CUCEI

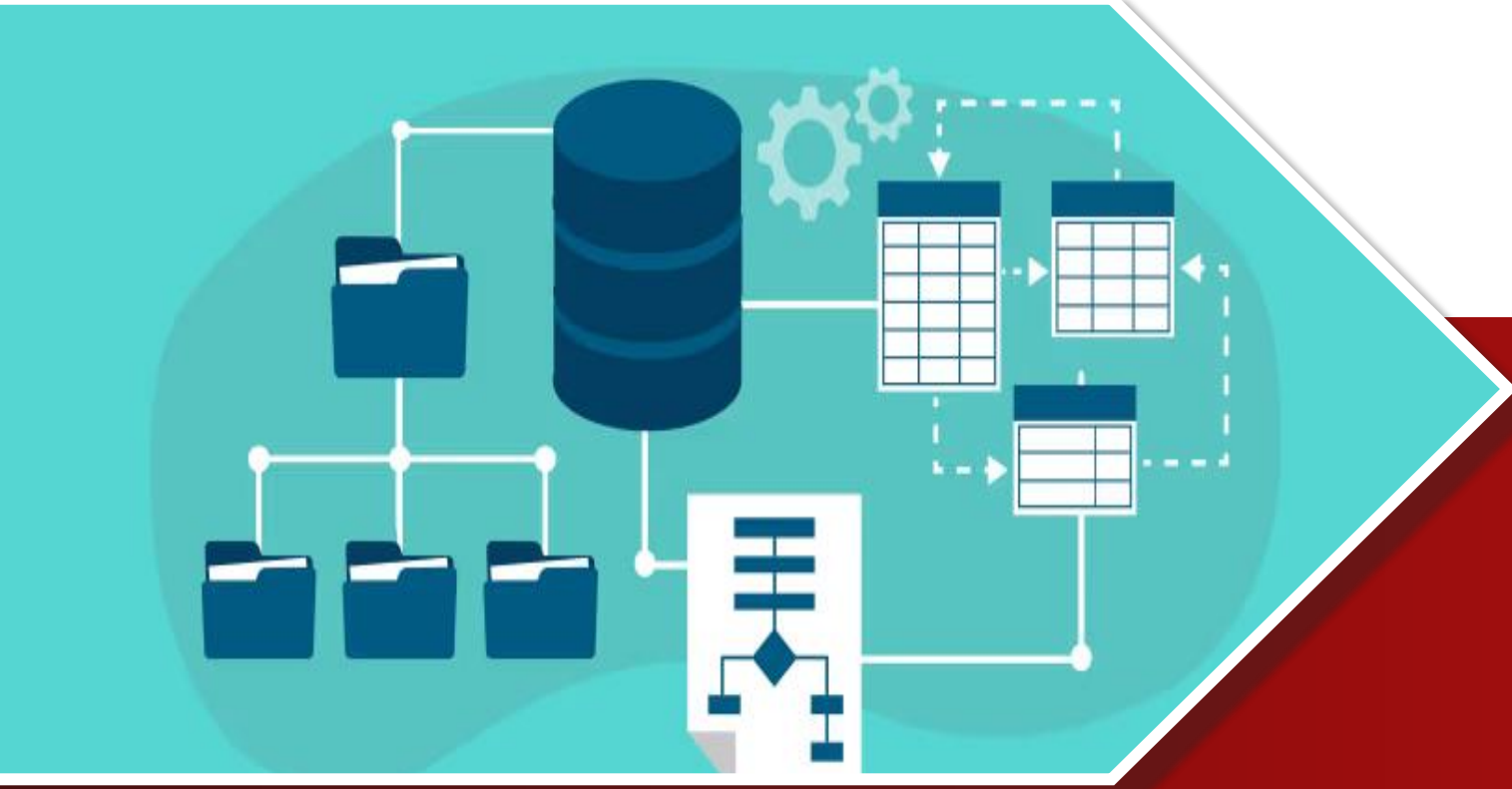
Actividad de Aprendizaje 03

La Lista:
Implementación Estática

Centro Universitario de Ciencias Exactas e Ingenierías

Materia: Estructuras de Datos

Clave: V0731 ***Sección***: D02



Alumno: Mariscal Rodríguez Omar Jesús

Código: 220858478

Profesor: Gutiérrez Hernández Alfredo

Fecha: 31 de Agosto de 2025

UNIVERSIDAD DE GUADALAJARA

PIENSA Y TRABAJA

Contenido

Test de Autoevaluación

Autoevaluación			
Concepto	***Sí***	***No***	***Acumulado***
Bajé el trabajo de internet o alguien me lo pasó (aunque sea de forma parcial)	***-100 pts***	***0 pts***	***0***
Incluí el código fuente ***en formato de texto (sólo si funciona cumpliendo todos los requerimientos)***	***+25 pts***	***0 pts***	***25***
Incluí las ***impresiones de pantalla (sólo si funciona cumpliendo todos los requerimientos)***	***+25 pts***	***0 pts***	***25***
Incluí una ***portada*** que identifica mi trabajo (nombre, código, materia, fecha, título)	***+25 pts***	***0 pts***	***25***
Incluí una ***descripción y conclusiones*** de mi trabajo	***+25 pts***	***0 pts***	***25***
		Suma:	***100***

Introducción y Abordaje del Problema

Con esta tercera actividad y tercera semana, empezamos con las actividades relacionadas a las estructuras de datos, y esta actividad fue la base de todo: la Lista, primeramente, desde una implementación estática; programando desde un modelo que nos proporcionaba cómo se supone que debía funcionar la lista, dictando donde retornar posiciones inválidas y donde dar errores.

De igual manera que con la actividad anterior, dividimos el abordaje del problema en la planeación y la programación

1.- Fase de Planeación

Dadas las indicaciones de la actividad, se decía que la lista de los éxitos que el programa debía guardar mediante la lista estática se mostraba todo el tiempo en pantalla, por lo que las modificaciones se reflejarían en todo momento, por lo que, en base a esta idea central, fui planeando como se vería todo el programa.

El "menú principal", sería la misma lista de los éxitos o canciones, enlistadas con sus características, en adición al índice que toman en la lista. Después, para no poner opciones típicas y hacer un poco más dinámico que simplemente una enumeración que alberguen las opciones; planee hacer algo estilo de "comandos", donde, por ejemplo, eliminar una canción podría ser un comando como C7, donde C es el tipo de comando (eliminar) y 7 el índice (o posición en la lista), de esta manera, también la experiencia de usuario mejora al no redirigirlo a otro submenú donde se le pregunte el número de la canción en cuestión.

La lista, también por practicidad, la visualicé como template, pero aproveché a usar ciertas conveniencias a doc al problema, es decir, utilicé métodos que sólo se usarían con la clase de la canción, aunque la flexibilidad debería ser suficiente como para solo tener que eliminar pocas y selectas cosas para hacerla flexible a cualquier tipo de dato.

Al planear el menú principal como un listado en formato de tabla, imaginé el resto de las pantallas en formato de tabla, por lo que sabría que utilizaría iomanip y https://cplusplus.com/reference/iomanip/ y sstream https://cplusplus.com/reference/sstream/?kw=sstream para facilitar mucho la creación estilo tablas.

Aprovechando la librería sstream, varias de los string's no los imprimiría directamente, sino que las dejaría como ostringstream y las tendría de fácil acceso para utilizar distintos métodos privados dentro de la clase Menú.

Un apartado más, en los programas anteriores tenía una clase abstracta llamada utilities, en este caso, decidí incluir los métodos de input de datos en el propio menú, para encapsular toda la lógica y utilidades de interacción con el usuario con un solo lugar, en caso de que tuviera varios menú's, como un menú de usuarios en una aplicación de streaming de música por ejemplo, sí sería útil tener la clase abstracta para reciclar código, pero al solo ser una única clase con interacción con el usuario, sería la única.

Aprovechando que dentro de la canción debía de haber un interprete y un autor, diseñé una clase sencilla de nombre para acompañarla en una composición.

Y con todo esto fue abordado la situación planteada.

Programación

Para la programación no hubo muchas más dificultades, lo que sí, fue que procuré utilizar mi propia lógica en la implementación de la lista, a pesar de tener apuntes de como se implementó en clase, esto me ayudó mucho para entender bien lo que pasa detrás de la lista de una manera sólida, fue también muy interesante el programar la clase del menú, al planear un menú mediante "comandos", tuve que ver como dividir la información de un string en dos, la parte de carácter y la parte de números, señalando cuando un comando estaba mal redactado y verificando que el número fuera válido dentro de la lista.

Implementé también una serie de excepciones dentro de la lista de acuerdo al modelo y un par más fuera de, los try catch, en su mayoría estuvieron en el menú principal, siendo lo que el usuario tiene control sobre el input, y así, asegurándome que cualquier excepción tuviera su try catch fuera de ahí, previniendo posibles errores y escenarios donde el programa pudiera colapsar.

Jugué bastante con los strings y la manera de realizar input's tanto numéricos como de cadenas de caracteres, formando relaciones muy interesantes entre los métodos de la clase menú, pero que dan bastante estilo al resultado final; por ejemplo, en programas anteriores, al validar un dato, dar el mensaje de error y volverlo a pedir, se borraba todo lo que se había imprimido antes, dejando una pantalla sin mucho contexto por sí sola, ahora, usando ostreamstring, operándolo y pasándolo como string a otras funciones hice que este texto anterior no desapareciera, dando la impresión que fuera una ventana estilo pop up que salta y solo se va, sin alterar el demás contenido. Una pequeña mejora que logré por ser curioso con las herramientas del lenguaje.

Código Fuente

Carpeta include

exceptions.h

```
#ifndef __EXCEPTIONS_H__
#define __EXCEPTIONS_H__

#include <exception>
#include <string>

class Exception : public std::exception {
 private:
  std::string msg;

 public:
  Exception() noexcept : msg("Error Indefinido") {}
  Exception(const Exception& ex) noexcept : msg(ex.msg) {}
  Exception(const std::string& m) : msg(m) {}
  Exception& operator=(const Exception& ex) noexcept {
    msg = ex.msg;
    return *this;
  }

  virtual ~Exception() {}
  virtual const char* what() const noexcept { return msg.c_str(); }
};

class OperationCanceledException : public Exception {
 public:
  OperationCanceledException() noexcept : Exception("Operacion
Cancelada") {}

  OperationCanceledException(const OperationCanceledException& ex)
noexcept
      : Exception(ex) {}

  OperationCanceledException(const std::string& m) : Exception(m) {}

  OperationCanceledException& operator=(
      const OperationCanceledException& ex) noexcept {
    Exception::operator=(ex);  // reutiliza asignación de la base
    return *this;
  }

  virtual ~OperationCanceledException() {}
};
```

```
#endif  // __EXCEPTIONS_H__
```

list.hpp

```
#ifndef __LIST_H__
#define __LIST_H__

#include <iomanip>
#include <iostream>
#include <sstream>

#include "exceptions.hpp"

template <class T, int ARRAYSIZE = 50>
class List {
 private:
  T data[ARRAYSIZE];
  int last;

  void copyAll(const List<T, ARRAYSIZE>&);

 public:
  List<T, ARRAYSIZE>();
  List<T, ARRAYSIZE>(const List<T, ARRAYSIZE>&);

  bool isEmpty();
  bool isFull();
  void insertElement(const T&, const int&);
  void deleteData(const int&);
  T* retrieve(const int&);

  // Getter's
  int getFirstPosition() const;
  int getLastPosition() const;

  int getPrevPosition(const int&) const;
  int getNextPosition(const int&) const;

  std::string toString() const;

  void deleteAll();

  bool isRankingAvalible(const int&) const;
  bool isValidPosition(const int&) const;

  List<T, ARRAYSIZE> operator=(const List<T, ARRAYSIZE>&);
};

template <class T, int ARRAYSIZE>
List<T, ARRAYSIZE>::List() : last(-1) {}
```

```
template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::copyAll(const List<T, ARRAYSIZE>& other) {
  for (int i = 0; i < other.last; i++)
    this->data[i] = other.data[i];
  this->last = other.last;
}

template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isValidPosition(const int& position) const {
  return !(position > last || position < 0);
}

template <class T, int ARRAYSIZE>
List<T, ARRAYSIZE>::List(const List<T, ARRAYSIZE>& other) {
  copyAll(other);
}

template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isEmpty() {
  return this->last == -1;
}

template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isFull() {
  return this->last == (ARRAYSIZE - 1);
}

// Inserción en el Punto de Interés
template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::insertElement(const T& newData, const int&
position) {
  if (isFull())
    throw Exception("Lista Llena, InsertElement(List)");

  if (!isValidPosition(position) && position != last + 1)
    throw Exception("Posicion Invalida, InsertElement(List)");

  if (!isRankingAvalible(newData.getRanking()))
    throw Exception("Ranking no se puede repetir, InsertElement(List)");

  for (int i = last; i >= position; i--)
    this->data[i + 1] = this->data[i];
  this->data[position] = newData;
  last++;
}

template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::deleteData(const int& position) {
```

```
    if (!isValidPosition(position))
      throw Exception("Poscion Invalida, delteData(List)");

  for (int i = position; i < last; i++)
    this->data[i] = this->data[i + 1];
  last--;
}

template <class T, int ARRAYSIZE>
T* List<T, ARRAYSIZE>::retrieve(const int& position) {
  if (!isValidPosition(position))
    throw Exception("Posicion Invalida, retrieve(List)");
  return &data[position];
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getFirstPosition() const {
  return isEmpty() ? -1 : 0;
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getLastPosition() const {
  return this->last;
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getPrevPosition(const int& position) const {
  return (!isValidPosition(position) || position == 0) ? -1 : (position -
1);
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getNextPosition(const int& position) const {
  return (!isValidPosition(position) || position == last) ? -1 :
(position + 1);
}

template <class T, int ARRAYSIZE>
std::string List<T, ARRAYSIZE>::toString() const {
  int widthRanking = 10, widthSongName = 40, widthName = 35;
  int widthBorder = (widthRanking * 2) + widthSongName + (widthName * 2)
+ 1;
  std::stringstream oss;
  oss << std::setfill('=') << std::setw(widthBorder) << "" << std::endl;
  oss << std::setfill(' ');

  oss << "|" << std::setw(widthBorder / 2) << "LISTA DE EXITOS"
      << std::setw(widthBorder / 2) << "|" << std::endl;
```

```
  oss << std::setfill('-') << std::setw(widthBorder) << "" << std::endl;
  oss << std::setfill(' ');

  oss << std::left << std::setw(widthRanking) << "| N Lista";
  oss << std::left << std::setw(widthRanking) << "| Ranking";
  oss << std::left << std::setw(widthSongName) << "| Nombre de la
Cancion";
  oss << std::left << std::setw(widthName) << "| Nombre del Autor";
  oss << std::left << std::setw(widthName) << "| Nombre del Interprete";
  oss << "|" << std::endl;

  oss << std::setfill('-') << std::setw(widthBorder) << "" << std::endl;
  oss << std::setfill(' ');

  for (int i = 0; i <= last; i++) {
    oss << "| " << std::left << std::setw(widthRanking - 2) << i << "| "
        << std::left << std::setw(widthRanking - 2) <<
data[i].getRanking()
        << "| " << std::left << std::setw(widthSongName - 2)
        << data[i].getSongName() << "| " << std::left
        << std::setw(widthName - 2) << data[i].getAuthor().toString() <<
"| "
        << std::left << std::setw(widthName - 2)
        << data[i].getInterpreter().toString() << "| " << std::endl;
  }

  oss << std::setfill('=') << std::setw(widthBorder) << "" << std::endl;
  oss << std::setfill(' ');

  return oss.str();
}

template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::deleteAll() {
  this->last = -1;
}

template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isRankingAvalible(const int& ranking) const {
  for (int i = 0; i <= last; i++)
    if (data[i].getRanking() == ranking)
      return false;
  return true;
}

template <class T, int ARRAYSIZE>
List<T, ARRAYSIZE> List<T, ARRAYSIZE>::operator=(
```

```
    const List<T, ARRAYSIZE>& other) {
  copyAll(other);

  return *this;
}

#endif  // __LIST_H__
```

menu.hpp

```
#ifndef __MENU_H__
#define __MENU_H__

#include <iomanip>
#include <iostream>
#include <sstream>
#include <string>

#include "list.hpp"
#include "song.hpp"

class Menu {
 private:
  List<Song> songList;

  int readInteger(std::string, const int&, const int&);
  Name readName(std::string);
  std::string readLinePrompt(const std::string&, bool);
  void handleOption(const std::string&);

  void mainMenu();
  void insertSong();
  void deleteSong(const int&);
  void deleteAllSongs();
  void editSong(const int&);

 public:
  Menu();
  Menu(const Menu&);
  Menu(const List<Song>&);
};

#endif  // __MENU_H__
```

name.hpp

```cpp
#ifndef __NAME_H__
#define __NAME_H__

#include <string>
#include "exceptions.hpp"

class Name {
 private:
  std::string first;
  std::string last;

 public:
  Name();
  Name(const Name&);
  Name(const std::string&, const std::string&);

  // Interfaz
  // Setter's
  void setFirst(const std::string&);
  void setLast(const std::string&);

  // Getter's
  std::string getFirst() const;
  std::string getLast() const;

  std::string toString() const;

  Name& operator=(const Name&);
};
#endif  // __NAME_H__
```

song.hpp

```
#ifndef __SONG_H__
#define __SONG_H__

#include <iomanip>
#include <sstream>
#include <string>

#include "name.hpp"

class Song {
 private:
  int ranking;
  std::string songName;
  Name author;
  Name interpreter;

 public:
  Song();
  Song(const Song&);

  /// @brief
  /// @param  Ranking
  /// @param  NombreCancion
  /// @param  NombreAutor
  /// @param  NombreInterprete
  Song(const int&, const std::string&, const Name&, const Name&);

  // Interfaz:
  // Setter's
  void setRanking(const int&);
  void setSongName(const std::string&);
  void setAuthor(const Name&);
  void setInterpreter(const Name&);

  // Getter's
  int getRanking() const;
  std::string getSongName() const;
  Name getAuthor() const;
  Name getInterpreter() const;

  std::string toString() const;

  Song& operator=(const Song&);
};
#endif  // __SONG_H__
```

Carpeta src

main.cpp

```
#include "menu.hpp"

int main() {
  Menu menu;

  return 0;
}
```

menu.cpp

```
#include "menu.hpp"

using namespace std;

Menu::Menu() {
  mainMenu();
}

Menu::Menu(const Menu& other) : songList(other.songList) {
  mainMenu();
}

Menu::Menu(const List<Song>& s) : songList(s) {
  mainMenu();
}

int Menu::readInteger(string oss,
                      const int& lowerLimit,
                      const int& upperLimit) {
  string aux("");
  int result;
  while (true) {
    try {
      system("CLS");
      cout << oss;
      getline(cin, aux);
      result = stoi(aux);

      if (result > upperLimit || result < lowerLimit)
        throw Exception("Numero Fuera de Rango");
      break;
    } catch (std::invalid_argument) {
      system("CLS");
      cout << "Entrada invalida" << endl;
      cout << "Intente nuevamente" << endl;
      system("PAUSE");
    } catch (Exception msg) {
      system("CLS");
      cout << msg.what() << endl;
      system("PAUSE");
    }
  }

  return result;
}

Name Menu::readName(string prompt) {
```

```
    Name result;
    result.setFirst(readLinePrompt(prompt, false));
    prompt += result.getFirst() + "\n";
    result.setLast(readLinePrompt(prompt + "Ingrese el Apellido: ",
false));

    return result;
}

string Menu::readLinePrompt(const string& prompt, bool allowEmpty) {
    string result;
    while (true) {
        system("CLS");
        cout << prompt;
        getline(cin, result);
        if (result == "0")
            throw OperationCanceledException();
        if (!allowEmpty && result.empty()) {
            system("CLS");
            cout << "No puede estar vacio.\nIntentelo nuevamente." << endl;
            system("PAUSE");
            continue;
        }
        return result;
    }
}

void Menu::handleOption(const std::string& promt) {
    std::string response;

    while (true) {
        system("CLS");
        std::cout << promt;
        std::getline(std::cin, response);

        if (response.empty()) {
            system("CLS");
            std::cout << "Comando invalido\n";
            system("PAUSE");
            continue;
        }

        // letra de opción (normalizamos a mayúscula)
        char option = static_cast<char>(
            std::toupper(static_cast<unsigned char>(response[0])));

        // buscar primer dígito después de la letra (saltando espacios)
        std::size_t pos = 1;
```

```
 while (pos < response.size() &&
        std::isspace(static_cast<unsigned char>(response[pos])))
   ++pos;

bool hasNumber = false;
int index = -1;
if (pos < response.size() &&
    std::isdigit(static_cast<unsigned char>(response[pos]))) {
  std::size_t start = pos;
  std::size_t end = start;
  while (end < response.size() &&
         std::isdigit(static_cast<unsigned char>(response[end])))
    ++end;
  std::string numstr = response.substr(start, end - start);
  try {
    index = std::stoi(numstr);
    hasNumber = true;
  } catch (...) {
    hasNumber = false;
  }
}

switch (option) {
  case 'A':
    insertSong();
    break;

  case 'B':
    if (!hasNumber) {
      system("CLS");
      std::cout << "Falta numero de posicion. Ej: B2\n";
      system("PAUSE");
      break;
    }
    if (!songList.isValidPosition(index)) {
      system("CLS");
      std::cout << "Posicion de lista invalida\n";
      system("PAUSE");
      break;
    }
    editSong(index);
    break;

  case 'C':
    if (!hasNumber) {
      system("CLS");
      std::cout << "Falta numero de posicion. Ej: C12\n";
      system("PAUSE");
```

```
        break;
      }
      if (!songList.isValidPosition(index)) {
        system("CLS");
        std::cout << "Posicion de lista invalida\n";
        system("PAUSE");
        break;
      }
      deleteSong(index);
      break;

    case 'D':
      if (!hasNumber) {
        system("CLS");
        std::cout << "Falta numero de posiciin. Ej: D12\n";
        system("PAUSE");
        break;
      }
      if (!songList.isValidPosition(index)) {
        system("CLS");
        std::cout << "Posicion de lista invalida\n";
        system("PAUSE");
        break;
      }
      system("CLS");
      {
        Song* s = songList.retrieve(index);
        if (s)
          std::cout << s->toString();
        else
          std::cout << "Cancion no encontrada\n";
      }
      system("PAUSE");
      break;
    case 'E':
      deleteAllSongs();
      break;
    case 'F':
      system("CLS");
      std::cout << "Saliendo del Programa.\nTenga un Lindo Dia :D\n";
      system("PAUSE");
      exit(-1);
      return;

    default:
      system("CLS");
      std::cout << "Comando invalido\nIntentelo nuevamente.\n";
      system("PAUSE");
```

```
        break;
      }  // switch
    }  // while
}

void Menu::mainMenu() {
  ostringstream oss;
  system("CLS");
  oss << songList.toString();
  oss << "Opciones: \n";
  oss << "[A] Agregar una Cancion.  [B<n>] Editar una Cancion [C<n>]
Eliminar "
         "una Cancion. [D<n>] Mostrar Detalles de Cancion. [E] Eliminar
Todas "
         "las Canciones. \n[F] Salir.\n";
  oss << "Seleccione un Comando: ";

  handleOption(oss.str());
}

void Menu::insertSong() {
  int widthBorder = 100;
  Song newSong;
  string myString("");
  int myInt(0);
  Name myName;
  ostringstream oss;

  do {
    system("CLS");
    // Linea Exterior
    oss << left << setfill('=') << setw(widthBorder) << "" << endl;
    oss << setfill(' ');

    // Título de Ventana
    oss << setw(widthBorder / 2 - 7) << "| " << "INSERTAR EXITO"
        << setw((widthBorder / 2) - 8) << "" << "|" << endl;
    oss << setfill('-') << setw(widthBorder) << "" << endl;
    oss << setfill(' ');

    oss << "Ingrese el Nombre de la Cancion: ";

    while (true) {
      try {
        myString = readLinePrompt(oss.str(), false);
        newSong.setSongName(myString);
        break;
      } catch (Exception msg) {
```

```
      system("CLS");
      cout << msg.what() << endl;
      cout << "Vuelva a intentarlo\n";
      system("PAUSE");
    }
  }

  oss << newSong.getSongName() << endl;
  oss << "Ingrese el Ranking de la Cancion: ";

  while (true) {
    try {
      system("CLS");
      myInt = readInteger(oss.str(), 0, 1000);
      if (!songList.isRankingAvalible(myInt))
        throw std::invalid_argument("Ranking ya utilizado");
      newSong.setRanking(myInt);
      break;
    } catch (Exception msg) {
      system("CLS");
      cout << msg.what() << endl;
      system("PAUSE");
    } catch (std::invalid_argument msg) {
      system("CLS");
      cout << msg.what() << endl;
      system("PAUSE");
    }
  }

  oss << newSong.getRanking() << endl;
  oss << "Ingrese el Nombre del Autor: ";
  myName = readName(oss.str());
  oss << myName.getFirst() << endl;
  oss << "Ingrese el Apellido: " << myName.getLast() << endl;

  newSong.setAuthor(myName);

  oss << "Ingrese el Nombre del Interprete: ";
  myName = readName(oss.str());
  oss << myName.getFirst() << endl;
  oss << "Ingrese el Apellido: " << myName.getLast() << endl;

  newSong.setInterpreter(myName);

  if (songList.isEmpty())
    songList.insertElement(newSong, 0);
  else {
    oss << "Ingrese la posicion en la lista que tendra la cancion: ";
```

```
        while (true) {
          try {
            myInt = readInteger(oss.str(), 0, 49);
            songList.insertElement(newSong, myInt);
            oss << myInt << endl;
            break;
          } catch (Exception msg) {
            system("CLS");
            cout << msg.what() << endl;
            cout << "Intente Nuevamente." << endl;
            system("PAUSE");
          }
        }
      }

      oss << "Cancion Agregada con Exito!." << endl;
      oss << "Desea Agregar Otra Cancion? (1. Si / 2. No): ";

      myInt = readInteger(oss.str(), 1, 2);

      oss.str("");
      oss.clear();
    } while (myInt != 2);

    mainMenu();
}

void Menu::deleteSong(const int& position) {
    system("CLS");
    ostringstream oss;
    int response;

    Song* target = songList.retrieve(position);
    oss << target->toString();
    oss << "Esta seguro que desea eliminar esta cancion? (1. Si/ 2. No): ";
    response = readInteger(oss.str(), 1, 2);
    if (response == 1) {
      songList.deleteData(position);
      oss << endl << "Cancion Eliminada con Exito!" << endl;
    } else
      oss << endl << "Operacion Cancelada" << endl;

    system("CLS");
    cout << oss.str();

    system("PAUSE");
    mainMenu();
}
```

```
void Menu::deleteAllSongs() {
  system("CLS");
  if (songList.getLastPosition() == -1) {
    cout << "Aun no hay canciones para eliminar" << endl;
    system("PAUSE");
    return;
  }

  ostringstream oss;
  int widhtBorder = 50;

  oss << left << setfill('=') << setw(widhtBorder) << "" << endl;
  oss << setfill(' ');
  oss << setw(widhtBorder / 2 - 13) << "| " << "ELIMMINAR TODAS LAS
CANCIONES"
      << setw((widhtBorder / 2) - 17) << "" << "|" << endl;
  oss << setfill('=') << setw(widhtBorder) << "" << endl;

  oss << "Esta seguro que desea eliminar las " <<
songList.getLastPosition() + 1
      << " canciones? (1. Si/ 2. No): ";
  int response = readInteger(oss.str(), 1, 2);
  system("CLS");
  if (response == 1) {
    songList.deleteAll();
    cout << "Canciones eliminadas con Exito!" << endl;
    cout << "Base de Datos Vacia." << endl;
  } else {
    cout << "Operacion Cancelada." << endl;
  }
  system("PAUSE");
  mainMenu();
}

void Menu::editSong(const int& position) {
  ostringstream oss;
  Song* target = songList.retrieve(position);
  int editOption, newRanking;
  string dataString;
  Name newName;

  oss << target->toString();
  oss << "5 Salir\n";

  editOption = readInteger(
      oss.str() + "Elige el atributo que quieras cambiar (1-5): ", 1, 5);
```

```
 switch (editOption) {
  case 1:
    oss << "Ingrese el Nuevo Ranking de la Cancion: ";
    newRanking = readInteger(oss.str(), 1, 50);
    if (!songList.isRankingAvalible(newRanking)) {
      system("CLS");
      cout << "El ranking ya esta ocupado" << endl;
      system("PAUSE");
      break;
    }
    target->setRanking(newRanking);
    cout << "Cambio hecho con Exito!";
    break;
  case 2:
    oss << "Ingrese el nuevo nombre de la cancion: ";
    dataString = readLinePrompt(oss.str(), false);
    target->setSongName(dataString);
    cout << "Cambio hecho con Exito!";
    system("PAUSE");
    break;
  case 3:
    oss << "Ingrese el nuevo autor de la cancion: ";
    newName = readName(oss.str());
    target->setAuthor(newName);
    cout << "Cambio hecho con Exito!";
    break;
  case 4:
    oss << "Ingrese el nuevo interprete de la cancion: ";
    newName = readName(oss.str());
    target->setInterpreter(newName);
    cout << "Cambio hecho con Exito!";
    break;
  case 5:
    return;
  default:
    break;
 }
 system("PAUSE");

 mainMenu();
}
```

name.cpp

```cpp
#include "name.hpp"

Name::Name() : first("default"), last("default") {}

Name::Name(const Name& other) : first(other.first), last(other.last) {}

Name::Name(const std::string& f, const std::string& l) : first(f),
last(l) {}

void Name::setFirst(const std::string& first) {
  if (first.empty())
    throw Exception("Nombre no puede estar vacío, setFirst(Name)");
  this->first = first;
}

void Name::setLast(const std::string& last) {
  if (last.empty())
    throw Exception("Apellido no puede estar vacío, setLast(Name)");
  this->last = last;
}

std::string Name::getFirst() const {
  return this->first;
}

std::string Name::getLast() const {
  return this->last;
}

std::string Name::toString() const {
  return this->first + " " + this->last;
}

Name& Name::operator=(const Name& other) {
  this->first = other.first;
  this->last = other.last;

  return *this;
}
```

song.cpp

```cpp
#include "song.hpp"

using namespace std;

Song::Song() : ranking(-1), songName("default"), author(), interpreter()
{}

Song::Song(const Song& other)
    : ranking(other.ranking),
      songName(other.songName),
      author(other.author),
      interpreter(other.interpreter) {}

Song::Song(const int& r, const std::string& n, const Name& a, const Name&
i)
    : ranking(r), songName(n), author(a), interpreter(i) {}

void Song::setRanking(const int& ranking) {
  if (ranking <= 0)
    throw Exception("El ranking debe ser positivo");
  this->ranking = ranking;
}

void Song::setSongName(const std::string& songName) {
  if (songName.empty())
    throw Exception("El nombre no puede estar vacio.");
  this->songName = songName;
}

void Song::setAuthor(const Name& author) {
  this->author = author;  // Name tiene sus propias validaciones
}

void Song::setInterpreter(const Name& interpreter) {
  this->interpreter = interpreter;
}

int Song::getRanking() const {
  return this->ranking;
}

std::string Song::getSongName() const {
  return this->songName;
}

Name Song::getAuthor() const {
  return this->author;
```

```
}

Name Song::getInterpreter() const {
  return this->interpreter;
}

std::string Song::toString() const {
  ostringstream oss;
  int widthBorder = 60;

  oss << setfill('=') << setw(widthBorder) << "" << endl;
  oss << setfill(' ');

  oss << "| " << setw((widthBorder / 2) + 10) << "INFORMACION DE LA
CANCION"
      << setw((widthBorder / 2) - 12) << "|" << endl;

  oss << setfill('-') << setw(widthBorder) << "" << endl;
  oss << setfill(' ');

  oss << "Posicion en el Ranking: " << ranking << endl;
  oss << "Nombre de la Cancion: " << songName << endl;
  oss << "Nombre del Autor: " << author.toString() << endl;
  oss << "Nombre del Inteprete: " << interpreter.toString() << endl;

  oss << endl << setfill('=') << setw(widthBorder) << "" << endl;
  oss << setfill(' ');

  return oss.str();
}

Song& Song::operator=(const Song& other) {
  this->ranking = other.ranking;
  this->songName = other.songName;
  this->author = other.author;
  this->interpreter = other.interpreter;

  return *this;
}
```

Ejecución del Programa

Comenzando con la ejecución, esta es la primera pantalla:

```
================================================================================================================================
|                                                    LISTA DE EXITOS                                                           |
--------------------------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor                | Nombre del Interprete           |
--------------------------------------------------------------------------------------------------------------------------------
================================================================================================================================
Opciones:
[A] Agregar una Cancion.  [B<n>] Editar una Cancion [C<n>] Eliminar una Cancion. [D<n>] Mostrar Detalles de Cancion. [E] Eliminar Todas las Canciones.
[F] Salir.
Seleccione un Comando:
```

Este es el formato de tabla que se actualizará cada vez con cada edición, tiene las características de las canciones y su posición en la lista.

Si intentamos acceder a una opción que requiera de canciones, como por ejemplo la opción E para eliminar todas las canciones, nos aparecerá este mensaje:

```
Aun no hay canciones para eliminar
Presione una tecla para continuar . . .
```

Y regresamos al menú principal. Entonces, pues, vayamos a la opción A, para registrar una nueva canción:

```
================================================================================================================
|                                                INSERTAR EXITO                                                |
----------------------------------------------------------------------------------------------------------------
Ingrese el Nombre de la Cancion:
```

Primero nos pide el nombre, que los string's en este programa no pueden estar vacíos, por lo que si solo damos enter nos aparece lo siguiente:

```
No puede estar vacio.
Intentelo nuevamente.
Presione una tecla para continuar . . .
```

Y nos regresa para insertar una canción. Si seguimos poniendo datos la agregaremos con éxito:

```
================================================================================================================
|                                                INSERTAR EXITO                                                |
----------------------------------------------------------------------------------------------------------------
Ingrese el Nombre de la Cancion: Etrusto Unico
Ingrese el Ranking de la Cancion: 1
Ingrese el Nombre del Autor: Jose
Ingrese el Apellido: Madero
Ingrese el Nombre del Interprete: Jose
Ingrese el Apellido: Madero
Cancion Agregada con Exito!.
Desea Agregar Otra Cancion? (1. Si / 2. No):
```

Como es la primera inserción que se hace, no nos pregunta en qué posición de la lista insertarla, simplemente lo hace en la posición 0, la única posición viable de la lista.

Si agregamos otra nueva canción, esta opción se nos presentará, a su vez, el ranking no puede estar repetido, ya que, en las listas de música, el ranking no puede estar repetido, una canción está en uno y no lo comparte.

Agreguemos otra canción para ver estas cosas. Si intentemos hacer esto:

```
=============================================================================================================
|                                              INSERTAR EXITO                                               |
-------------------------------------------------------------------------------------------------------------
Ingrese el Nombre de la Cancion: Labios Rotos
Ingrese el Ranking de la Cancion: 1
```

Teniendo ya una canción en el primer ranking, nos aparecerá un mensaje de error y nos volverá a pedir el ranking:

```
Ranking ya utilizado
Presione una tecla para continuar . . .
```

Siguiendo con la inserción:

```
=============================================================================================================
|                                              INSERTAR EXITO                                               |
-------------------------------------------------------------------------------------------------------------
Ingrese el Nombre de la Cancion: Labios Rotos
Ingrese el Ranking de la Cancion: 24
Ingrese el Nombre del Autor: Leon
Ingrese el Apellido: Larregui
Ingrese el Nombre del Interprete: Zoe
Ingrese el Apellido: Band
Ingrese la posicion en la lista que tendra la cancion: 0
Cancion Agregada con Exito!.
Desea Agregar Otra Cancion? (1. Si / 2. No):
```

La insertamos en la posición 0 (inserción en el punto de interés), por lo que en el menú debería aparecer en primera posición justo antes de la canción “Etrusto Único”, y si regresamos al menú:

```
=============================================================================================================
|                                              LISTA DE EXITOS                                              |
-------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor          | Nombre del Interprete |
-------------------------------------------------------------------------------------------------------------
| 0       | 24      | Labios Rotos                        | Leon Larregui             | Zoe Band              |
| 1       | 1       | Etrusto Unico                       | Jose Madero               | Jose Madero           |
=============================================================================================================
Opciones:
[A] Agregar una Cancion.  [B<n>] Editar una Cancion [C<n>] Eliminar una Cancion. [D<n>] Mostrar Detalles de Cancion. [E] Eliminar Todas las Canciones.
[F] Salir.
Seleccione un Comando:
```

Vemos que sí es así.

Agreguemos bastantes más canciones para ver con claridad las pruebas; después de unas cuantas inserciones, el menú tiene este aspecto:

```
================================================================================================================
|                                                LISTA DE EXITOS                                               |
----------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor          | Nombre del Interprete     |
----------------------------------------------------------------------------------------------------------------
| 0       | 1       | Etrusto Unico                       | Jose Madero               | Jose Madero               |
| 1       | 7       | Rayo de Luz                         | Jose Madero               | Jose Madero               |
| 2       | 12      | Los Malaventurados No Lloran        | Jose Madero               | Panda Band                |
| 3       | 22      | Soñé                                | Jose Madero               | Panda Band                |
| 4       | 5       | Cita en el Quirófano                | Jose Madero               | Panda Band                |
| 5       | 17      | Cuando No Es Como Debería Ser       | Jose Madero               | Panda Band                |
| 6       | 24      | Labios Rotos                        | Leon Larregui             | Zoe Band                  |
| 7       | 2       | Vía Láctea                          | Leon Larregui             | Zoe Band                  |
| 8       | 30      | Paula                               | Leon Larregui             | Zoe Band                  |
| 9       | 14      | Arrullo de Estrellas                | Leon Larregui             | Zoe Band                  |
| 10      | 47      | Nada                                | Leon Larregui             | Zoe Band                  |
| 11      | 28      | La Ley del Monte                    | Vicente Fernandez         | Vicente Fernandez         |
| 12      | 41      | Mujeres Divinas                     | Vicente Fernandez         | Vicente Fernandez         |
| 13      | 3       | Volver Volver                       | Vicente Fernandez         | Vicente Fernandez         |
| 14      | 33      | Hermoso Cariño                      | Vicente Fernandez         | Vicente Fernandez         |
| 15      | 19      | El Rey                              | Vicente Fernandez         | Vicente Fernandez         |
| 16      | 8       | Julieta                             | Pablo Preciado            | Latin Mafia               |
| 17      | 21      | No Me Conozco                       | Miloh Smith               | Latin Mafia               |
| 18      | 29      | Patadas de Ahogado                  | Miloh Smith               | Latin Mafia               |
| 19      | 38      | Morir De Amor                       | Pablo Preciado            | Latin Mafia               |
| 20      | 44      | Eres un Ángel                       | Miloh Smith               | Latin Mafia               |
| 21      | 6       | Amorfoda                            | Benito Martinez           | Bad Bunny                 |
| 22      | 40      | Ojitos Lindos                       | Benito Martinez           | Bad Bunny                 |
| 23      | 15      | Dákiti                              | Benito Martinez           | Bad Bunny                 |
| 24      | 25      | Callaita                            | Benito Martinez           | Bad Bunny                 |
| 25      | 49      | Tití Me Preguntó                    | Benito Martinez           | Bad Bunny                 |
| 26      | 18      | Eres                                | Leon Larregui             | Zoe Band                  |
| 27      | 11      | Love                                | Leon Larregui             | Zoe Band                  |
| 28      | 23      | Déjate Conectar                     | Leon Larregui             | Zoe Band                  |
| 29      | 34      | Final Feliz                         | Leon Larregui             | Zoe Band                  |
| 30      | 9       | Azul                                | Leon Larregui             | Zoe Band                  |
| 31      | 13      | Solo a Terceros                     | Jose Madero               | Panda Band                |
| 32      | 35      | Miércoles                           | Jose Madero               | Panda Band                |
| 33      | 27      | Disculpa los Malos Pensamientos     | Jose Madero               | Panda Band                |
| 34      | 10      | Ya No Jalaba                        | Jose Madero               | Panda Band                |
| 35      | 36      | Un Hombre con Suerte                | Vicente Fernandez         | Vicente Fernandez         |
```

Tenemos canciones en una lista con un ranking único cada uno con su información correspondiente. Visitemos la opción de editar una canción que es el comando B<n> donde B es la opción de editar y <n> el número de la posición de la lista.

Editemos la canción en la posición 13, la canción “Volver Volver”, para ello, ingresamos el comando B13:

```
============================================================
|                INFORMACION DE LA CANCION                 |
------------------------------------------------------------
Posicion en el Ranking: 3
Nombre de la Cancion: Volver Volver
Nombre del Autor: Vicente Fernandez
Nombre del Inteprete: Vicente Fernandez

============================================================
5 Salir
Elige el atributo que quieras cambiar (1-5):
```

Nos muestra la información de la canción y nos pide la posición del dato que queremos editar, cabe destacar que este input también esta preparado mediante un try catch, para poder filtrar si se le ingresa un dato inválido o cosas como un string:

```
Entrada invalida
Intente nuevamente
Presione una tecla para continuar . . .
```

Editemos el nombre de la canción:

```
============================================================
|                  INFORMACION DE LA CANCION                |
------------------------------------------------------------
Posicion en el Ranking: 3
Nombre de la Cancion: Volver Volver
Nombre del Autor: Vicente Fernandez
Nombre del Inteprete: Vicente Fernandez

============================================================
5 Salir
Ingrese el nuevo nombre de la cancion:
```

Nos pide el nuevo nombre, no acepta string's vacíos, y una vez ingresemos otro nombre:

```
============================================================
|                  INFORMACION DE LA CANCION                |
------------------------------------------------------------
Posicion en el Ranking: 3
Nombre de la Cancion: Volver Volver
Nombre del Autor: Vicente Fernandez
Nombre del Inteprete: Vicente Fernandez

============================================================
5 Salir
Ingrese el nuevo nombre de la cancion: Un Millon de Primaveras
Cambio hecho con Exito!Presione una tecla para continuar . . .
```

Se nos avisa que el cambio se realizó con éxito y nos devuelve al menú principal, donde podemos ver reflejado el cambio:

```
================================================================================================================
|                                                LISTA DE EXITOS                                               |
----------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor          | Nombre del Interprete     |
----------------------------------------------------------------------------------------------------------------
| 0       | 1       | Etrusto Unico                       | Jose Madero               | Jose Madero               |
| 1       | 7       | Rayo de Luz                         | Jose Madero               | Jose Madero               |
| 2       | 12      | Los Malaventurados No Lloran        | Jose Madero               | Panda Band                |
| 3       | 22      | Soñé                                | Jose Madero               | Panda Band                |
| 4       | 5       | Cita en el Quirófano                | Jose Madero               | Panda Band                |
| 5       | 17      | Cuando No Es Como Debería Ser       | Jose Madero               | Panda Band                |
| 6       | 24      | Labios Rotos                        | Leon Larregui             | Zoe Band                  |
| 7       | 2       | Vía Láctea                          | Leon Larregui             | Zoe Band                  |
| 8       | 30      | Paula                               | Leon Larregui             | Zoe Band                  |
| 9       | 14      | Arrullo de Estrellas                | Leon Larregui             | Zoe Band                  |
| 10      | 47      | Nada                                | Leon Larregui             | Zoe Band                  |
| 11      | 28      | La Ley del Monte                    | Vicente Fernandez         | Vicente Fernandez         |
| 12      | 41      | Mujeres Divinas                     | Vicente Fernandez         | Vicente Fernandez         |
| 13      | 3       | Un Millon de Primaveras             | Vicente Fernandez         | Vicente Fernandez         |
| 14      | 33      | Hermoso Cariño                      | Vicente Fernandez         | Vicente Fernandez         |
| 15      | 19      | El Rey                              | Vicente Fernandez         | Vicente Fernandez         |
| 16      | 8       | Julieta                             | Pablo Preciado            | Latin Mafia               |
| 17      | 21      | No Me Conozco                       | Miloh Smith               | Latin Mafia               |
| 18      | 29      | Patadas de Ahogado                  | Miloh Smith               | Latin Mafia               |
| 19      | 38      | Morir De Amor                       | Pablo Preciado            | Latin Mafia               |
| 20      | 44      | Eres un Ángel                       | Miloh Smith               | Latin Mafia               |
| 21      | 6       | Amorfoda                            | Benito Martinez           | Bad Bunny                 |
| 22      | 40      | Ojitos Lindos                       | Benito Martinez           | Bad Bunny                 |
| 23      | 15      | Dákiti                              | Benito Martinez           | Bad Bunny                 |
| 24      | 25      | Callaita                            | Benito Martinez           | Bad Bunny                 |
| 25      | 49      | Tití Me Preguntó                    | Benito Martinez           | Bad Bunny                 |
| 26      | 18      | Eres                                | Leon Larregui             | Zoe Band                  |
| 27      | 11      | Love                                | Leon Larregui             | Zoe Band                  |
| 28      | 23      | Déjate Conectar                     | Leon Larregui             | Zoe Band                  |
| 29      | 34      | Final Feliz                         | Leon Larregui             | Zoe Band                  |
| 30      | 9       | Azul                                | Leon Larregui             | Zoe Band                  |
| 31      | 13      | Solo a Terceros                     | Jose Madero               | Panda Band                |
| 32      | 35      | Miércoles                           | Jose Madero               | Panda Band                |
| 33      | 27      | Disculpa los Malos Pensamientos     | Jose Madero               | Panda Band                |
| 34      | 10      | Ya No Jalaba                        | Jose Madero               | Panda Band                |
| 35      | 36      | Un Hombre con Suerte                | Vicente Fernandez         | Vicente Fernandez         |
```

Observamos que la posición 13 ya fue actualizada.

Desde este menú, igualmente no se permite actualizar un raking por uno ya ocupado, se avisa y se pide otro, y está limitado del 1 al 50 (dado que el problema indica que se guardarán las primeras 50 canciones del ranking)

Intentemos ahora eliminar una canción con el comando C<n>, eliminemos, por ejemplo, la canción: “Arrullo de Estrellas”, es la posición 9. Ingresamos C9:

```
============================================================
|                INFORMACION DE LA CANCION                 |
------------------------------------------------------------
Posicion en el Ranking: 14
Nombre de la Cancion: Arrullo de Estrellas
Nombre del Autor: Leon Larregui
Nombre del Inteprete: Zoe Band

============================================================
Esta seguro que desea eliminar esta cancion? (1. Si/ 2. No):
```

Se nos muestra la canción y nos pide una confirmación, y al darle acceso:

```
============================================================
|                INFORMACION DE LA CANCION                 |
------------------------------------------------------------
Posicion en el Ranking: 14
Nombre de la Cancion: Arrullo de Estrellas
Nombre del Autor: Leon Larregui
Nombre del Inteprete: Zoe Band

============================================================
Esta seguro que desea eliminar esta cancion? (1. Si/ 2. No):
Cancion Eliminada con Exito!
Presione una tecla para continuar . . .
```

Regresamos al menú, donde la posición 9 ya no es la misma y la lista sufrió un corrimiento:

```
================================================================================================================
|                                                LISTA DE EXITOS                                               |
----------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor          | Nombre del Interprete     |
----------------------------------------------------------------------------------------------------------------
| 0       | 1       | Etrusto Unico                       | Jose Madero               | Jose Madero               |
| 1       | 7       | Rayo de Luz                         | Jose Madero               | Jose Madero               |
| 2       | 12      | Los Malaventurados No Lloran        | Jose Madero               | Panda Band                |
| 3       | 22      | Soñé                                | Jose Madero               | Panda Band                |
| 4       | 5       | Cita en el Quirófano                | Jose Madero               | Panda Band                |
| 5       | 17      | Cuando No Es Como Debería Ser       | Jose Madero               | Panda Band                |
| 6       | 24      | Labios Rotos                        | Leon Larregui             | Zoe Band                  |
| 7       | 2       | Vía Láctea                          | Leon Larregui             | Zoe Band                  |
| 8       | 30      | Paula                               | Leon Larregui             | Zoe Band                  |
| 9       | 47      | Nada                                | Leon Larregui             | Zoe Band                  |
| 10      | 28      | La Ley del Monte                    | Vicente Fernandez         | Vicente Fernandez         |
| 11      | 41      | Mujeres Divinas                     | Vicente Fernandez         | Vicente Fernandez         |
| 12      | 3       | Un Millon de Primaveras             | Vicente Fernandez         | Vicente Fernandez         |
| 13      | 33      | Hermoso Cariño                      | Vicente Fernandez         | Vicente Fernandez         |
| 14      | 19      | El Rey                              | Vicente Fernandez         | Vicente Fernandez         |
| 15      | 8       | Julieta                             | Pablo Preciado            | Latin Mafia               |
| 16      | 21      | No Me Conozco                       | Miloh Smith               | Latin Mafia               |
| 17      | 29      | Patadas de Ahogado                  | Miloh Smith               | Latin Mafia               |
| 18      | 38      | Morir De Amor                       | Pablo Preciado            | Latin Mafia               |
| 19      | 44      | Eres un Ángel                       | Miloh Smith               | Latin Mafia               |
| 20      | 6       | Amorfoda                            | Benito Martinez           | Bad Bunny                 |
| 21      | 40      | Ojitos Lindos                       | Benito Martinez           | Bad Bunny                 |
| 22      | 15      | Dákiti                              | Benito Martinez           | Bad Bunny                 |
| 23      | 25      | Callaita                            | Benito Martinez           | Bad Bunny                 |
| 24      | 49      | Tití Me Preguntó                    | Benito Martinez           | Bad Bunny                 |
| 25      | 18      | Eres                                | Leon Larregui             | Zoe Band                  |
| 26      | 11      | Love                                | Leon Larregui             | Zoe Band                  |
| 27      | 23      | Déjate Conectar                     | Leon Larregui             | Zoe Band                  |
| 28      | 34      | Final Feliz                         | Leon Larregui             | Zoe Band                  |
| 29      | 9       | Azul                                | Leon Larregui             | Zoe Band                  |
| 30      | 13      | Solo a Terceros                     | Jose Madero               | Panda Band                |
| 31      | 35      | Miércoles                           | Jose Madero               | Panda Band                |
| 32      | 27      | Disculpa los Malos Pensamientos     | Jose Madero               | Panda Band                |
| 33      | 10      | Ya No Jalaba                        | Jose Madero               | Panda Band                |
| 34      | 36      | Un Hombre con Suerte                | Vicente Fernandez         | Vicente Fernandez         |
| 35      | 26      | Estos Celos                         | Vicente Fernandez         | Vicente Fernandez         |
```

Donde las posiciones desde la 9 en adelante fueron actualizadas.

El comando D<n> solo nos muestra una pequeña tabla con la canción aislada, ingresando D1 para ver la canción “Rayo de Luz” del artista José Madero:

```
============================================================
|                INFORMACION DE LA CANCION                 |
------------------------------------------------------------
Posicion en el Ranking: 7
Nombre de la Cancion: Rayo de Luz
Nombre del Autor: Jose Madero
Nombre del Inteprete: Jose Madero

============================================================
Presione una tecla para continuar . . .
```

Y nos regresa al menú principal.

Como última funcionalidad, el Comando E elimina todas las canciones, y se muestra lo siguiente al darle:

```
================================================
|           ELIMMINAR TODAS LAS CANCIONES       |
================================================
Esta seguro que desea eliminar las 47 canciones? (1. Si/ 2. No):
```

Nos pide una confirmación para eliminar la totalidad de canciones que hay (en este ejemplo, 47 canciones), y si le damos que sí:

```
Canciones eliminadas con Exito!
Base de Datos Vacia.
Presione una tecla para continuar . . .
```

Y al regresar al menú principal:

```
=============================================================================================================
|                                                LISTA DE EXITOS                                            |
-------------------------------------------------------------------------------------------------------------
| N Lista | Ranking | Nombre de la Cancion                | Nombre del Autor              | Nombre del Interprete          |
-------------------------------------------------------------------------------------------------------------
=============================================================================================================
Opciones:
[A] Agregar una Cancion.  [B<n>] Editar una Cancion [C<n>] Eliminar una Cancion. [D<n>] Mostrar Detalles de Cancion. [E] Eliminar Todas las Canciones.
[F] Salir.
Seleccione un Comando:
```

No tenemos nada, por lo que volvimos al punto del comienzo.

Finalmente, salimos del programa con el comando F:

```
Saliendo del Programa.
Tenga un Lindo Dia :D
Presione una tecla para continuar . . .
```

Se nos da una pequeña despedida y salimos del ejecutable.

Conclusiones.

Realizar este trabajo sobre listas estáticas fue muy enriquecedor, en un principio puede verse como un simple arreglo de elementos, pero es sumamente adaptable al poder agregarle operaciones específicas para el problema que quieres solucionar, experimenté bastante con ella y me puse a pensar en sus utilidades en otras situaciones, sería interesante aplicarla a una clase padre y estar trabajando con apuntadores, aunque también habría que tener cuidado para no que no hay fugas de memoria y manejar muy bien los destructores para que esto no pase. El manejo de excepciones en un try catch era algo, que si bien, ya conocía, no había puesto tan en práctica, y es muy útil para identificar distintos tipos de excepciones y manejarnos en consecuencia.

La lista como una estructura de datos se adecuó bastante bien a este problema, sin embargo, la implementación también puede verse sus limitaciones, siendo la principal de ellas su falta de flexibilidad en el crecimiento dinámico, aunque esto no lo quita valor, es una herramienta valiosa para problemas en los que se conoce de antemano la cantidad de datos con los que se trabajará, como fue este problema; además, la lista es la base comprender el resto de estructuras de datos. La búsqueda, inserción, eliminación y demás son métodos que tienen una buena relación de costo computacional en ciertos casos, sin embargo, conforme avance el curso y adquiramos nuevas herramientas en formato de estructuras de datos, gran parte de la solución será identificar la mejor estructura para un determinado problema en función de lo que necesitamos.

Este es un comienzo muy emocionante a la par que interesante para estructuras de datos, espero pronto conocer estructuras más avanzadas, dinámicas y de distintas índoles.