

## Primer proyecto programado Proyecto para la utilización de estructuras dinámicas lineales.

### **Objetivos**

Desarrollar un programa en el lenguaje c++, utilizando las ventajas de OO.

Implementar estructuras de datos dinámicas: listas. No puede hacer uso de la clase list, ni arreglos, ni vectores....

Implementar nuevos tipos de datos: punteros.

### **Definición**

Realizar un programa para registrar datos de grupos musicales, después del registro se consultará la información para conocer los reportes

#### **Información que debe registrarse en el programa:**

Grupo musical (Lista simple con inserción al inicio)	Nombre, año de fundación, Se relaciona, con su único director, sus integrantes y eventos en los que ha participado.
Personas (Lista doble ordenado por cédula).	Nombre, cédula, edad ... (puede pertenecer a varios grupos). Se relaciona con la sublista de horario disponible y con los instrumentos que toca.
Sublista de horario disponible (lista simple con inserción al final)	Día, hora inicio y hora final
Instrumentos (lista circular con inserción al final)	Id, nombre, tipo.
Eventos (Lista simple con inserción al final)	Nombre, lugar, duración, día y hora. Se relaciona con las características (sublista de instrumentos solicitados)
Sublista de instrumentos solicitados (lista simple con inserción al inicio)	Cantidad y enlace al instrumento.
Historial de eventos (lista doblemente enlazada y circular)	Enlace al evento, calificación, enlace al grupo, y sublista de quienes participaron.

Queda prohibido la duplicación código y de información, debe trabajar muy bien los enlaces hacia la información respectiva.

### **Inserción y actualización de información**

-El programa debe permitir insertar los datos de todas las listas y sublistas. Excepto en el historial y la participación de los grupos en eventos).

- Se podrá modificar un dato de todas las listas y sublistas. En los eventos se podrá cambiar día y la hora.
- Se podrá borrar nodos de todas las listas y sublistas, excepto el historial.
- Debe validar que no se ingresen datos repetidos.
- Validar los tiempos (día y horas). Por ejemplo: las 26 horas, no existe..
- Debe traer datos por defecto dentro del programa, mínimo 10 de cada uno (en listas y sublistas). El historial lo crea el programa.

## ***Consultas***

1. Cual persona tiene más particiones en eventos.
2. Cuales eventos no se pueden realizar porque no se logró cumplir con todas las características.
3. Cuales eventos no se pueden realizar porque el día y hora no esta disponible entre las personas.
4. Cuál es el instrumento más y menos popular (más tocado).
5. Cuál persona está en más grupos musicales.
6. Cual persona toca más instrumentos.
7. Cuál director promedia mejor calificación, con respecto al historial de eventos.
8. Cuál grupo tiene más participaciones en eventos. En caso de empate imprimir todos los empatados.

## ***Reportes***

1. Imprimir todas las listas: eventos, instrumentos, personas y grupos musicales.
2. Imprimir la lista de personas y por cada persona su información, los instrumentos que toca y el horario disponible.
3. Imprimir los grupos musicales, y por cada grupo indicar quien su director, sus integrantes y los eventos en que participado.
4. A partir los eventos actuales registrados obtener la ejecución de ellos (si corresponde insertar en el historial e insertar en el grupo la participación), luego imprimir el historial y por cada uno indicar nombre del evento, cual grupo lo cubrió y cuales personas participaron en dicho evento.

## ***Interfaz de usuario***

Debe contar con un menú principal para las operaciones de mantenimiento de datos, consultas y reportes respectivos.

Debe mostrar mensajes claros al usuario final de todo lo que está pasando en su programa, por ejemplo: “Dato ya insertado en la lista”...

### ***Clases o estructuras a usar***

Debe realizar un diagrama donde incluya todas las listas y sublistas, con los atributos necesarios para poder responder a todas las consultas y reportes. Entregar en la primera semana.

**Nota:** Tomar en cuenta todas las aclaraciones que se dieron en día de entrega y discusión de este proyecto.

### ***Documentación Externa***

Portada.

Descripción del problema.

Solución del problema (la última solución, indique cuales son las estructuras utilizadas, diagrama de las clases o estructuras con sus listas y sublistas, lógica de cómo se trabajó para realizar el programa, como fue el recorrido utilizado para resolver cada una de las consultas y reportes).

Análisis de Resultados (Resultados finales, indique que partes están completas, cuales defectuosos, y cuales no se realizaron y el porqué).

Conclusiones y recomendaciones con respecto al proyecto.

Recomendaciones al profesor (con respecto al proyecto, lenguaje, tiempo para la realización, la explicación en clase, horas de consulta etc.)

### ***Documentación Interna***

Fecha de inicio y Fecha última modificación.

Descripción para cada estructura (clase) y su uso en el programa,

Describir cada función e instrucciones dentro de estas.

#### **Aspectos Administrativos**

- La tarea debe programarse en lenguaje c++ . Debe enviar los programas fuentes.
- El desarrollo de este trabajo se puede realizar en grupos de tres como máximo.
- Entrega de la tarea: \_1 de abril de 2023, antes de las 11:55 p.m.
- Se calificará con citas de revisión para la defensa de la tarea de ser necesario.
- Si se encuentra copia la calificación será de cero para todos los implicados.
- Si existe virus o si se encuentra mal identificando se rebajarán puntos por descuido del estudiante. Si no habré el proyecto no se calificará la parte programada.
- Se recomienda que se comience a trabajar desde hoy.

## Evaluación

Rubro de evaluación	Puntos
El programa permite insertar en las listas y sublistas.	6
El programa valida que no se ingresen datos repetidos, que las días y horas sean correctas.	2
El sistema realiza una adecuada asociación de las listas y sub-listas según el enunciado	4
Modificar los datos en todas las listas y sublistas.	3
Borrar en todas las listas y sublistas	6
El sistema cuenta con datos previamente insertados.	2
El sistema genera correctamente las consultas solicitadas:	8
El sistema genera correctamente los reportes solicitados	4
<b>Total</b>	
	<b>15%</b>

## Documentación

Rubro de evaluación	Puntos
Portada	1
Descripción del problema.	2
Solución del problema (la última solución, indique cuales son las estructuras utilizadas, diagrame las clases o estructuras con sus listas y sublistas, lógica de cómo se trabajó para realizar el programa, <b>como fue el recorrido utilizado</b> para resolver cada una de las consultas y reportes).	7
Análisis de Resultados (Resultados finales, indique que partes están completas, cuales defectuosos, y cuales no se realizaron y el porqué).	2
Conclusiones y Recomendaciones	2
Formato, redacción y documentación interna	-5
<b>Total</b>	<b>14</b>
	<b>5%</b>