



Universidad de Antioquia

Facultad de Ingeniería

2570201: Informática II

Informe de análisis desafío 2

Luis Daniel González Correa

Jean Paul Agudelo Gutiérrez

Ingeniería de Telecomunicaciones

Medellín, 2025



Universidad de Antioquia

Facultad de Ingeniería

2570201: Informática II

Informe de análisis desafío 2

Luis Daniel González Correa

Jean Paul Agudelo Gutiérrez

Ingeniería de Telecomunicaciones

Dr. Augusto Enrique Salazar Jiménez

Medellín, 2025

Contexto de introducción

UdeATunes es una plataforma de streaming musical que busca apoderarse de la melómanos en Colombia y a largo plazo de forma mundial, usando el lenguaje c++ y la programación orientada a objetos (POO), donde se tiene usuarios sin suscripción (usuarios estándar) y usuarios que pagan una suscripción de 19900 (usuarios premium), cabe recalcar que como usuarios estándar se presentarán anuncios cada cierta cantidad de canciones.

Los artistas, son de manera obvia la parte fundamental de la plataforma de UdeATunes, donde se almacenan sus códigos de identificación, edad, país de origen, cantidad de seguidores y posición en el listado global.

Cada álbum se divide en nombre, identificador, duración del álbum (duración de todas las canciones contenidas en él), ubicación de los archivos de audio.

Tendrá un sistema de créditos para productores, músicos y compositores. Este sistema se usará para calcular las regalías más adelante.

Análisis

Para poder realizar el proyecto se deben tener varias cosas en cuenta solo para la parte de los usuarios:

- I. Los usuarios que no poseen la versión premium escuchan en una calidad inferior la música a 128 kbps y cada 2 anuncios se deben presentar un anuncio dependiendo la prioridad del anuncio
- II. Los usuarios premium, además de escuchar en una calidad superior la música a 320 kbps, no escuchan anuncios, pueden crear una lista de canciones favoritas de máximo 10000 canciones, pueden escuchar música de otros usuarios.
- III. Información detallada de un identificador único para cada uno en caso de un álbum 9 dígitos
- IV. Duración de la canción, junto a los créditos de esta canción para repartir los créditos entre los artistas, productores y compositores.
- V. Ubicación y la ruta, tener en cuenta que se guarda o se reproduce en un archivo .ogg

Además de ello, se deben tener los espacios publicitarios dependiendo de la prioridad o categorías (C,B,AAA) que se mostrarán al usuario estándar dependiendo de la prioridad del mismo.

Finalmente, se requiere un módulo de medición de recursos que muestre, después de cada operación, la cantidad de iteraciones ejecutadas y la memoria total consumida por las estructuras activas.

El desafío debe ser de manera eficiente, escalable y modular, que por nuestra parte, lo intentaremos hacer lo mejor posible.

Solución preliminar

Cuando ejecutemos el código se cargarán los datos iniciales para UdeATunes, dándole espacio a los objetos necesarios a partir de las clases Usuario, usuarioEstandar, usuarioPremium, Artista, Álbumes, Cancion, Publicidad, Creditos y Plataforma. Cada clase contará con su propio constructor para inicializar los atributos y, cuando sea necesario, con un destructor encargado de liberar memoria dinámica, especialmente en las listas para ahorrar recursos

La clase **Plataforma** actuará como núcleo del sistema. Contendrá listas de usuarios, artistas y publicidades (usuarios, artistas, publicidades) y métodos como cargarDatos(), guardarDatos(), buscarUsuario() y reproduccionAleatoria(). Al iniciar la ejecución, cargarDatos() leerá la información almacenada en archivos y generará las instancias necesarias para el funcionamiento del programa.

Los usuarios estarán representados por la clase usuario, con atributos como nickName, tipoDeMembresia, ciudad, pais y fechaDeRegistro. Y para esta clase se deriva la de **usuarioEstandar**, que incluye el atributo anuncios y el método reproducirMusica(), el cual insertará una instancia de publicidad y la calidad de reproducción será de 128 kbps.

Y para los **usuarioPremium** tendrán una listaDeFavoritos y métodos como editarFavoritos(), seguirLista() y ajustarLista(). Los premiums podrán reproducir música en 320 kbps, acceder a listas de otros usuarios premium y repetir canciones indefinidamente.

La clase **Artista** manejará atributos como idArtista, nombre, edad, pais, seguidores, posicionGlobal y una lista de Album. Su método agregarAlbum() permitirá vincular nuevos álbumes y mostrarInfo() desplegará la información general del artista.

Cada **Album** estará identificado por idAlbum de nueve dígitos y contendrá datos como nombre, acronico, duracionTotal, fechaDeLanzamiento, sello, portada.

La clase **Cancion** incluirá los atributos idCancion, nombre, duracion, RutaDelArchivo, RutaDeLaImagen, y los créditos. Su método reproducir() simula como si el audio se reprodujera y mostrarInfo() los detalles de la canción.

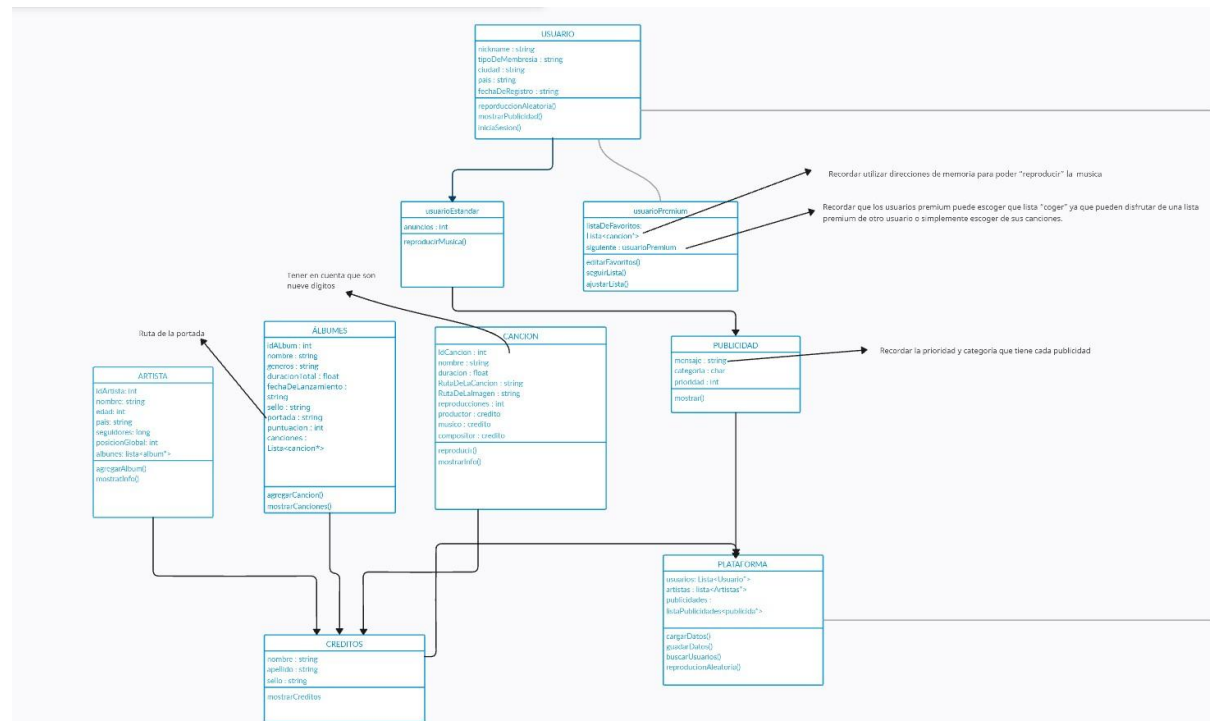
Los créditos estarán definidos en la clase **Creditos**, con atributos nombre, apellido y sello, y el método mostrarCreditos() para a futuros temas de regalías

La clase **Publicidad** gestiona anuncios para los usuarios estándar, con atributos como mensaje, categoria C, B o AAA, y prioridad.

Resultado esperado

Después que terminemos la implementación, esperamos obtener lo más real a una simulación de una plataforma de streaming. El prototipo final nos debe demostrar el correcto uso de los conceptos de POO, el manejo eficiente de memoria y la posibilidad de escalar el código.

Diagrama de clases



<https://app.creately.com/d/DfBJZHBYvEe/view>

Bibliografía

Salazar, A., & Guerra, A. (2025). *Guía desafío 2 - Informática II* [No publicado]. Universidad de Antioquia