

Vive la música con UdeATunes

Uno de los objetivos de programar consiste en modelar una situación del mundo real, delimitada por un contexto. Al representar en el programa las entidades y acciones de interés, se pueden estudiar o automatizar las dinámicas propias del contexto. Si el programa es un modelo apropiado de la situación real, entonces los resultados de su ejecución reflejan las interacciones reales; con cierta confianza.

Objetivos

- Desarrollar la capacidad de análisis y solución de problemas en los estudiantes, enfrentándolos a problemáticas de la vida cotidiana.
- Verificar si el estudiante adquirió las destrezas y conocimientos fundamentales de la programación orientada a objetos en C++: abstracción, encapsulación, relaciones, diseño de diagrama de clases, funciones amigas, sobrecarga y uso de plantillas.

Si usted ha llevado un proceso disciplinado de aprendizaje a lo largo del semestre, esta es una oportunidad de demostrarlo y podrá plantear una solución satisfactoria. En caso contrario, podrá identificar sus debilidades y tomar medidas a fin de poder abordar situaciones similares a futuro.

Trate de valorar la verdadera complejidad del problema planteado, no se rinda antes de intentarlo o de plantear los posibles escenarios de solución. Se dará cuenta que si bien, al principio le puede parecer difícil; ya ha tenido la oportunidad de enfrentarse a problemas similares. Si se toma el tiempo adecuado para analizar, el proceso de codificación será eficaz y no le tomará mucho tiempo.

Esperamos que disfrute del desafío propuesto. Lea primero todo el documento antes de comenzar y asegúrese de entender muy bien las instrucciones antes de desarrollar esta actividad evaluativa.

Fue revisado por los profesores Aníbal Guerra y Augusto Salazar.

Introducción

Escuchar música ofrece múltiples beneficios, desde la mejora del estado de ánimo, la reducción del estrés y la ansiedad; hasta el alivio del dolor, la mejora de la memoria y la concentración. La música activa diferentes áreas del cerebro, promoviendo la

liberación de endorfinas y otras sustancias que generan bienestar y placer, ayudando a impulsar el rendimiento físico durante el ejercicio, facilitar el aprendizaje, la cognición y mejorar la calidad del sueño (OpenAI, 2025).

En el pasado, la reproducción musical se realizaba principalmente mediante soportes físicos, como discos y CD; así como a través de formatos digitales, entre los que destacaban archivos MP3 y WMA. En la actualidad, esta práctica ha sido sustituida en gran medida por el uso de plataformas de *streaming* musical, que concentran la mayor parte del consumo de audio a nivel global. Estas plataformas ofrecen acceso a un reproductor que permite escuchar la música directamente de internet, sin necesidad de almacenar los archivos en tu dispositivo permanentemente.

El objetivo de este desafío consiste en desarrollar un programa que emule el funcionamiento de un servicio de streaming, utilizando POO. El sistema debe permitir la gestión eficiente de las canciones, álbumes, artistas usuarios y listas de reproducción, entre otros.

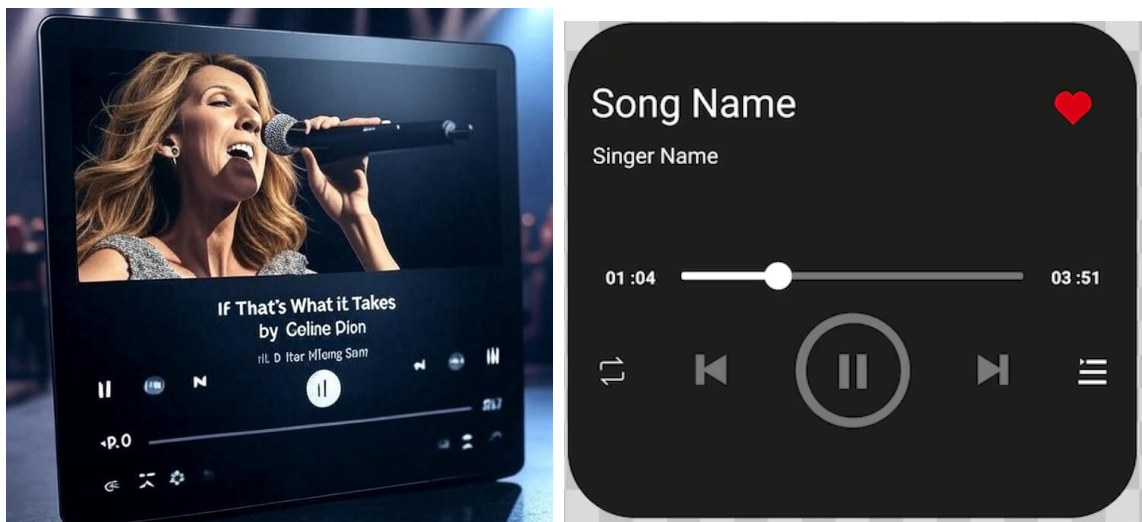


Figura 1. Ejemplos del skin del reproductor de canciones en un servicio de streaming (generado con Grok, Twitter)

UdeATunes aspira a posicionarse como una compañía líder del streaming musical, conformando una comunidad de melómanos en el territorio colombiano. En el largo plazo, se espera consolidar su operación a través de una plataforma en línea que ofrezca sus servicios a nivel mundial.

Cada usuario se registra en la plataforma con un nickname (nombre de usuario único), tipo de membresía (estándar/premium), ciudad, país y una fecha de inscripción en la plataforma. Los usuarios pueden cambiar el tipo de membresía en cualquier momento, aunque esa opción no se implementará en la primera versión del sistema. Los usuarios premium, quienes pagan 19900 pesos mensuales, tienen una serie de beneficios especiales:

- ★ Pueden poseer una lista personalizada de canciones favoritas, con todas las canciones que el usuario ha agregado previamente (hasta 10000 canciones).
- ★ Pueden elegir seguir, de manera continua, la lista de canciones favoritas de otro usuario premium.
- ★ Escucharán todas las canciones en formato de alta calidad 320 kbps, mientras que los usuarios estándar sólo las escucharán a 128 kbps.
- ★ Pueden reproducir todas las canciones de la plataforma sin que les aparezca publicidad.

Dado que los usuarios estándar no pagan a la plataforma, cada dos canciones reproducidas visualizarán en pantalla un mensaje publicitario (de hasta 500 caracteres). Ningún usuario debe visualizar el mismo mensaje de forma consecutiva o con algún patrón de reproducción fijo. En la base de datos de la plataforma pueden existir hasta 50 mensajes publicitarios, sin embargo, no todos tienen la misma prioridad. Los mensajes de categoría C tienen una probabilidad de visualización simple, los de categoría B tienen el doble de prioridad, y los de categoría AAA tienen el triple de prioridad que los de categoría C.

Los artistas son parte fundamental de UdeATunes, porque atraen a las masas. De ellos se almacena su código identificador, edad, país de origen, cantidad de seguidores en la plataforma y posición actual en la lista de tendencias global. Cada artista tiene un catálogo con todos sus álbumes. Cada álbum corresponde hasta a cuatro géneros musicales (Pop, Rock, Jazz, Música Clásica, Electrónica, Hip Hop, Reggae, Blues, Latina), tiene una fecha de lanzamiento, una duración total, un nombre, un código identificador, un sello disquero que patrocina el lanzamiento, la portada del disco (imagen en formato png) y una puntuación del 1 al 10 que le han otorgado los usuarios.

Cada álbum se conforma de un grupo de canciones, y cada canción tiene un nombre, identificador, duración, la ubicación de sus archivos de audio, los créditos de la canción y una cantidad de veces que ha sido reproducida. El identificador de las canciones es un entero de nueve dígitos y se subdivide en: el identificador del artista (5 primeros dígitos), el identificador del álbum (siguientes 2 dígitos) y el identificador de la canción (últimos 2 dígitos). Se asume que cada canción pertenece a un único álbum.

Los créditos de la canción se dividen en 3 categorías: Productores, Músicos y Compositores; cada uno de ellos se representa con sus nombres, apellidos y el código de afiliación a la sociedad de autores y compositores de Colombia (10 caracteres

alfanuméricos). Esta categorización de los créditos es muy importante ya que se usará a futuro para calcular las regalías que cada colaborador recibe cada vez que se reproduce una canción en la cual han participado, según las leyes de derecho de autor.

Los archivos de audio de cada canción se guardan en formato “.ogg”, y se encuentran almacenados en una ubicación externa expresada como una ruta absoluta en notación linux. En consecuencia, toda canción tendrá dos audios almacenados en la misma ubicación. Por ejemplo, para la canción “A tu lado” del álbum “Lugar Secreto”, de la cantante “Claudia Lorelle”, se tendrán los archivos:

```
/users/storage/(sub-ruta X variable)/claudia\ lorelle/audio/a\ tu\ lado_128.ogg  
/users/storage/(sub-ruta X variable)/claudia\ lorelle/audio/a\ tu\ lado_320.ogg  
/users/storage/(sub-ruta X variable)/claudia\ lorelle/image/lugar\ secreto.png
```

*La última ruta corresponde al almacenamiento de la portada del álbum.

Al momento de reproducir una canción se deben imprimir en pantalla la ruta completa a la portada del álbum correspondiente y la ruta completa donde se localiza la canción, incluyendo el nombre y la extensión del archivo de audio.

“Mensaje publicitario” (Si aplica)
Categoría del mensaje (Si aplica)

Cantante:
Álbum:
Ruta a la portada del álbum:

Título de la canción reproducida:
Ruta al archivo de audio:
Duración:

Opciones de reproducción:
1.- Reproducir 2.- Detener
*Todas las opciones que apliquen

Figura 2. Ejemplo de la interfaz a mostrar durante las opciones de reproducción (propio)

Funcionalidades esenciales

Presente un menú para acceder a funcionalidades que permitan:

I - Carga / actualización de datos: Se requiere desarrollar los algoritmos requeridos a fin de leer y actualizar los datos desde/hacia el almacenamiento permanente. Esta funcionalidad no debe aparecer en el menú presentado a los usuarios.

Los programadores deben diseñar el formato de uno o más archivos destinados a contener la información de los usuarios, artistas, álbumes, canciones, listas de reproducción y mensajes publicitarios. Medite adecuadamente este asunto.

El formato de los archivos debe presentarse y explicarse adecuadamente dentro del informe de documentación a entregar. La actualización de dichos datos puede restringirse únicamente a lo que este enunciado solicita.

II - Ingreso a la plataforma

Funcionalidad sencilla que permite iniciar sesión con las credenciales personales y mostrar el menú de funcionalidades disponibles según el tipo de usuario. Los datos requeridos para verificar el inicio de sesión se recuperan del almacenamiento permanente. En este enunciado no se contempla una funcionalidad de registro de usuarios.

III - Reproducción aleatoria

Esta funcionalidad permite reproducir música de manera totalmente aleatoria, de entre todas las canciones de la plataforma. Al reproducir, debe mostrarse en pantalla la ruta del archivo que se está reproduciendo y la ruta de la portada del álbum al que pertenece la canción específica. Adicionalmente, se deben mostrar opciones para: Detener reproducción, e iniciar reproducción. Si el usuario es de categoría estándar la reproducción se acompaña de publicidad, según lo ya especificado.

Para los usuarios premium, además se deben mostrar opciones de interacción para: pasar a la siguiente canción (siguiente), volver a la canción anterior (previa) y repetir la canción actual (indefinidamente hasta que esta opción sea desactivada). Los usuarios estándar no pueden consultar canciones previas, pero los usuarios premium pueden regresar hasta cuatro canciones hacia atrás (en este caso, las canciones en modo repetir sólo se registran una vez). Recuerde validar la lógica implícita entre estas opciones, por ejemplo no se puede detener la reproducción a menos que esté activa, no hay canción previa al inicio, etc.

Nota: A efectos de poder testear esta funcionalidad, incorpore un temporizador de 3 segundos que cambie automáticamente a la siguiente canción, y detenga la reproducción luego de K canciones ($K=5$). Utilice la librería `chronos`.

IV - Mi lista de favoritos: Funcionalidad disponible únicamente para usuarios premium, que permite a cada usuario particularizar en una única lista sus canciones favoritas, hasta 10000 canciones.

Debe desplegar un menú que presente las siguientes opciones:

a. Editar mi lista de favoritos: Esta funcionalidad permite al usuario agregar o quitar canciones a su lista de favoritos. En este caso, la búsqueda la realiza según el código id de canción, se despliegan los datos de la canción en pantalla y se le brinda al usuario la posibilidad de agregarla o eliminarla de su lista de favoritos. En la lista de favoritos no debe haber canciones repetidas.

b. Seguir otra lista de favoritos: Esta funcionalidad permite al usuario seguir de forma continua la lista de favoritos de otro usuario, a partir del nombre de usuario del usuario a seguir. Si el usuario “seguidor” ya tenía una lista de favoritos propia, a esta se adicionará la nueva lista del usuario a seguir.

c. Ejecutar mi lista de favoritos: permite reproducir mi lista de favoritos siguiendo las especificaciones de reproducción establecidas previamente en este documento. El usuario debe seleccionar si quiere que sus favoritos se reproduzcan en el orden original o de forma aleatoria.

En la implementación inicial de esta opción, los usuarios pueden regresar hasta M canciones de su lista de reproducción actual ($M=6$).

V. Medición del consumo de recursos: Al finalizar la ejecución de cada una de las funcionalidades descritas, se deben mostrar en pantalla dos métricas para cuantificar la eficiencia:

- a. La cantidad de iteraciones requeridas (directa e indirectamente) para completar esa funcionalidad específica.
- b. El total de memoria que se está consumiendo en ese momento exacto de la ejecución, por causa de **TODAS** las estructuras de datos y objetos existentes en el sistema. Incluya en este valor las variables locales y parámetros por valor involucrados en la funcionalidad.

Requisitos del desarrollo

El cliente requiere un programa para la gestión del servicio de streaming musical UdeATunes (prototipo 1.0). Este debe permitir representar las estructuras de datos requeridas y ejecutar las operaciones especificadas en este enunciado. La interacción con el usuario se realiza totalmente a través de la consola.

Se recomienda limitar el desarrollo a lo aquí descrito, ya que la dimensión del problema en un escenario totalmente real puede tener una complejidad superior al tiempo estipulado para la entrega. En caso de duda sobre los requerimientos, consulte con el cliente. Es muy importante que tenga presente el criterio de eficiencia para su desarrollo, ya que será parte fundamental de la evaluación.

De acuerdo con lo anterior, entre otras cosas, usted deberá:

1. **[10%]** Contextualice el problema, analícelo y diseñe el diagrama de clases correspondiente a su solución. Refleje adecuadamente las relaciones implícitas en la problemática. Recuerde utilizar la notación UML simplificada impartida en las clases teóricas. A pesar de tener una baja ponderación, la entrega de este diagrama es obligatoria y sin ella no se dará lugar a la sustentación. Considere en su modelo hacer uso adecuado de todos los conceptos estudiados, como: constructores de copia, getters, setters, sobrecarga de métodos y de operadores (*sobrecargar al menos 2*), etc.
2. **[25%]** Previo a la implementación, verifique el cumplimiento del requisito de eficiencia especificado en la sección 3 del apartado “Requisitos de la entrega” de este documento. Seleccione concienzudamente los tipos y estructuras de datos que le permitirán implementar las clases planteadas en su diagrama.
3. **[15%]** Implemente las funcionalidad: “Medición del consumo de recursos”.
4. **[50%]** Presente la implementación de su programa, cuya interacción se centra en un menú que permita acceder de manera independiente a las funcionalidades correspondientes a cada clase. Considere las restricciones implícitas en la lógica del problema. Por ejemplo, no puede haber canciones, álbumes o usuarios repetidos, una lista de favoritos no puede contener dos veces la misma canción.

Formalidades de la entrega

A continuación, se describen los requisitos que se deben cumplir. El incumplimiento de cualquiera de ellos implicaría alguna penalidad o que su nota sea cero.

1. Genere un informe en donde se detalle el desarrollo del proyecto, explique entre otras cosas:
 - a. Análisis del problema y consideraciones para la alternativa de solución propuesta.
 - b. 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.
 - c. Algoritmos implementados debidamente intra-documentados. No exceda la intra-documentación. No use IA para generar la documentación.
 - d. Problemas de desarrollo que afrontó.
 - e. Evolución de la solución y consideraciones para tener en cuenta en la implementación.
2. La solución debe ser implementada en lenguaje C++ y debe basarse en el paradigma de POO.
3. La implementación debe considerar el criterio de eficiencia y el buen diseño tanto de las estructuras de datos como de los módulos implementados. Nota: considere además en este apartado la utilización

de referencias vs la realización de copias innecesarias sobre datos estructurados.

4. La implementación no debe considerar el uso de la STL. Las estructuras de datos usadas deben ser creación propia y deben utilizar memoria dinámica.
5. Se debe crear un repositorio público para cargar todos los archivos relacionados a la solución planteada (informe, código fuente y otros anexos).
6. Una vez cumplida la fecha de entrega no se debe hacer modificación alguna al repositorio.
7. Se deben hacer *commits* de forma regular (al menos uno al día) de tal forma que se evidencie la evolución de la propuesta de solución y su implementación.
8. Se debe adjuntar un enlace de *youtube* a un video que debe incluir lo siguiente:
 - a. Presentación de la solución planteada. Análisis realizado y explicación de la arquitectura del sistema (3 minutos máximo).
 - b. Demostración de funcionamiento del sistema. Explicar cómo funciona: ejemplos demostrativos (3 minutos máximo). Sin evidencias de esta ejecución no habrá sustentación.
 - c. Explicación del código fuente. Justifique la elección de las variables y estructuras de control usados, y destaque la ventaja que ofrecen estos en comparación con otras opciones (5 minutos máximo).
 - d. La duración total del video no debe exceder 11 minutos ni ser inferior a 5 minutos. No debe estar acelerado.
 - e. Asegúrese que el video tenga buen sonido y que se puedan visualizar bien los componentes presentados.

Nota: Debe ser explícita la participación de los integrantes del equipo en la elaboración del video. No se aceptan videos generados por IA.
9. El plazo de entrega se divide en dos momentos:
 - a. El día 17 de Octubre para adjuntar la evidencia del proceso de análisis y diseño de la solución.
 - b. El día 24 de Octubre para adjuntar la evidencia del proceso de implementación.
10. Se deben adjuntar **dos enlaces**: uno al repositorio y otro al video, nada más.
11. Para la evaluación del desafío se realizará una sustentación oral en un horario concertado con el profesor. La asistencia a la sustentación es obligatoria para optar a calificación.