

# Proyecto Final de Curso: Gestor de Lista de Reproducción de Canciones

## Descripción del Proyecto

En este proyecto, el estudiante desarrollará un programa para gestionar una lista de reproducción de canciones. El programa debe utilizar estructuras de datos adecuadas para almacenar y manipular la lista de canciones.

## Las características clave del programa incluirán:

- Agregar canciones a la lista de reproducción.
- Eliminar canciones de la lista de reproducción.
- Cambiar el orden de las canciones en la lista.
- Reproducción aleatoria de canciones.(opcional)
- Ajuste dinámico del tamaño de la lista de reproducción en memoria según se agregan o eliminan canciones.
- Interfaz gráfica (opcional)

## Requisitos del Proyecto

Estructura de Datos: Utilizar una o varias estructuras de datos para almacenar las canciones. Estas deberán ser algunas de las que se implementaron a lo largo del curso. No se admitirá el uso de estructuras provenientes de las clases y/o colecciones de Java, a excepción de aquellas utilizadas exclusivamente para la interfaz gráfica en caso lo requiera.

Funciones Principales:

- `agregar_cancion(cancion)`: Añade una canción a la lista de reproducción.
- `eliminar_cancion(cancion)`: Elimina una canción de la lista de reproducción.
- `cambiar_orden(posicion_actual, nueva_posicion)`: Cambia la canción de la posición actual a una nueva posición.
- `reproduccion_aleatoria()`: Reproduce las canciones en orden aleatorio.
- Manejo Dinámico de Memoria: La lista debe ajustarse dinámicamente en tamaño al agregar o eliminar canciones.
- Diferentes vistas de las canciones, tales como, vistas ordenadas por la popularidad, ordenadas por el año, o por un determinado año, ordenadas por el tiempo de duración. Todos estas vistas pueden ser ascendentes o descendentes según selección del usuario.

## Conjunto de datos de prueba

Para poner a prueba la implementación de su programa/estructuras, se deberá usar el siguiente conjunto de datos : [archive.zip](#)

Este conjunto de datos se extrajo de la plataforma Spotify, que permite a los usuarios acceder a datos musicales proporcionados a través de API. El conjunto de datos recopilado incluye alrededor de 1 millón de pistas entre 2000 y 2023. Además, hay un total de 61,445 artistas únicos y 82 géneros en los datos. El conjunto de datos tiene 20 columnas con la siguiente descripción(15 columnas):

Audio Features	Description
Popularity	Track popularity (0 to 100)
Year	Year released (2000 to 2023)
Danceability	Track suitability for dancing (0.0 to 1.0)
Energy	The perceptual measure of intensity and activity (0.0 to 1.0)
Key	The key, the track is in (-1 to -11)
Loudness	Overall loudness of track in decibels (-60 to 0 dB)
Mode	Modality of the track (Major '1' / Minor '0')
Speechiness	Presence of spoken words in the track
Acousticness	Confidence measure from 0 to 1 of whether the track is acoustic
Instrumentalness	Whether tracks contain vocals. (0.0 to 1.0)
Liveness	Presence of audience in the recording (0.0 – 1.0)
Valence	Musical positiveness (0.0 to 1.0)
Tempo	Tempo of the track in beats per minute (BPM)
Time_signature	Estimated time signature (3 to 7)
Duration_ms	Duration of track in milliseconds

## Entregables del proyecto y Fechas importantes

<p><b>El líder del grupo</b> debe enviar por el aula virtual los siguientes entregables: <b>en un solo archivo .zip</b>:</p> <ul style="list-style-type: none"> <li>• Código fuente (debe ser la última versión que se encuentra en el repositorio), en un único archivo .zip o .rar (*).</li> <li>• Artículo en formato IEEE (dos columnas) en el describan la experiencia, en un archivo .pdf</li> <li>• Diapositivas de la presentación de su proyecto, en un archivo .pdf</li> </ul>	Hasta 23/07 a las 23:00
Presentaciones - exposiciones	22/07 al 26/07 en horario de clases

\* En relación al código, este debe ser subido al repositorio GitHub que se les proporcione a cada grupo, conforme se hagan los avances por cada uno de los miembros lo cual permitirá evidenciar la contribución individual. La versión final deberá ser subida al aula virtual. La versión que se envíe al aula virtual y la que se encuentre en el repositorio no deben de ser diferentes por ningún motivo.

## Evaluación

Los siguientes son los criterios que serán considerados en la evaluación del proyecto:

Criterio	Insuficiente (E – 0% - 10%)	Suficiente (C- 40%)	Bien (B- 70%)	Excelente (A-100%)	Puntuación
<b>Funcionalidad Básica</b> (agregar, eliminar, cambiar orden, reproducción aleatoria)	No cumple con las funciones mínimas requeridas.	Cumple con algunas funciones mínimas.	Cumple con todas las funciones mínimas.	Cumple con todas las funciones y algunas opcionales.	4.0
<b>Estructuras de Datos utilizada</b>	No utiliza estructuras adecuadas.	Utiliza estructuras pero con errores menores.	Utiliza estructuras adecuadas correctamente.	Utiliza estructuras complejas y optimizadas.	2.0
<b>Interfaz de Usuario (opcional)</b>	No implementada.	N/A	N/A	Implementada correctamente.	1.0
<b>Ordenación y Filtrado</b>	No implementado o con errores severos.	Implementado pero con errores menores.	Implementado correctamente.	Implementado correctamente y de manera eficiente.	2.0

<b>Pruebas con Datos Proporcionados</b>	No realiza pruebas adecuadas.	Realiza algunas pruebas pero incompletas.	Realiza pruebas adecuadas y correctas.	Realiza pruebas exhaustivas y bien documentadas.	2.0
<b>Eficiencia y Reuso del Código</b>	Código ineficiente y desorganizado.	Código con algunas ineficiencias.	Código eficiente y bien organizado.	Código muy eficiente y excelentemente organizado.	2.0
<b>Artículo</b>	Ausente o muy pobre.	Incompleta o poco clara.	Clara y completa.	Muy clara, completa y profesional.	2.0
<b>Presentación del Proyecto</b>	Presentación muy pobre o ausente.	Presentación básica y con errores.	Presentación adecuada y clara.	Presentación muy clara, organizada y profesional.	2.0
<b>Contribución Individual</b>	No contribuyó o contribución muy baja.	Contribución moderada, participación mínima.	Buena contribución, participación adecuada.	Excelente contribución, alta participación.	3.0
- Evaluado por profesor (2.0)					
- Co-evaluación (1.0)					

**Puntuación Total: 20**

#### **Notas:**

1. La contribución individual se evaluará mediante co-evaluaciones de los miembros del equipo, así como la observación del profesor durante el desarrollo del proyecto.
2. Los criterios de funcionalidad básica deben estar claramente implementados y funcionar sin errores significativos para obtener una buena puntuación en esta categoría.
3. La interfaz de usuario es opcional, pero se valorará positivamente si está bien implementada.
4. La selección de las estructuras de datos, será calificada teniendo en cuenta la justificación del porqué de su elección.
5. Se espera que el artículo tenga una documentación clara y detallada, que incluya descripciones de las funciones y estructuras de datos utilizadas.
6. Se deben realizar pruebas exhaustivas utilizando el conjunto de datos proporcionado para garantizar la robustez del programa