Desafío II

Informática II

Semestre: 2025-2

Integrantes del equipo

Ana Sofia Agudelo Blandon  
Keiner Marcelo Torres Villeros

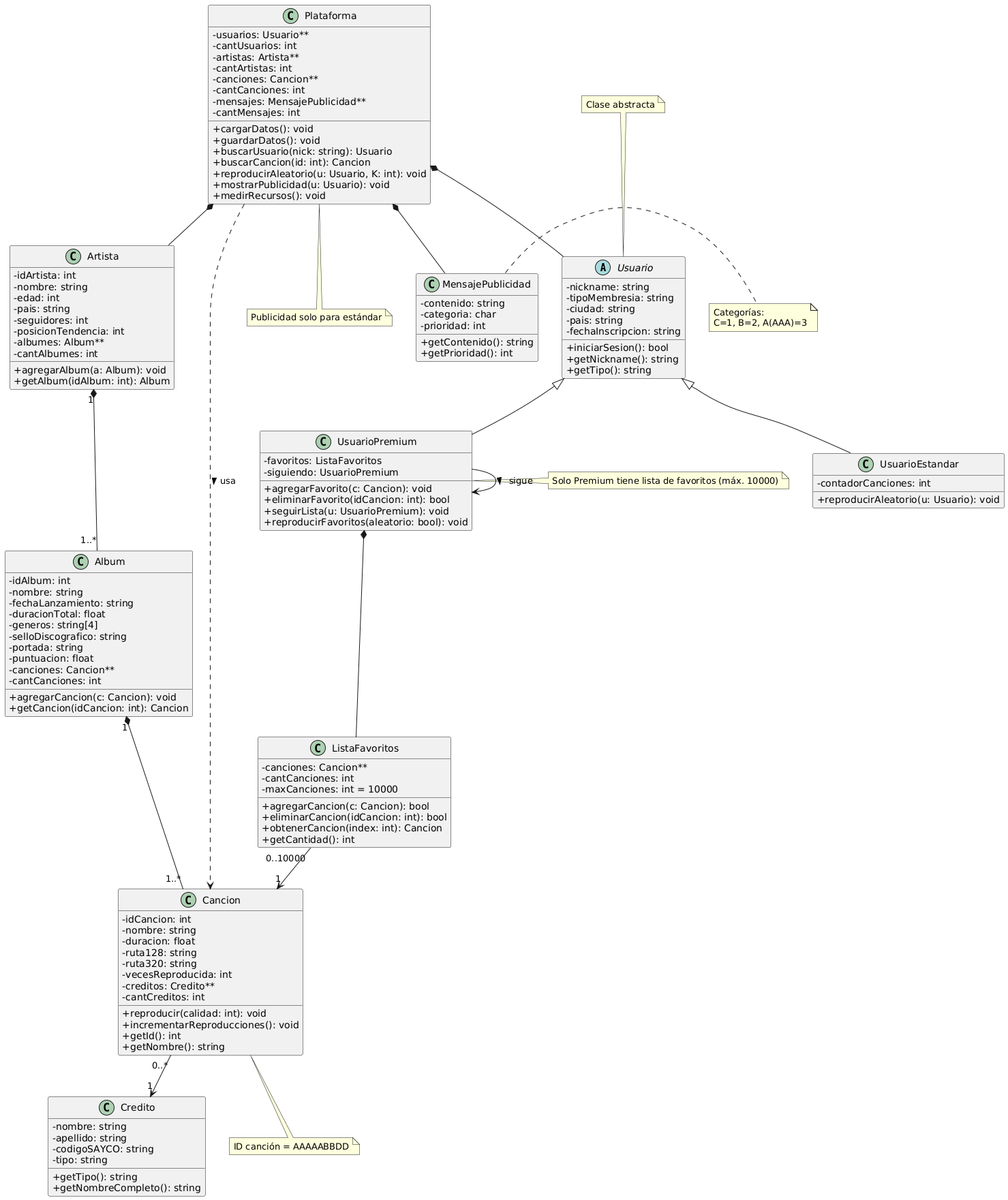
Fecha de entrega

Primera entrega: 17 de octubre

1. Análisis del problema y consideraciones para la alternativa de solución propuesta.

R// La idea detrás del desafío es crear un sistema de streaming musical llamado UdeATunes, que imite plataformas comerciales como Spotify o Apple Music. El reto consiste en modelar entidades del mundo real, como usuarios, artistas, álbumes y canciones, y entender sus complejas interrelaciones, además de incluir funciones específicas para la reproducción, gestión de listas y publicidad. La solución que proponemos busca equilibrar la complejidad funcional con la eficiencia técnica, aplicando principios sólidos de programación orientada a objetos, todo mientras se cumplen las restricciones establecidas (sin usar STL y con manejo manual de memoria). El diseño fue concebido, con fácil mantenimiento y expansión en memoria, está preparado para amoldarse a futuras actualizaciones del sistema.

1. Diagrama de clases de la solución planteada. Adicionalmente, describa en alto nivel la lógica de las tareas que usted definió para aquellos subprogramas cuya solución no sea trivial.

R// 

**Herencia Usuario → UsuarioEstandar UsuarioPremium:**  
La herencia se usa, porque los dos tipos de usuario comparten atributos y comportamiento base (login, datos personales), aunque, tienen comportamientos distintos como la publicidad, favoritos y navegación previa que se modelan en las subclases.  
**Composición Artista → Album, Album → Cancion, UsuarioPremium → ListaFavoritos:** Estas relaciones, indican la dependencia de existencia, pues un Álbum, no tiene sentido fuera del Artista que lo creo, además, una Canción pertenece solamente a un Álbum, también, la ListaFavoritos pertenece al UsuarioPremium y se destruye si el usuario se elimina.  
**Asociaciones Cancion ↔ Credito, UsuarioPremium ↔ UsuarioPremium:**Los créditos de la producción, músicos y compositores, se asocian a una canción; un usuario premium puede seguir la lista de otro, eso es una asociación con cardinalidad.  
**Dependencia Plataforma → Cancion, Plataforma → MensajePublicidad:** Plataforma utiliza canciones y mensajes para ofrecer funcionalidades, como reproducción aleatoria o para mostrar anuncios, pero no es necesario que Plataforma “cree” las canciones pues las administra, así es como modelamos la dependencia/uso.  
**Estructuras de datos arreglos dinámicos:** El enunciado prohíbe STL, por eso en el UML se muestra Tipo contador cantX para indicar los arreglos dinámicos manejados manualmente con new/delete, lo que cumple la restricción.  
**Notas operativas publicidad, límites:** Las notas facilitan comprobar las reglas funcionales por ejemplo, límite de 10.000 favoritos. ¿Sabes? 50 anuncios, prioridad, la publicidad aparece después de dos canciones, etc.

1. Algoritmos implementados debidamente intra-documentados. No exceda la intra-documentación. No use IA para generar la documentación.
2. Problemas de desarrollo que afrontó.
3. Evolución de la solución y consideraciones para tener en cuenta en la implementación.