

Práctica: Implementación de una Lista de Reproducción de Música con Listas Enlazadas

Objetivo

El objetivo de esta práctica es desarrollar una aplicación en consola que simule una **lista de reproducción de música**, utilizando **estructuras de datos enlazadas**. Los estudiantes deberán analizar y justificar qué tipo de estructura es la más adecuada para implementar cada funcionalidad.

Se espera que el programa permita agregar canciones, avanzar y retroceder entre ellas, eliminarlas, reproducir en modo aleatorio, adelantar la reproducción y generar *subplaylists*.

Se otorgarán **puntos adicionales** a los estudiantes que usen buenas prácticas de programación y diseño y que implementen una interfaz más amigable y organizada.

Requisitos de la Implementación

1. Estructura de la Lista de Reproducción

- Implementar una **estructura de datos enlazada** para gestionar la lista de canciones.
 - Cada canción debe tener:
 - **Título**
 - **Artista**
 - **Duración** (entre **10 y 15 segundos**).
 - El estudiante debe analizar **qué tipo de estructura** es la más adecuada:
 - **Lista enlazada simple**
 - **Lista doblemente enlazada**
 - **Lista enlazada circular**
 - **Pila implementada con listas enlazadas**
 - **Cola implementada con listas enlazadas**
-

2. Funcionalidades Obligatorias

- ✓ **Agregar una canción a la playlist**

- Permitir al usuario ingresar una canción con su **título, artista y duración**.
- Agregar la canción a la lista.
- No se permiten canciones con el **mismo título** repetido.

✓ Avanzar a la siguiente canción

- Si la lista no está vacía, la reproducción avanza a la siguiente canción.
- Si se llega al final de la lista, debe **volver al inicio** (si se eligió una estructura circular).

✓ Retroceder a la canción anterior

- Si la lista no está vacía, la reproducción retrocede a la canción anterior.
- Si se está en la primera canción, debe **regresar a la última**.

✓ Eliminar una canción

- Se debe permitir **eliminar** una canción por su título.
- Si la canción eliminada estaba en reproducción, se debe reproducir la siguiente automáticamente.
- Si solo queda una canción y se elimina, la lista debe quedar vacía.

✓ Mostrar la canción en reproducción

- Mostrar en pantalla el **título, artista y duración** de la canción que se está reproduciendo.

✓ Mostrar toda la playlist

- Listar **todas** las canciones de la playlist en el orden en que se encuentran en la lista.

✓ Reproducir en orden aleatorio (shuffle mode)

- Se debe permitir activar un modo aleatorio donde las canciones se reproduzcan en orden aleatorio **sin repetir hasta que todas se reproduzcan**.

✓ Adelantar la canción un % específico

- El usuario puede ingresar un porcentaje (**10%, 20%, 50%**, etc.) para adelantar la canción.
- Si el tiempo adelantado supera la duración de la canción, debe pasar a la siguiente automáticamente.

✓ Simulación del tiempo de reproducción

- Cada canción debe reproducirse por su duración **real** (entre **5 y 10 segundos**).
- Mientras se reproduce, el sistema debe mostrar el **tiempo transcurrido**.
- Cuando termine la canción, debe **pasar automáticamente a la siguiente**.

✓ Generar una subplaylist


- Permitir al usuario seleccionar un **subconjunto de canciones** y generar una **nueva playlist** con ellas.
- La subplaylist debe ser independiente y funcionar igual que la playlist original.
- Luego de ser creada, el sistema debe proponer la opción de reproducir y trabajar con la subplaylist creada. Si el usuario elige esta opción, el programa funcionará exactamente igual que como el original pero ahora sobre la subplaylist creada.

Evaluación y Bonificación Extra

Criterio	Puntaje
Implementación funcional de todas las características	40%
Sustentación	60%
Buenas prácticas de programación:	+10 %
Estética e interactividad de la consola	+10 %

Bonificación Extra (20 pts)

Se otorgarán puntos adicionales si la solución está bien estructurada y la interfaz de consola es **intuitiva, interactiva y bien organizada**. Algunas ideas para mejorar la estética:

- **Menús bien estructurados con opciones numeradas.**
- **Mensajes claros y visualmente diferenciados** (por ejemplo, colores o símbolos ASCII ).
- **Animaciones sencillas** (por ejemplo, mostrar que la canción se está reproduciendo con una barra de progreso).
- **Indicadores visuales de la canción actual en la playlist.**

Ejemplo de Ejecución Esperada

Menú Principal

 Bienvenido a tu Playlist Interactiva 

- 1) Agregar canción
- 2) Avanzar a la siguiente canción
- 3) Retroceder a la canción anterior
- 4) Eliminar una canción
- 5) Mostrar canción en reproducción
- 6) Mostrar toda la playlist
- 7) Activar modo aleatorio
- 8) Adelantar una canción

9 Generar una subplaylist

10 Salir

Seleccione una opción: _

+ Agregar una Canción

Ingrese el título: Bohemian Rhapsody

Ingrese el artista: Queen

Ingrese la duración (10-15 seg): 12

✓ Canción agregada exitosamente.

▶ Mostrar la Canción en Reproducción

🎵 Ahora reproduciendo:

🎧 Bohemian Rhapsody - Queen (12s)

▶▶ Adelantar un 50% de la Canción

Ingrese el porcentaje de adelanto: 50

🕒 Adelantando 6s...

✂ Activar Modo Aleatorio

✂ Modo aleatorio activado.

🎵 Nueva reproducción: Hotel California - Eagles (11s)

📄 Generar una Subplaylist

Ingrese los títulos de las canciones a incluir en la subplaylist (separados por comas):

> Bohemian Rhapsody, Hotel California

✓ Subplaylist creada con 2 canciones.

1 Quiere reproducir esta nueva subplaylist? ____

✗ Eliminar una Canción

Ingrese el título de la canción a eliminar: Bohemian Rhapsody

✓ Canción eliminada exitosamente.

🕒 Simulación de Tiempo de Reproducción

▶ Reproduciendo: Smells Like Teen Spirit - Nirvana (14s)

🕒 1s... 2s... 3s... (progresivamente hasta 14s) / Barra de progreso

🔄 Pasando a la siguiente canción...

Entrega y Sustentación

- **Grupo 61:** 24 de abril
 - **Grupo 62:** 25 de abril
 - **Grupo 63:** 24 de abril
 - **Grupo 64:** 25 de abril
-