



## INFORME DESAFIO II

**Jharlin Castro Moreno**  
Jharlin.castro@udea.edu.co

**Mauricio Rafael Aguas Ramírez**  
mauricio.aguas@udea.edu.co

\*Departamento de electrónica y telecomunicaciones, Facultad de ingeniería, UDEA, Medellín, Antioquia, Colombia

**Resumen:** Este informe presenta el análisis y diseño preliminar de una solución al Desafío 2 de la asignatura Informática II, el cual consiste en el desarrollo de un sistema de streaming musical denominado UdeATunes. Este proyecto representa una evolución natural de los conceptos aprendidos en programación estructurada hacia el paradigma de Programación Orientada a Objetos (POO) utilizando el lenguaje C++ con el framework Qt. Este pre-desarrollo constituye la base para la implementación final.

**Palabras claves:** Programación Orientada a Objetos (POO), Sistema de Streaming Musical, C++, Herencia y Polimorfismo, Memoria Dinámica, Estructuras de Datos, Algoritmos de Búsqueda, Gestión de Usuarios Premium/Estándar, Persistencia de Datos, Abstracción y Encapsulación, Diagrama UML, Optimización de Rendimiento

### Introducción

El presente informe corresponde a la etapa de pre-desarrollo del Desafío 2. UdeATunes surge como una propuesta académica que emula el funcionamiento de plataformas de streaming musical populares como Spotify, Apple Music o Amazon Music, pero implementada completamente en C++ utilizando únicamente memoria dinámica y estructuras de datos propias, sin el uso de la Standard Template Library (STL).

### Contextualización

El desarrollo de un sistema de streaming musical presenta un contexto ideal para la aplicación de conceptos avanzados de programación orientada a objetos.

La naturaleza del dominio permite modelar entidades del mundo real (usuarios, artistas, álbumes, canciones) y sus interrelaciones complejas, proporcionando un terreno fértil para la implementación de jerarquías de clases, polimorfismo, abstracción y encapsulación

### Análisis del problema

El sistema UdeATunes debe resolver múltiples problemáticas interrelacionadas que reflejan la complejidad de los servicios de streaming musical modernos. La principal complejidad radica en la gestión diferenciada de dos tipos de usuarios con experiencias completamente diferentes: los usuarios estándar (gratuitos) que tienen acceso limitado con calidad de audio reducida de 128 kbps, publicidad cada 2 canciones y funcionalidades básicas de reproducción sin capacidad de navegación hacia atrás; y los usuarios premium que pagan \$19,900 COP mensuales y reciben acceso completo con calidad superior de 320 kbps, sin publicidad, listas de favoritos personalizables hasta 10,000 canciones, capacidad de seguir listas de otros usuarios

y navegación avanzada que les permite retroceder hasta 4 canciones en reproducción general y hasta 6 canciones cuando están en sus favoritos.

La gestión publicitaria representa uno de los aspectos más complejos del sistema, ya que debe ser estratégica y no intrusiva. El sistema maneja un máximo de 50 mensajes publicitarios activos, cada uno con contenido limitado a 500 caracteres. Implementa un sistema de prioridades ponderadas donde la categoría C tiene prioridad simple (peso 1), la categoría B tiene prioridad doble (peso 2) y la categoría AAA tiene prioridad triple (peso 3). El algoritmo de selección debe evitar repeticiones consecutivas del mismo mensaje y presentarse automáticamente cada 2 canciones para usuarios estándar, utilizando una selección por ruleta ponderada que considera las prioridades de cada mensaje.

Los casos de uso críticos que el sistema debe manejar incluyen autenticación segura y gestión de sesiones de usuario, búsquedas eficientes por identificador de canción que requieren decodificación de los 9 dígitos para localizar rápidamente artista, álbum y canción específica, gestión de listas de favoritos con validaciones de límites y sin permitir duplicados, reproducción continua con transiciones fluidas entre canciones y publicidad, y sincronización de estados entre diferentes módulos del sistema para mantener coherencia en la experiencia de usuario. El sistema debe implementar cinco funcionalidades esenciales: carga y actualización de datos desde almacenamiento permanente, ingreso a la plataforma con autenticación diferenciada según tipo de usuario, reproducción aleatoria con controles específicos para cada tipo de membresía, gestión de lista de favoritos exclusiva para usuarios premium que incluye agregar, eliminar, editar y ejecutar listas, y medición de recursos con métricas detalladas de

iteraciones y consumo de memoria para evaluar el rendimiento del sistema.

## **Diseño propuesto**

La solución arquitectónica propuesta se basa en los principios de la Programación Orientada a Objetos, implementando un diseño modular, escalable y eficiente que aborda cada una de las problemáticas identificadas. 4.1 Arquitectura General del Sistema Se adopta una arquitectura en capas inspirada en el patrón Model-View-Controller (MVC):

Capa de Presentación: Interfaz de consola con menús contextuales según tipo de usuario

Capa de Lógica de Negocio: Controladores que gestionan las reglas de negocio complejas

Capa de Datos: Entidades del dominio y gestores de persistencia

Capa de Utilidades: Estructuras de datos propias y algoritmos de apoyo

## Diagrama de Clases Propuesto

