Pre Informe

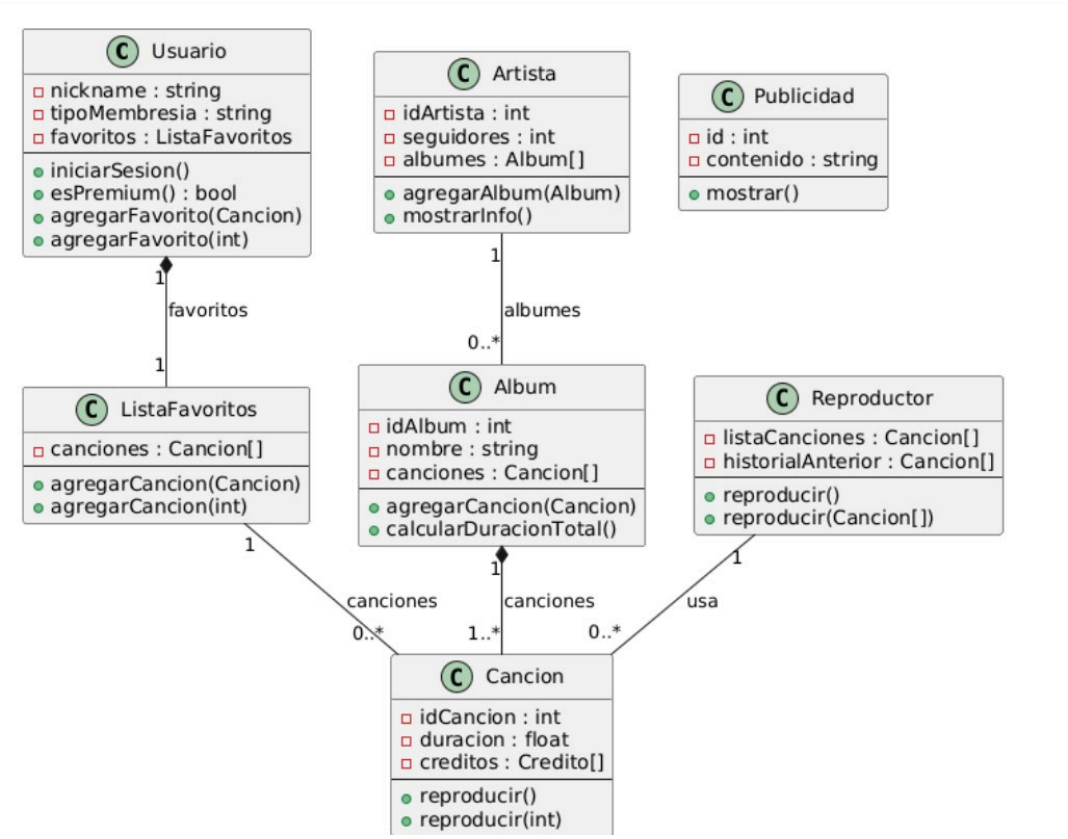
Desafío 2

Universidad de Antioquia

Estudiantes: Manuela Arboleda Montoya – Alexa Carolina Gamboa Mier

Analisis del problema:

El desafío es modelar y desarrollar un servicio de streaming musical llamado UdeATunes utilizando la Programación Orientada a Objetos (POO) en C++. El sistema debe gestionar eficientemente las entidades clave del negocio: usuarios (estándar y premium), artistas, álbumes, canciones, listas de reproducción y mensajes publicitarios.



Se incluirá en el código de los algoritmos más críticos, como la gestión del historial de reproducción, asegurando que el código tenga comentarios concisos y claros para explicar la lógica de las secciones no obvias.

Problemas de desarrollo:

- Gestión de la Persistencia de Datos: El diseño del formato de archivos para almacenar todas las entidades (usuarios, artistas, álbumes, canciones, publicidad) será crítico. Asegurar que la carga y la actualización sean eficientes y que se mantengan las complejas relaciones puede ser complejo.

- Lógica de Reproducción con Historial: Implementar el seguimiento del historial de hasta 4 o 6 canciones (para usuarios premium en reproducción aleatoria y favoritos, respectivamente) requiere una gestión cuidadosa de índices o punteros para garantizar que las canciones en modo "repetir" solo se cuenten una vez en el historial

- Medición de Consumo de Recursos: La métrica de total de memoria consumida por todas las estructuras de datos y objetos existentes en el sistema, incluyendo variables locales y parámetros por valor, es particularmente difícil de obtener con precisión en C++ estándar y requerirá el uso de librerías o técnicas específicas de bajo nivel, o una implementación de contador manual de bytes ocupados por cada objeto creado.

Consideraciones para tener en cuenta en la Implementación

- Uso de plantillas, librerías, validación de lógica, cadenas de ruta

Determinamos que el desafío más grande será manejar y guardar todos los datos del sistema (las canciones, los usuarios premium, los artistas, etc.). Se planea crear su propio sistema de archivos para guardar todo.

Sin embargo, hay que asegurarse de que el programa sea rápido y eficiente al buscar, guardar y actualizar la música y las cuentas. Si esto falla, el sistema será muy lento o se dañará.

Necesitamos escribir un código inteligente que sepa elegir y mostrar anuncios al azar, pero de forma equilibrada, para que el servicio sea rentable, y tenrmos que programar cómo el sistema recuerda qué ha escuchado cada usuario. Esto es esencial para que la aplicación funcione como un servicio de streaming real y pueda hacer recomendaciones.