

Objetivos generales

- Comprender y aplicar los principios de las estructuras básicas de datos.
- Familiarizarse con el lenguaje C/C++
- Aplicar los conocimientos obtenidos en el laboratorio

Objetivos específicos

- Que el estudiante pueda aplicar las listas enlazadas, doblemente enlazadas y circulares.
- Utilizar pilas y colas para la resolución de problemas (FIFO Y FILO)
- Utilizar punteros y memoria.

Descripción de la actividad

La empresa Spotify lo ha contratado para desarrollar un reproductor de audio en terminal para sistemas Linux utilizando lenguaje C/C++.

Para poder probar el funcionamiento correcto de la misma, le enviarán archivos para que pueda reproducir sus canciones el formato será:

Así como un archivo de entrada para poder cargar más fácil y rápido las canciones y que sea universal para todos los reproductores; Este será con formato XML.

Funcionamiento

La aplicación será desde terminal por lo que deberá contar con un menú sencillo y numérico de entrada. Contará con las siguientes opciones:

1. Operaciones de canciones
 - 1.1. Insertar
 - 1.2. Eliminar
 - 1.3. Buscar nombre
 - 1.4. Listar canciones
2. Operaciones de Playlist
 - 2.1. Crear
 - 2.2. Eliminar
 - 2.3. Actualizar
 - 2.4. Listar
 - 2.5. Agregar canciones
 - 2.6. Eliminar canciones
3. Reproducción
 - 3.1. Normal
 - 3.2. Repetir
 - 3.3. Siguiente
 - 3.4. Anterior
 - 3.5. Lista de reproducción
 - 3.6. Agregar canción a la lista de reproducción
4. Carga masiva
5. Salir

Opciones de canciones

Esta opción será el equivalente a la “store” de spotify con un catálogo variado de canciones, estas canciones podrán ser agregadas de modo manual y por archivo de texto.

Insertar

Esta opción permitirá el ingreso de canciones al sistema, si en alguno de los campos es nulo o vacío deberá volver a solicitarlo. Los campos son:

- Nombre
- Path

Eliminar

Permite eliminar una canción por medio del nombre o el índice. Por lo cual deberá preguntar cuál de las dos opciones desea y dependiendo de eso deberá aceptar un int o un nombre. Deberá mostrar el nombre, path y el índice de la canción y después deberá pedir confirmación antes de proceder con su eliminación.

Los campos son:

- Nombre | id

Listar canciones

Deberá mostrar todas las canciones guardadas en el sistema. Con la siguiente información:

- id (pos en la que está guardado)
- Nombre
- path

Buscar

Pedirá el nombre de la canción y lo buscará dentro de la lista.

Mostrará los siguientes datos:

- Nombre
- Path
- Índice

Pedirá:

- Nombre

Opciones de Playlist

Se podrá crear diferentes playlist en las cuales se almacenarán las canciones que el usuario desee. Las playlist no tendrán límite de canciones y su orden será conforme se introduzcan las canciones.

Crear

En esta opción se podrán crear varias playlist, almacenando información necesaria para su reproducción.

Entre los cuales están:

- nombre
- descripción
- Listado de canciones.

Para el listado de canciones se puede dejar opcional el introducir las canciones en ese momento. Los otros campos son obligatorios.

Listar

Esta opción mostrará todas las playlist en el sistema, el usuario tendrá la opción de listar las canciones de una playlist específica seleccionando el índice de la canción (estos índices son la posición de la canción en la lista). El formato en pantalla será el siguiente: índice. nombre - descripción.

Ej. 1. MyPlaylist - Esta es una playlist de ejemplo.

Ej. 45. OtraPlaylist - Otra playlist de ejemplo.

Eliminar

De esta forma podrá eliminar la playlist con todas sus canciones. Los campos necesarios son:

- id

Actualizar

En esta únicamente se podrán modificar el nombre de la playlist y su descripción. Se indicará el índice que se quiere modificar y se le enviará el nombre y la descripción. (Si alguno de los datos está vacío se debe quedar con el dato anterior).

Agregar canción

Se imprimirá en pantalla la lista principal de canciones (Store) y seleccionará el índice de la canción que desea agregar. Podrá agregar varias canciones en este modo. Campo necesario: id/pos

Eliminar canción

Se listará la lista de canciones en la playlist y por medio de un índice seleccionará la canción a eliminar de la lista. Podrá eliminar varias canciones en este modo. Campo necesario: id/pos

Reproducir

Las canciones se deben reproducir desde terminal. Para esto se deberá elegir una playlist la cual contendrá todas las canciones a reproducir.

Normal

Reproduce las canciones en el orden que se encuentran en la playlist y al terminar la misma se detiene. (Nota: Para esta reproducción se **debe** utilizar una **lista doblemente enlazada**)

Repetir

Reproduce las canciones en el orden que se encuentran en la playlist y al terminar vuelve a iniciar desde la primer canción (Nota: Para esta reproducción se **debe** usar una **lista circular o una doblemente circular**)

Siguiente

Reproduce la canción que sigue en la lista.

Anterior

Reproduce la canción anterior en la lista.

Lista de reproducción

Te muestra las canciones qué se reproducirá después de la canción actual.

Agregar canción a la lista de reproducción

Listará la lista de canciones (store) y permitirá agregar una canción después de la canción que se esté reproduciendo en ese momento. Para esto deberá implementar una **Pila** como “lista prioritaria” por lo cual antes de seguir con la lista de reproducción deberá confirmar que esta “lista prioritaria” esté vacía.

(Nota: Para esta reproducción se **debe** utilizar una **Pila** → **FILO**)

Carga Masiva

Será un archivo de entrada con el cual se podrá ingresar las canciones, playlist y otros movimientos, servirá como un backup para recuperar la información. El formato de entrada será XML, **No se calificará errores de gramática** sin embargo con esto se ingresará la información.

El formato será el siguiente:

Insertar / Agregar

Esta etiqueta permite el ingresar/crear múltiples canciones y listas, la sintaxis es:

```
<Insertar>
...
...
</Insertar>
```

Insertar canción

Nota: La “pos” es opcional. (Si se ingresa pos debe comprobar qué la lista

```
<Insertar>
...
<cancion>
  <Nombre> nombreCancion </Nombre>
  <path> pathCancion </Path>
  <Pos> posicionEnElArreglo <Pos>
</cancion>
<cancion>
  <Nombre> nombreCancion </Nombre>
  <path> pathCancion </Path>
</cancion>
</Insertar>
```

Insertar lista sin canciones

```
<Insertar>
  <Lista>
    <Nombre> nombreCancion </Nombre>
    <Descripcion> pathCancion </Descripcion>
  </Lista>
</Insertar>
```

Insertar lista con canciones

Las canciones deben estar ingresadas antes de poder ser ingresadas a una lista; Por eso las pos es la posición en la “Store”.

```
<Insertar>
  <Lista>
    <Nombre> nombreCancion </Nombre>
    <Description> pathCancion </Description>
    <Canciones>
      <pos> n </pos>
      <pos> n + 2 </pos>
      <pos> n + m </pos>
    </Canciones>
  </Lista>
</Insertar>
```

Eliminar

Eliminar cancion por id

```
<Eliminar>
  ...
  <cancion>
    <id> </id>
  </cancion>
  ...
</Eliminar>
```

Eliminar cancion por Nombre

```
<Eliminar>
  ...
  <cancion>
    <Nombre> </Nombre>
  </cancion>
  ...
</Eliminar>
```

Eliminar lista por Nombre:

```
<Eliminar>
  ...
  <Lista>
    <Nombre> </Nombre>
  </Lista>
  ...
</Eliminar>
```

Eliminar lista por id

```
<Eliminar>
  ...
  <Lista>
    <id> </id>
  </Lista>
  ...
</Eliminar>
```


Eliminar canción de la lista por id

```
<Eliminar>
  <Lista>
    <id> posList </id>
    <canciones>
      <pos> n </pos>
      <pos> n + m </pos>
      ...
      ...
    </canciones>
  </Lista>
</Eliminar>
```

Notas:

El archivo de entrada **NO TENDRÁ ERRORES LÉXICOS NI SINTÁCTICOS** sin embargo si puede contar con errores el listado de canciones tiene 12 y se pide insertar la canción en la posición 25 o se desea guardar en una lista la canción en la posición 52;

Todas las estructuras de datos utilizadas (Listas enlazadas, dobles, circulares, pilas, colas) deberán ser programadas por uds, no se pueden usar librerías para estas.

Importante

- Usar lenguaje de programación C/C++
- Copias obtendrán nota de cero y se notificará a coordinación.
- Si se va a utilizar código de internet, entender la funcionalidad para que se tome como válido.

Entrega

La fecha de entrega es el día martes 11 de Abril a las 14:00. Los componentes a entregar en repositorio de git son:

- Código fuente (.c / .cpp)
- Documento con el análisis para la aplicación y manual técnico.
- Manual de usuario.

Calificación

Pendiente de definir.