

INFORME PLELIMINAR DESAFIO II INFORMATICA II

Manuel Felipe Salazar Burgos

Universidad de Antioquia

Index Terms— Programación orientada a objetos; C++; plataformas de streaming; UdeATunes; diseño de clases; eficiencia; listas de reproducción; reproducción de música; publicidad dinámica.

1. INTRODUCCIÓN

La música forma parte importante de la vida cotidiana de las personas, ya que permite expresar emociones, mejorar el estado de ánimo y facilitar la concentración. Con el paso del tiempo, las formas de escuchar música han cambiado. Antes se usaban medios físicos como discos o CD, pero actualmente la mayoría de las personas utiliza plataformas de *streaming*, que permiten reproducir canciones directamente desde internet.

El proyecto **UdeATunes** tiene como objetivo desarrollar un sistema que simule el funcionamiento básico de una plataforma de *streaming* musical, aplicando los principios de la *Programación Orientada a Objetos* (POO) en el lenguaje C++. El sistema debe permitir gestionar usuarios, artistas, álbumes, canciones y mensajes publicitarios, considerando las diferencias entre los usuarios estándar y los usuarios premium.

Este trabajo busca aplicar de forma práctica los conceptos aprendidos durante el curso, como la abstracción, la encapsulación, las relaciones entre clases, la sobrecarga de métodos y operadores, y el uso de memoria dinámica. A través de este modelo se pretende representar una versión simplificada de un servicio real de reproducción musical.

En esta primera fase se presenta el análisis del problema y la propuesta de diseño del sistema. Se incluye la identificación de las clases principales, sus relaciones y el diagrama de clases inicial que servirá como base para la implementación del programa en las siguientes etapas.

2. ANÁLISIS DEL PROBLEMA Y CONSIDERACIONES PARA LA ALTERNATIVA DE SOLUCIÓN PROPUESTA

El problema principal que aborda este proyecto consiste en diseñar e implementar un sistema que represente de manera estructurada el funcionamiento básico de una plataforma de *streaming* musical. Este sistema debe permitir la interacción entre diferentes entidades como usuarios, artistas, álbumes,

canciones y mensajes publicitarios, considerando además las particularidades de cada tipo de usuario.

El desafío no solo implica desarrollar las funcionalidades descritas, sino también aplicar los principios de la *Programación Orientada a Objetos* (POO) de forma adecuada. Esto requiere identificar las clases necesarias, definir sus atributos y métodos, establecer las relaciones entre ellas y garantizar que el sistema sea modular, extensible y eficiente en el uso de memoria.

Entre los aspectos más importantes del problema se encuentran:

- La gestión de dos tipos de usuarios: **estándar** y **premium**, cada uno con permisos y características diferentes.
- La representación de la estructura musical mediante **artistas**, **álbumes** y **canciones**, estableciendo relaciones jerárquicas y de composición.
- La implementación de un sistema de **publicidad** que se muestre de forma aleatoria a los usuarios estándar, considerando distintos niveles de prioridad.
- La **reproducción de canciones** con opciones adicionales para los usuarios premium, como avanzar, retroceder o repetir una pista.
- La necesidad de medir el **consumo de recursos** del sistema, tanto en memoria como en número de iteraciones ejecutadas.

Para abordar este problema se propone una alternativa de solución basada en el diseño de un conjunto de clases que representen de forma clara las entidades principales del sistema. El diseño parte de los siguientes lineamientos:

- Utilizar relaciones de **composición** entre las clases **Artista**, **Álbum** y **Canción**, de manera que un álbum contenga sus canciones y un artista agrupe sus álbumes.
- Aplicar los principios de **encapsulación** y **abstracción** para proteger los datos internos de cada clase y facilitar el mantenimiento del código.

- Incorporar **constructores de copia, sobrecarga de operadores y métodos amigos** cuando sea necesario para optimizar el manejo de datos y las operaciones entre objetos.
- Implementar estructuras de datos **dinámicas y propias**, evitando el uso de la biblioteca STL, tal como se especifica en los requisitos del desafío.
- Garantizar la **eficiencia** del sistema mediante una adecuada gestión de memoria y control del flujo de ejecución.

Con esta propuesta se busca construir una base sólida para la implementación del sistema en C++, permitiendo que las funcionalidades solicitadas se desarrollen de manera ordenada, modular y coherente con el paradigma orientado a objetos.

3. DIAGRAMA DE CLASES Y DESCRIPCIÓN DE LAS ENTIDADES

En la Figura 1 se presenta el diagrama de clases propuesto para el sistema UdeATunes. El diseño refleja las relaciones de composición y herencia necesarias para modelar los diferentes elementos del servicio de *streaming*.

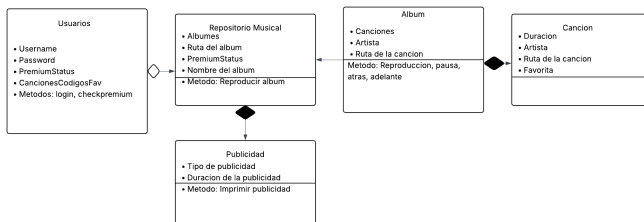


Fig. 1: Diagrama de clases propuesto para UdeATunes.

A continuación, se describen las principales clases del sistema:

- **Usuario:** Representa a cada persona registrada. Contiene datos como nombre, país, tipo de membresía y fecha de inscripción.
- **UsuarioPremium:** Hereda de **Usuario**. Puede crear y gestionar una lista de canciones favoritas y reproducir música sin anuncios.
- **UsuarioEstandar:** Hereda de **Usuario**. Reproduce canciones con publicidad después de cada dos pistas.
- **Artista:** Contiene la información del artista y la lista de sus álbumes.
- **Álbum:** Agrupa las canciones y almacena su información general (nombre, duración, portada, puntuación, géneros).

- **Canción:** Representa cada pista individual con su nombre, duración, identificador y ruta de los archivos de audio.
- **Publicidad:** Administra los anuncios del sistema y su prioridad.
- **Plataforma:** Gestiona la reproducción musical, los usuarios y la base de datos general del sistema.