Canción, Álbum, Artista, PersonaCrédito y Anuncio garantiza una implementación organizada y fácil de mantener.

El uso de arreglos semidinámicos y punteros ofrece un control directo sobre la memoria, cumpliendo con la restricción de no usar STL y adaptándose a las necesidades de crecimiento del sistema. Además, la decisión de manejar las listas de reproducción del usuario (propia y seguida) mediante punteros independientes simplifica las operaciones de seguimiento y cambio, evitando la redundancia de datos. A su vez, el método propuesto para almacenar los anuncios permite un acceso uniforme y un control de aparición basado en prioridad, garantizando variedad y evitando patrones repetitivos.

En conjunto, el diseño aprobado y las estrategias de gestión de memoria planteadas sientan una base sólida para la implementación del sistema UdeATunes, asegurando eficiencia, claridad estructural y facilidad de expansión en futuras etapas del desarrollo.