**Informe del análisis utilizados en el desafío #2.**

**Alejandro Bedoya Zuluaga**

**Jeronimo Herrera Espinosa**

**1.Introducción**

En este desafío, abordaremos principalmente la temática y el modelo de Programación Orientada a Objetos (POO), el propósito de este será el de crear desde cero una aplicación de reproducción de música, en las cuales se tendrán que contemplar varias funciones y clases primordiales para su funcionamiento, y las cuales serán profundizadas más adelante.

Este informe detalla el análisis que decidimos aplicar, las clases junto con sus atributos y métodos podrán ser visualizados en el Diagrama de clases estilo UML simplificado.

**2.Descripción del problema**

En este reto se nos darán unos datos/listas en forma de archivos de texto plano .txt en los cuales se contemplarán los artistas, con sus respectivos álbumes y las respectivas canciones pertenecientes a estos, y a su vez también se nos darán las distintas listas de favoritos que un usuario premium puede poseer.

Para el funcionamiento de esta aplicación se deberá dar la posibilidad de iniciar sesión como un usuario existente, el cual podrá ser premium o estándar, para estas dos clases hay las siguientes ventajas/condiciones:

**Premium:** los usuarios premium pueden poseer una lista de favoritos personalizada, a su vez, pueden elegir seguir la lista de favoritos de otro usuario, podrán escuchar sus canciones en 390kbpsy podrán escuchar todas sus canciones sin publicidad.

**Estándar:** los usuarios estándar, tendrán que ver un anuncio cada dos canciones, estos mensajes publicitarios podrán ser de tres tipos y los cuales tendrán distinta prioridad, de menor a mayor, los tipos son C, B y AAA, también los usuarios estándar tendrán que escuchar las canciones en 128kbps.

**3.Analisis utilizado**

Para empezar, decidimos definir los siguientes archivos de texto para almacenar el data Set que se nos dará para la ejecución del programa realizado:

-**Usuarios:**  En este se almacenarán los datos(atributos) de los distintos usuarios ya registrados en el sistema, Un ejemplo de la estructura de una línea de dicho archivo .txt seria:

**Nickname, tipo de membresía, ciudad, país, fecha de registro, contraseña(password) <<endl**

**-ListasFavoritos:** En este se almacenarán las canciones pertenecientes a la lista de favoritos de cada usuario premium, por ejemplo, las listas dentro de dicho archivo .txt se almacenan de la siguiente manera:

**Dueño de la lista, idcancion1, idcancion2, idcancion3, … << endl.**

**-publicidad:** Dentro de este archivo se almacenan los mensajes publicitarios que serán mostrados a los usuarios estándar cada 2 canciones, un ejemplo seria:

**Categoría del mensaje, mensaje publicitario << endl**.

**-artistas:** Como su nombre lo indica, en este archivo de texto se almacenan los datos(atributos) de cada uno de los artistas guardados dentro del programa, por ejemplo:

**Código artista, nombre, edad, país, # de seguidores, posición << endl.**

**-álbumes:**  este archivo almacena todos los álbumes almacenados en la plataforma, los cuales podrán ser asignados a su artista creador gracias a un código, y los cuales comprenden la siguiente estructura dentro del archivo .txt:

**Código álbum (código artista + # del álbum), nombre del álbum, géneros, fecha lanzamiento, sello, portada << endl.**

**-canciones:** En este archivo se almacenan las canciones pertenecientes a cada uno de los álbumes de los artistas, dichas canciones (al igual que los álbumes) podrán ser identificadas gracias a un código, un ejemplo de un renglón/línea de dicho archivo es:

**Código canción (código artista + # del álbum + # de la canción), nombre, ruta canción 128, ruta canción 320, duración (segundos) << endl**

**-creditosCanciones:** en este archivo se almacenarán los créditos de las canciones para los productores, músicos y compositores que participaron en esta la cual seguirá una estructura del tipo:

**Código canción, productores: nombres/apellido/código de afiliados, músicos: nombres/ apellido/código de afiliados, compositores: nombre/apellido/código de afiliados << endl**

Para este desafío, se decidieron implementar ciertas clases primordiales, las cuales son:

**Plataforma:** la cual nos servirá como la base del todo y la primera interacción con los usuarios, ya que esta tiene relación de referencia con las clases Usuarios, a su vez en esta clase se encuentran los métodos utilizados para cargar los datos que estarán guardados en archivos externos de texto plano (.txt).

**Usuarios:** En esta clase se encontrará guardados, como lo indican su nombre los usuarios, junto con, en sus atributos, su tipo (premium y estándar), junto con los otros ya mencionados en el diagrama.

**Publicidad:** En esta clase se guardan los distintos mensajes publicitarios junto con su categoría, también en esta se encuentra la función en la cual se selecciona un mensaje publicitario al azar (con influencia en su categoría) y se muestra en pantalla a los usuarios estándar cada 2 canciones.

**Artista:** Esta clase modela al artista de las canciones registradas en la plataforma, en esta se almacena información relevante sobre el artista tales como el país, edad, id, nombre… y demás información relevante que pueda servir de referencia para la canción.

Esta clase mantiene una relación de uno a muchos con la clase álbum, ya que un artista puede tener muchos álbumes los cuales a su vez contendrán muchas canciones.

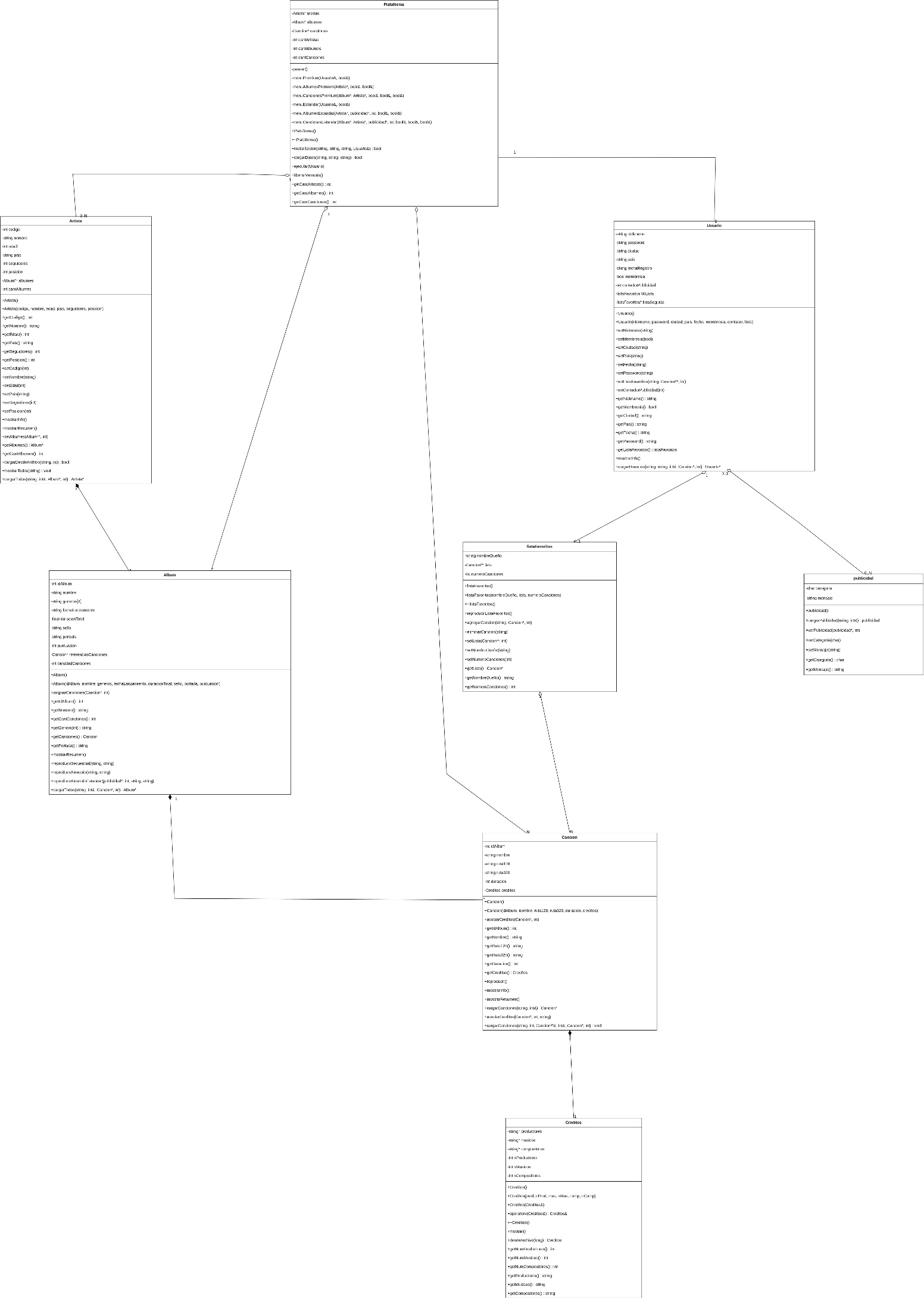
**Álbum:** Similar a la clase artista, contine información relevante sobre los álbumes de los artistas, estas a su vez se relacionan de uno a muchos con la clase canción, ya que un álbum puede tener varias canciones.

**canción:** En esta clase se almacenarán como su nombre lo indica las distintas canciones que se encuentren registradas en el programa (data set), junto con sus distintos atributos como las rutas a los archivos de audio de 128 y 320.

**Créditos:** En esta se guardan los créditos de las distintas canciones que se encuentran registradas.

**listaFavoritos:** Esta es la clase es debido a la funcionalidad de los usuarios premium, los cuales como se dijo ya, pueden poseer una lista de canciones favoritas con un máximo de hasta 10000 canciones, en esta los usuarios pueden crear playlist de favoritos y seguir a otra lista de favoritos de otro usuario.

El diagrama de clase que creamos es el siguiente



Para este desafío haremos uso de distintas librerías externas las cuales nos ayudaran con varias funcionalidades las cuales son:

- <string>

-<random>

-<fstream>

-<sstream>

-<cstdlib>

-<ctime>

-<thread>

, las cuales nos servirán para el funcionamiento del programa, como por ejemplo la reproducción aleatoria de las canciones dentro de una lista, la lectura de los archivos .txt y el manejo de los strings, etc.

**4.Para Finalizar**

En este desafío se decidió adoptar un enfoque muy basado en la eficiencia y sobre todo en memoria, con decisiones como el no leer todos los usuarios a la vez sino únicamente el usuario que se registre en ese momento al programa, etc.

Para concluir, en este desafío se adoptó un enfoque basado en la Programación Orientada a Objetos (POO), mediante la creación de diversos módulos y clases que permiten estructurar el programa de forma más clara y organizada. Este enfoque contribuye a mejorar la legibilidad del código y facilita la interacción entre los distintos objetos, promoviendo así un diseño modular, reutilizable y fácil de mantener.

Además, la implementación de las clases y sus relaciones permite representar de manera fiel los componentes principales del sistema, reflejando cómo interactúan entre sí dentro de la plataforma. Gracias a esta estructura, el desarrollo se vuelve más escalable y flexible, facilitando la incorporación de nuevas funcionalidades sin afectar el funcionamiento general del programa.