Proyecto Programado
Estructuras Dinámicas Lineales

Objetivos:

- Comprender e implementar estructuras de datos dinámicas como listas enlazadas simples, dobles y circulares.
- Aplicar el uso de punteros y nodos personalizados (struct) para organizar y gestionar información compleja.
- Diseñar soluciones algorítmicas para búsquedas, ordenamientos y reportes en un entorno simulado y útil.
- Estimular el análisis crítico, la validación de datos, la eficiencia de almacenamiento y la representación adecuada de relaciones entre datos.
- Desarrollar un sistema funcional en C++, con documentación interna y externa.

Enunciado:

Imaginá que una empresa emergente de tecnología ha contratado a tu equipo como desarrolladores de software. Su visión es clara: crear la plataforma definitiva para gestionar colecciones musicales personales, con un enfoque en artistas independientes, sellos discográficos emergentes y coleccionistas apasionados por la música.

MusiGest será una aplicación que permita organizar una gran biblioteca musical de forma estructurada, dinámica y consultable. El sistema debe ser capaz de registrar artistas, canciones, álbumes, géneros musicales, playlists, y sellos discográficos, y permitir múltiples operaciones sobre ellos.

Para cumplir este objetivo, deberás diseñar seis estructuras de datos interconectadas utilizando exclusivamente listas dinámicas construidas desde cero, sin utilizar librerías estándar como std::list ni std::vector.

Información que debe registrarse en el programa:

Estructura	*Tipo de Lista*	*Atributos*	*Relaciones Clave*
Canciones	Lista simple con inserción al inicio	ID, título, duración, año, ID de álbum, ID de artista	Pertenece a un álbum, artista y género. Puede estar en múltiples playlists.
Artistas	Lista doble ordenada alfabéticamente	ID, nombre artístico, nombre real, país, sello discográfico	Posee múltiples álbumes y canciones

Álbumes	Lista simple con inserción al final	ID, título, año, número de canciones	Pertenece a un artista. Contiene una sublista de canciones.
Géneros Musicales	Lista circular con inserción al final	ID, nombre, descripción	Contiene una sublista de canciones
Playlists	Lista simple con inserción al inicio	ID, nombre, creador, fecha	Contiene sublistas de canciones
Sellos Discográficos	Lista doble y circular con inserción final	ID, nombre, país, año de fundación	Tiene múltiples artistas asociados

Nota: No se permite la duplicación de código e información, se debe hacer un correcto uso de los enlaces hacia la información respectiva.

Inserción y actualización

- El programa debe permitir agregar, modificar y eliminar los datos de todas las listas y sublistas.
- Validar la unicidad de IDs y nombres clave.
- Verificar que los datos sean coherentes (ej. Duración no negativa, años válidos).
- Debe validar que no se ingresen datos repetidos.
- Asociar correctamente las entidades.
- Al iniciar el sistema, deben cargarse automáticamente al menos 10 registros de cada entidad mediante datos quemados en código (no lectura de archivos).

Consultas

- ¿Cuál es el género con más canciones registradas?
- ¿Qué artista tiene más álbumes publicados?
- ¿Cuál es la canción más larga de la base de datos?
- ¿Qué playlist contiene más canciones?
- ¿Cuántos álbumes hay por cada artista?
- ¿Qué sello discográfico tiene más artistas firmados?
- ¿Cuáles son las canciones publicadas en un año determinado?
- ¿Cuáles álbumes superan cierto número de canciones?

Reportes

- Imprimir todas las listas principales.
- Reporte detallado de cada artista con sus álbumes, canciones y sello.
- Lista de playlists con sus canciones correspondientes.
- Álbumes con canciones ordenadas por duración.
- Sellos discográficos con sus artistas firmados.

Interfaz de Usuario

- Menú general de navegación por secciones: Mantenimiento, Consultas, Reportes, Salida.
- Submenús para cada entidad con opciones CRUD.
- Diseño limpio, navegación intuitiva y respuestas amigables en cada operación.
- Usar funciones organizadas para cada tipo de operación.
- Debe mostrar mensajes claros al usuario final de todo lo que pasa dentro de su programa (ejemplo: “Dato ya insertado”).

Documentación Externa

- ✓ Debe realizar un diagrama donde incluya todas las listas y sublistas para poder responder a todas las consultas y reportes.
- ✓ Portada.
- ✓ Descripción del problema y justificación.
- ✓ Solución propuesta (colocar únicamente la última solución, cuáles estructuras se utilizaron, diagrama de clases o estructuras, lógica de cómo se trabajó para realizar el programa, minuta de trabajo por cada estudiante).
- ✓ Análisis de resultados (resultados finales, indicando que partes están completas, cuáles defectuosas y cuáles no se realizaron).
- ✓ Conclusiones y recomendaciones con respecto al proyecto.
- ✓ Bibliografía y herramientas utilizadas.

Documentación Interna

- ✓ Fecha de inicio y última modificación.
- ✓ Nombre de los integrantes del grupo de trabajo.
- ✓ Descripción para cada estructura y su uso en el programa.
- ✓ Descripción para cada función.

Aspectos Administrativos

- ✓ El proyecto debe programarse en el lenguaje C++ y debe entregarse el programa fuente (cpp).
- ✓ El desarrollo del trabajo se puede realizar en grupos de 3 personas como máximo.
- ✓ La fecha de entrega es 8 mayo de 2025, antes de las 11:45 p.m.
- ✓ Se calificará con citas de revisión para la defensa de la tarea de ser necesario.
- ✓ En caso de plagio, la calificación será de cero para todos los implicados.
- ✓ En caso de encontrarse algún virus o si se encuentra mal identificado se rebajarán puntos por descuido del estudiante.
- ✓ Si el proyecto no abre o no se ejecuta correctamente no se calificará la parte programada.
- ✓ Se recomienda que comience a trabajar desde hoy.

Evaluación

Código (70%)

Rubro de evaluación	***Puntaje***
Inserción, modificación, eliminación en listas y sublistas	6
Validaciones e integridad de datos	2
Relaciones entre estructuras	4
Consultas y reportes solicitados	8
Carga automática de datos por defecto	2
Interfaz y navegación por menús	5
Documentación interna (comentarios y estructura)	8
Modularidad y legibilidad del código	4
TOTAL	39
	10%

Documentación (30%)

Rubro de evaluación	***Puntaje***
Portada y presentación	1
Justificación del problema	2
Solución con diagrama de estructuras	7
Análisis de resultados	2
Conclusiones y recomendaciones	2
Formato, ortografía, redacción general	-5
TOTAL	14
	5%