



ACTIVIDAD 05: **MÉTODOS DE BÚSQUEDA**

ESTUDIANTE: **EFRAIN ROBLES PULIDO**

CODIGO: **221350095**

NOMBRE DE LA MATERIA: **ESTRUCTURAS DE DATOS I**

SECCIÓN: **D12**

CLAVE: **I5886**

FECHA: **DOMINGO 20 DE FEBRERO DE 2022**

Tabla de autoevaluación:

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

Introducción:

Después de leer las instrucciones de la plataforma, se utilizó el programa realizado de la actividad de las listas de canciones previamente hecha, agregándole nuevas funciones para la búsqueda en la clase Lista para poder realizar el trabajo correctamente, tanto la lineal como la binaria, y con la ayuda de las plantillas pude adaptar este nuevo trabajo para cumplir los requerimientos de este trabajo. Así como agregar los operadores relacionales que se podrían utilizarse entre los objetos deseados a cada clase donde estoy manejando atributos que son como datos de la lista, también se le agrego otros métodos en la clase Canción en donde se hacen comparaciones entre los objetos recibidos para analizar los atributos deseados que nos darán valores enteros determinando si es 0, significara que son iguales. También como los atributos son cadenas, se tuvo que transformar las cadenas todo a mayúsculas y quitar sus espacios para asegurarnos que todo sea igual. Y para esta actividad se le agrego un atributo y sus respectivos setters y getters a la clase de Canción para nombrar el archivo .MP3 y sea parte de la información de la canción. Finalmente, tendrá la misma función de la lista de canciones, pero se le agrego en la clase Menú, un método privado para hacer la búsqueda, dentro de ella se puede seleccionar el método de búsqueda deseado como por canción o por interprete mediante un objeto creado para su búsqueda, y con el método linealmente o binaria, dando como salida la posición en donde se encuentra y el nombre del archivo .MP3 que tenga.

Código fuente:

```
/*Efrain Robles Pulido*/
#include "Menu.h"
#include <iostream>

using namespace std;

int main() {
    List<Cancion,3000> newList;
    Menu interfaz (newList);
    interfaz.mainMenu();
    return 0;
}

#ifndef MENU_H_INCLUDED
#define MENU_H_INCLUDED

#include "List.h"
#include "Cancion.h"

#include <iostream>
#include <conio.h>
#include <iomanip>
#include <algorithm>

class Menu {
private:
    //Atributos
    List<Cancion,3000>* myList;

    //Metodos Privados
    void newElement();
    void deleteElement();

    void showListMenu();

    void findAnElement();

    void displayMenu();

    void enterMenu();
    void clearScreen();
public:
    //Constructor
    Menu(List<Cancion,3000>&);

    //Metodo
    void mainMenu();
};

#endif // MENU_H_INCLUDED
```

```

#include "Menu.h"

using namespace std;

//Metodos privados
void Menu::newElement () {
    clearScreen();
    string myStr;
    int myInt;
    Nombre newNombre;
    Cancion newCancion;

    cout << "\n\t Agregue Cancion" << endl;
    //Datos de la cancion
    cout << "Ingrese nombre de la cancion: ";
    cin.ignore();
    getline(cin, myStr);
    newCancion.setCancion(myStr);

    //Datos del nombre del autor
    cout << "Ingrese primer nombre del autor: ";
    getline(cin, myStr);
    newNombre.setFirstName(myStr);
    cout << "Ingrese segundo nombre del autor: ";
    getline(cin, myStr);
    newNombre.setLastName(myStr);
    newCancion.setAutor(newNombre);

    //Datos del nombre del interprete
    cout << "Ingrese primer nombre del interprete: ";
    getline(cin, myStr);
    newNombre.setFirstName(myStr);
    cout << "Ingrese segundo nombre del interprete: ";
    getline(cin, myStr);
    newNombre.setLastName(myStr);
    newCancion.setInterprete(newNombre);

    //Posicion del ranking
    cout << "Ingrese posicion en el ranking: ";
    cin >> myInt;
    cin.ignore();
    newCancion.setRanking(myInt);

    //Archivo .MP3
    cout << "Ingrese el nombre del archivo MP3: ";
    getline(cin, myStr);
    myStr+= ".mp3";
    newCancion.setArchivo(myStr);

    if ((myList->getLastPos())> -1) {
        cout<< "Ingrese posicion deseada, (Ultima Posicion es "<<myList-
>getLastPos()<<"): ";
        cin>>myInt;
    }
    else {
        myInt = myList->getLastPos();
    }
}

```

```

    try {
        myList->insertElement(myInt,newCancion);
    }
    catch(ListException ex) {
        cout << "Ocurrio un error" << endl;
        cout << "Error de sistema:" << ex.what() << endl;
        enterMenu();
        return;
    }

    cout << "\t\tCancion Agregada" << endl;

}

void Menu::deleteElement() {
    int pos;
    cout<< "Ingrese la posicion de la cancion a eliminar: ";
    cin>>pos;
    myList->deleteElement(pos);
    cout<<"\t\tCancion eliminado " <<endl;
}

void Menu::displayMenu() {
    cout<< "\t\t MENU " <<endl;
    cout<< "\t[A]ñadir nuevos canciones: " <<endl;
    cout<< "\t[E]liminar cancion: " <<endl;
    cout<< "\t[M]ostror toda la lista" <<endl;
    cout<< "\t[B]usqueda" <<endl;
    cout<< "\t[S]alir " <<endl;
    cout<< "\tSelecciona una opcion (A,E,M,B,S): ";
}

void Menu::showListMenu() {
    clearScreen();
    cout << setw(13) << "POSICION | " << setw(37) << "NOMBRE DE LA CANCION
| " << setw(37) << "AUTOR |"
        << setw(38) << "INTERPRETE |" << setw(9) << "RANKING |" <<
    setw(38) << "NOMBRE DE ARCHIVO |"
        << endl;
    cout<< myList->toString() <<endl<<endl;
}

void Menu::findAnElement() {
    string myString;
    char myCharA, myCharB;
    int pos(0);

    Nombre newInterprete;
    Cancion element;

    do {
        clearScreen();
        cout<<"\tIngrese tipo de busqueda"<<endl;
        cout<<"Buscar por Nombre de la [C]ancion / Nombre del
[I]nterprete / [R]egresar al menu: ";
    }

```

```

    cin>> myCharA;
    cin.ignore();
    myCharA = toupper(myCharA);

    switch (myCharA) {
        case 'C': //Busqueda por Cancion
            cout<<endl<<"Ingrese la cancion a buscar: ";
            getline(cin, myString);
            element.setCancion(myString);

            do {
                cout<<endl<<"Ingrese metodo de busqueda deseado
[L]ineal / [B]inaria: ";
                cin>> myCharB;
                cin.ignore();
                myCharB = toupper(myCharB);

                switch(myCharB) {
                    case 'L': //Busqueda Lineal por Cancion
                        cout<<endl<<"\tBusqueda Lineal"<<endl<<endl;
                        if((pos = myList->findElementLinear(element,
Cancion::compareByCancion)) != -1) {
                            element = myList->retrieve(pos);
                            cout<<"La cancion esta en la posicion:
"<<pos<<"\t/ Archivo: "<<element.getArchivo()<<endl<<endl;
                        }
                        else {
                            cout<<"\tNo existe la cancion "<<endl;
                        }
                        break;
                    case 'B': //Busqueda Binaria por Cancion
                        cout<<endl<<"Busqueda Binaria"<<endl<<endl;
                        if((pos = myList->findElementBinary(element,
Cancion::compareByCancion)) != -1) {
                            element = myList->retrieve(pos);
                            cout<<"La cancion esta en la posicion:
"<<pos<<"\t/ Archivo: "<<element.getArchivo()<<endl<<endl;
                        }
                        else {
                            cout<<"\tNo existe la cancion "<<endl;
                        }
                        break;
                    default:
                        cout<<endl<<"ERROR, Ingrese opcion valida
"<<endl;
                        enterMenu();
                }
            }
            while((myCharB != 'L') and (myCharB != 'B'));
            break;

        case 'I': //Busqueda por Interprete
            cout<<endl<<"Ingrese primer nombre del interprete: ";
            getline(cin, myString);
            newInterprete.setFirstName(myString);
            cout<<endl<<"Ingrese segundo nombre del interprete: ";
            getline(cin, myString);

```

```

        newInterprete.setLastName(myString);

        element.setInterprete(newInterprete);

        do {
            cout<<endl<<"Ingrese metodo de busqueda deseado
[L]ineal / [B]inaria: ";
            cin>> myCharB;

            cin.ignore();
            myCharB = toupper(myCharB);

            switch(myCharB) {
                case 'L'://Busque Lineal por Interprete
                    cout<<endl<<"\tBusqueda Lineal"<<endl<<endl;
                    if((pos = myList->findElementLinear(element,
Cancion::compareByInterprete)) != -1) {
                        element = myList->retrieve(pos);
                        cout<<"La cancion esta en la posicion:
"<<pos<<"\t/ Archivo: "<<element.getArchivo()<<endl<<endl;
                    }
                    else {
                        cout<<"\tNo existe la cancion "<<endl;
                    }
                    break;
                case 'B'://Busqueda Binaria por Interprete
                    cout<<endl<<"Busqueda Binaria"<<endl<<endl;
                    if((pos = myList->findElementBinary(element,
Cancion::compareByInterprete)) != -1) {
                        element = myList->retrieve(pos);
                        cout<<"La cancion esta en la posicion:
"<<pos<<"\t/ Archivo: "<<element.getArchivo()<<endl<<endl;
                    }
                    else {
                        cout<<"\tNo existe la cancion "<<endl;
                    }
                    break;
                default:
                    cout<<endl<<"ERROR, Ingrese opcion valida
"<<endl;
                    enterMenu();
            }
        }
        while((myCharB != 'L') and (myCharB != 'B'));
        break;

    case 'R':
        break;

    default:
        cout<<endl<<"ERROR, Ingrese opcion valida "<<endl;
    }
    enterMenu();
}
while(myCharA != 'R');
}

```

```

void Menu::enterMenu() {
    cout<< endl<< "Presiona cualquier tecla para continuar";
    getch();
}

void Menu::clearScreen() {
    system("cls");
}

//Constructores
Menu::Menu(List<Cancion,3000>& _list): myList(&_list) {}

//Metodo
void Menu::mainMenu() {
    char opc, opcContinue;

    do {
        showListMenu();
        displayMenu();
        cin>> opc;
        cin.ignore();
        opc = toupper(opc);

        switch (opc) {
            case 'A':
                do {
                    showListMenu();
                    displayMenu();
                    newElement();
                    cout<< "Ingresar otra cancion [S]/[N]: ";
                    cin>> opcContinue;
                    opcContinue = toupper(opcContinue);
                    enterMenu();
                }
                while(opcContinue != 'N');
                break;

            case 'E':
                showListMenu();
                if(!(myList->isEmpty())) {
                    deleteElement();
                    showListMenu();
                }
            else {
                cout<< "\nLISTA VACIA, debe ingresar canciones"
<<endl;
            }
            enterMenu();
            break;

            case 'M':
                cout<<endl<<endl;
                showListMenu();
                enterMenu();
                break;

```



```

        case 'B':
            findAnElement();
            break;

        case 'S':
            cout << "\n\tFIN del programa" << endl;
            break;

        default:
            cout<< "ERROR, ingrese opcion valida" <<endl;
            enterMenu();
    }
}

while (opc!='S');
}

#ifdef CANCION_H_INCLUDED
#define CANCION_H_INCLUDED

#include "nombre.h"
#include <iostream>
#include <string>

class Cancion {
private:
    //Atributos
    std::string cancion;
    Nombre autor;
    Nombre interprete;
    int ranking;
    std::string archivo;

    //Metodo privado
    void copyAll(const Cancion&); //Guardara el objeto creado a una
copia

public:
    //constructor
    Cancion();
    Cancion(const Cancion&);

    //Metodos (Getters,Setters y etc)
    std::string getCancion() const;
    Nombre getAutor() const;
    Nombre getInterprete() const;
    int getRanking() const;
    std::string getArchivo() const;

    void setCancion(const std::string&);
    void setAutor(const Nombre&);
    void setInterprete(const Nombre&);
    void setRanking(const int&);
    void setArchivo(const std::string&);

    std::string toString() const;

    //Sobrecarga de operadores

```

```

        Cancion& operator = (const Cancion&);
        bool operator == (const Cancion&) const;
        bool operator != (const Cancion&) const;
        bool operator < (const Cancion&) const;
        bool operator <= (const Cancion&) const;
        bool operator > (const Cancion&) const;
        bool operator >= (const Cancion&) const;

        static int compareByCancion(const Cancion&, const Cancion&);
        static int compareByInterprete(const Cancion&, const Cancion&);
};

#endif // CANCION_H_INCLUDED

#include "Cancion.h"
#include <string>
#include <algorithm>

using namespace std;

//Metodo privado
void Cancion::copyAll(const Cancion& c) {
    Cancion = c.Cancion;
    autor = c.autor;
    interprete = c.interprete;
    ranking = c.ranking;
    archivo = c.archivo;
}

//Constructores
Cancion::Cancion() {}

Cancion::Cancion(const Cancion& n) {
    copyAll(n);
}

//Metodos (Getters, Setters y etc)
std::string Cancion::getCancion() const {
    return Cancion;
}

Nombre Cancion::getAutor() const {
    return autor;
}

Nombre Cancion::getInterprete() const {
    return interprete;
}

int Cancion::getRanking() const {
    return ranking;
}

std::string Cancion::getArchivo() const {
    return archivo;
}

```

```

    }

void Cancion::setCancion(const std::string& _cancion) {
    cancion = _cancion;
}

void Cancion::setAutor(const Nombre& _autor) {
    autor = _autor;
}

void Cancion::setInterprete(const Nombre& _interprete) {
    interprete = _interprete;
}

void Cancion::setRanking(const int& _ranking) {
    ranking = _ranking;
}

void Cancion::setArchivo(const std::string& _archivo) {
    archivo = _archivo;
}

std::string Cancion::toString() const {
    string myStr;
    char mychar[200];
    sprintf(mychar, "%35s | %35s | %35s | %6i | %35s",
    cancion.c_str(), autor.toString().c_str(), interprete.toString().c_str(),
    ranking, archivo.c_str());
    myStr += mychar;
    //c_str() es un puntero temporal cuyo contenido es una cadena que
    apunta a una matriz de caracteres
    return myStr;
}

//Sobrecarga de operadores
Cancion& Cancion::operator=(const Cancion& n) {
    copyAll(n);
    return *this;
}

bool Cancion::operator==(const Cancion& c) const {
    return cancion == c.cancion;
}

bool Cancion::operator!=(const Cancion& c) const {
    return !(*this==c);
}

bool Cancion::operator<(const Cancion& c) const {
    return cancion < c.cancion;
}

bool Cancion::operator<=(const Cancion& c) const {
    return (*this < c) or (*this==c);
}

```

```

bool Cancion::operator>(const Cancion& c) const {
    return !(*this<=c);
}

bool Cancion::operator>=(const Cancion& c) const {
    return !(*this<c);
}

int Cancion::compareByCancion(const Cancion& a, const Cancion& b) {

    string Acancion = a.getCancion();//Cadena que se recibe de cada
    elemento de la lista
    string Bcancion = b.getCancion();//Cadena que nosotros introducimos

    std::transform(Acancion.begin(), Acancion.end(),
    Acancion.begin(), ::toupper);//Pone en mayusculas todo
    Acancion.erase(std::remove(Acancion.begin(), Acancion.end(), ' '),
    Acancion.end());//Elimina todos los espacios

    std::transform(Bcancion.begin(), Bcancion.end(),
    Bcancion.begin(), ::toupper);//Pone en mayusculas todo
    Bcancion.erase(std::remove(Bcancion.begin(), Bcancion.end(), ' '),
    Bcancion.end());//Elimina todos los espacios

    return Acancion.compare(Bcancion);
}

int Cancion::compareByInterprete(const Cancion& a, const Cancion& b) {

    string Acancion = a.interprete.toString();//Cadena que se recibe de
    cada elemento de la lista
    string Bcancion = b.interprete.toString();//Cadena que nosotros
    introducimos

    std::transform(Acancion.begin(), Acancion.end(),
    Acancion.begin(), ::toupper);//Pone en mayusculas todo
    Acancion.erase(std::remove(Acancion.begin(), Acancion.end(), ' '),
    Acancion.end());//Elimina todos los espacios

    std::transform(Bcancion.begin(), Bcancion.end(),
    Bcancion.begin(), ::toupper);//Pone en mayusculas todo
    Bcancion.erase(std::remove(Bcancion.begin(), Bcancion.end(), ' '),
    Bcancion.end());//Elimina todos los espacios

    return Acancion.compare(Bcancion);
}

#ifdef NOMBRE_H_INCLUDED
#define NOMBRE_H_INCLUDED

#include <iostream>
#include <string>

class Nombre {
private:
    //Atributos
    std::string firstName;

```

```

        std::string secondName;

        //Metodo privado
        void copyAll(const Nombre&); //Guardara el objeto creado a una
copia
    public:
        //Constructores
        Nombre ();
        Nombre(const Nombre&);

        //Sobrecarga de operador
        Nombre& operator = (const Nombre&);

        //Metodos (Getters, Setters y etc)
        std::string getFirstName() const;
        std::string getLastName() const;

        void setFirstName(const std::string&);
        void setLastName(const std::string&);

        std::string toString() const;

        //Sobrecarga de operadores
        bool operator == (const Nombre&) const;
        bool operator != (const Nombre&) const;
        bool operator < (const Nombre&) const;
        bool operator <= (const Nombre&) const;
        bool operator > (const Nombre&) const;
        bool operator >= (const Nombre&) const;
};

#endif // NOMBRE_H_INCLUDED

#include "nombre.h"
using namespace std;

//Metodo privado
void Nombre::copyAll(const Nombre& n) {
    firstName = n.firstName;
    secondName = n.secondName;
}

//Constructores
Nombre::Nombre() {}

Nombre::Nombre(const Nombre& n) {
    copyAll(n);
}

//Metodos (Getters, Setters y etc)
std::string Nombre::getFirstName() const {
    return firstName;
}

std::string Nombre::getLastName() const {

```

```

        return secondName;
    }

void Nombre::setFirstName(const std::string& name_1) {
    firstName = name_1;
}

void Nombre::setLastName(const std::string& name_2) {
    secondName = name_2;
}

std::string Nombre::toString() const {
    return firstName + " " + secondName;
}

//Sobrecarga de operadores
Nombre& Nombre::operator=(const Nombre& n) {
    copyAll(n);
    return *this;
}

bool Nombre::operator==(const Nombre& n) const {
    return toString() == n.toString();
}

bool Nombre::operator!=(const Nombre& n) const {
    return !(*this == n);
}

bool Nombre::operator<(const Nombre& n) const {
    return toString() < n.toString();
}

bool Nombre::operator<=(const Nombre& n) const {
    return *this < n or *this == n;
}

bool Nombre::operator>(const Nombre& n) const {
    return !(*this <= n);
}

bool Nombre::operator>=(const Nombre& n) const {
    return !(*this < n);
}

#ifdef LIST_H_INCLUDED
#define LIST_H_INCLUDED

#include <iostream>
#include <string>
#include "ListException.h"

//Definicion
template <class T, int ARRAYSIZE = 1024>
class List {
    private:

```

```

    //Atributos
    T data[ARRAYSIZE];
    int last;

    //Metodos privados
    void copyAll(const List<T, ARRAYSIZE>&); //Guardara el objeto
    creado a una copia
    bool isValidPos(const int&); //Si esta en el rango adecuado

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

    //Sobrecarga de operador
    List<T, ARRAYSIZE>& operator = (const List<T, ARRAYSIZE>&);

    //Metodos para una Lista
    bool isEmpty();
    bool isFull();

    void insertElement(const int&, const T&);
    void deleteElement(const int&);
    void deleteAllElement();
    T retrieve(const int&);

    int getFirstPos();
    int getLastPos();
    int getPreviousPos(const int&);
    int getNextPos(const int&);

    std::string toString();

    int findElementLinear(const T&, int cmp(const T&, const T&));
    int findElementBinary(const T&, int cmp(const T&, const T&));

};

//Implementacion
using namespace std;

//Metodos privados
template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::copyAll(const List<T, ARRAYSIZE>& l) {

    for(int i(0); i <= l.last; i++) {
        data[i] = l.data[i];
    }

    last = l.last;
}

template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isValidPos(const int& p) {
    return p >= 0 && p <= last;
}

```

```

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

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

//Sobrecarga de operador
template <class T, int ARRAYSIZE>
List<T, ARRAYSIZE>& List<T, ARRAYSIZE>::operator = (const List<T,
ARRAYSIZE>& l) {
    copyAll(l);
    return *this;
}

//Metodos para una Lista
template <class T, int ARRAYSIZE>
bool List<T, ARRAYSIZE>::isEmpty() {
    return last == -1;
}

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

template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::insertElement(const int& pos, const T&
newElement) {
    if(isFull()) {
        throw ListException("Desbordamiento de datos, insertElement");
    }

    if(pos != -1 && !isValidPos(pos)) {
        throw ListException("Posicion Invalida, insertElement");
    }

    //Detectara si la posicion introducida esta dentro del rango la lista
    creada */
    if(pos > -1 && pos < last) {
        int i(last); //Insercion en el punto de interes
        while(i >= pos) {
            data[i+1] = data[i];
            i--;
        }
        data[pos] = newElement;
        last++;
    }
    else {
        //Insercion despues del punto de interes de la posicion -1 y ultima
        posicion
        data[pos + 1] = newElement;
        last++;
    }
}

```



```

    }
}

template <class T, int ARRAYSIZE>
void List<T, ARRAYSIZE>::deleteElement(const int& p) {
    if(!isValidPos(p)) {
        throw ListException("Posicion Invalida, deleteElement");
    }

    int i(p);
    while(i < last) {
        data[i] = data[i+1];
        i++;
    }
    last--;
}

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

template <class T, int ARRAYSIZE>
T List<T, ARRAYSIZE>::retrieve(const int& p) {
    if(!isValidPos(p)) {
        throw ListException("Posicion Invalida, retriive");
    }
    return data[p];
}

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

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getLastPos() {
    return last;
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getPreviousPos(const int& p) {
    if(isEmpty() or p < 1 or p > last) {
        return -1;
    }
    else {
        return p-1;
    }
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::getNextPos(const int& p) {
    if(isEmpty() or p < 0 or p > last-1) {
        return -1;
    }
}

```

```

    }
    return p+1;
}

template <class T, int ARRAYSIZE>
string List<T, ARRAYSIZE>::toString() {
    string myStr;
    char pos[50];

    if(isEmpty()) {
        sprintf(pos, "%50s", "-----EMPTY LIST-----");
    };
    myStr = pos;
}
else {
    for(int i(0); i <= last; i++) {
        sprintf(pos, "%10i |", i);
        myStr += pos + data[i].toString() + "\n";
    }
}
return myStr;
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::findElementLinear(const T& element, int cmp(const
T&, const T&)) {
    int i(0);
    while(i <= last) {
        if(cmp(data[i],element) == 0) {
            return i;
        }
        i++;
    }
    return -1;
}

template <class T, int ARRAYSIZE>
int List<T, ARRAYSIZE>::findElementBinary(const T& element, int cmp(const
T&, const T&)) {
    int i(0), j(last), m(0);
    while(i <= j) {
        m = (i + j)/2;
        if(cmp(data[m],element) == 0) {
            return m;
        }

        if (cmp(element,data[m]) == -1) {
            j = m - 1;
        }
        else {
            i = m + 1;
        }
    }
    return -1;
}

#endif // LIST_H_INCLUDED

```

```

#ifndef LISTEXCEPTION_H_INCLUDED
#define LISTEXCEPTION_H_INCLUDED

#include <exception> //via para tronar el programa
#include <string>

class ListException : public std::exception {
private:
    std::string msg;
public:
    explicit ListException(const char* message): msg(message) {}
    explicit ListException(const std::string& message): msg(message)
{}

    virtual ~ListException() throw() {}
    virtual const char* what() const throw () {
        return msg.c_str();
    }
};

#endif // LISTEXCEPTION_H_INCLUDED

```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
-----EMPTY LIST-----					

MENU

[A]adir nuevos canciones:
[E]liminar cancion:
[M]ostrar toda la lista
[B]usqueda
[S]alir

Selecciona una opcion (A,E,M,B,S):

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

Agregue Cancion

Ingrese nombre de la cancion: A Dios le pido

Ingrese primer nombre del autor: Jose

Ingrese segundo nombre del autor: Gonzalez

Ingrese primer nombre del interprete: Juanes

Ingrese segundo nombre del interprete: Vasquez

Ingrese posicion en el ranking: 2

Ingrese el nombre del archivo MP3: ADiosLePidoByJ.G

Cancion Agregada

Ingresar otra cancion [S]/[N]: s

Presiona cualquier tecla para continuar

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
6	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
7	El gran varon	Willie Colón	Neil Young	17	ElGranVaronByW.C.mp3
8	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
9	El perdón	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
10	El farsante	Leonard Cohen	Brian Holland	25	ElFarsanteByL.C.mp3
11	El día que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuierasByH.W.mp3
12	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
13	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
14	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M..mp3
15	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
16	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
17	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
18	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
19	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
20	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
21	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
22	Amor prohibido	Paul McCartney	Selena Gomez	13	AmorProhibidoByP.M.mp3
23	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
24	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
25	A Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
26	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

MENU

[A]ñadir nuevos canciones:

[E]liminar cancion:

[M]ostrar toda la lista

[B]usqueda

[S]alir

Selecciona una opcion (A,E,M,B,S): e

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Liste.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
6	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
7	El gran varon	Willie Colón	Neil Young	17	ElGranVaronByW.C.mp3
8	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
9	El perdón	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
10	El farsante	Leonard Cohen	Brian Holland	25	ElFarsanteByL.C.mp3
11	El día que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuierasByH.W.mp3
12	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
13	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
14	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M..mp3
15	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
16	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
17	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
18	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
19	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
20	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
21	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
22	Amor prohibido	Paul McCartney	Selena Gomez	13	AmorProhibidoByP.M.mp3
23	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
24	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
25	Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
26	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

Ingrese la posicion de la cancion a eliminar: 7

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
6	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
7	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
8	El perdón	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
9	El fansante	Leonard Cohen	Brian Holland	25	ElFansanteByL.C.mp3
10	El día que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuierasByH.W.mp3
11	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
12	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
13	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M.mp3
14	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
15	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
16	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
17	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
18	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
19	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
20	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
21	Amor prohibido	Paul McCartney	Selena Gomez	13	AmorProhibidoByP.M.mp3
22	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
23	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
24	Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
25	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

Presiona cualquier tecla para continuar

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
6	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
7	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
8	El perdón	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
9	El fansante	Leonard Cohen	Brian Holland	25	ElFansanteByL.C.mp3
10	El día que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuierasByH.W.mp3
11	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
12	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
13	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M.mp3
14	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
15	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
16	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
17	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
18	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
19	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
20	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
21	Amor prohibido	Paul McCartney	Selena Gomez	13	AmorProhibidoByP.M.mp3
22	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
23	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
24	A Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
25	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

MENU

[A]ñadir nuevos canciones:

[E]liminar cancion:

[M]ostrar toda la lista

[B]usqueda

[S]alir

Selecciona una opcion (A,E,M,B,S): a

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Agregue Cancion
Ingrese nombre de la cancion: Botella tras Botella
Ingrese primer nombre del autor: Alex
Ingrese segundo nombre del autor: Luna
Ingrese primer nombre del interprete: Cristian
Ingrese segundo nombre del interprete: Nodal
Ingrese posicion en el ranking: 28
Ingrese el nombre del archivo MP3: BotellaTrasBotellaByA.L
Ingrese posicion deseada, (Ultima Posicion es 25): 5
Cancion Agregada
Ingresar otra cancion [S]/[N]: n

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Botella tras Botella	Alex Luna	Cristian Nodal	28	BotellaTrasBotellaByA.L.mp3
6	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
7	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
8	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
9	El perdon	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
10	El farsante	Leonard Cohen	Brian Holland	25	ElFarsanteByL.C.mp3
11	El día que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuierasByH.W.mp3
12	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
13	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
14	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M.mp3
15	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
16	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
17	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
18	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
19	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
20	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
21	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
22	Amor prohibido	Paul McCartney	Selena Gomez	13	AmorProhibidoByP.M.mp3
23	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
24	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
25	A Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
26	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

MENU
[A]ñadir nuevos canciones:
[E]liminar cancion:
[M]ostrar toda la lista
[B]usqueda
[S]alir
Selecciona una opcion (A,E,M,B,S): b

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: c

Ingrese la cancion a buscar: Color Esperanza

Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: 1

Busqueda Lineal

La cancion esta en la posicion: 17 / Archivo: ColorEsperanzaByD.T.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: c

Ingrese la cancion a buscar: Color Esperanza

Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: b

Busqueda Binaria

No existe la cancion

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: c

Ingrese la cancion a buscar: Despacito

Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: l

Busqueda Lineal

La cancion esta en la posicion: 13 / Archivo: DespacitoByD.Y.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: c

Ingrese la cancion a buscar: Despacito

Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: b

Busqueda Binaria

La cancion esta en la posicion: 13 / Archivo: DespacitoByD.Y.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: i

Ingrese primer nombre del interprete: Mark

Ingrese segundo nombre del interprete: Antony

Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: l

Busqueda Lineal

La cancion esta en la posicion: 20 / Archivo: BesameMuchoByC.B.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
    Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: i
Ingrese primer nombre del interprete: Mark
Ingrese segundo nombre del interprete: Antony
Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: b
Busqueda Binaria
La cancion esta en la posicion: 20      /      Archivo:  BesameMuchoByC.B.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
    Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: i
Ingrese primer nombre del interprete: Willy
Ingrese segundo nombre del interprete: William
Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: l
    Busqueda Lineal
La cancion esta en la posicion: 0      /      Archivo:  MiGenteByJ.B.mp3

Presiona cualquier tecla para continuar
```

"F:\ESTRUCTURA\A5 - Busqueda\Busqueda\bin\Debug\Lista.exe"

```
    Ingrese tipo de busqueda
Buscar por Nombre de la [C]ancion / Nombre del [I]nterprete / [R]egresar al menu: i
Ingrese primer nombre del interprete: Willy
Ingrese segundo nombre del interprete: William
Ingrese metodo de busqueda deseado [L]ineal / [B]inaria: b
Busqueda Binaria
    No existe la cancion

Presiona cualquier tecla para continuar
```


POSICION	NOMBRE DE LA CANCION	AUTOR	INTERPRETE	RANKING	NOMBRE DE ARCHIVO
0	Mi gente	John Balvin	Willy William	14	MiGenteByJ.B.mp3
1	Oye Como Va	Tito Puente	Carlos Santana	26	OyeComoVaByT.P.mp3
2	La Bamba	Carlos Vives	Ritchie Valens	12	LaBambaByC.V.mp3
3	Historia de un amor	Eddie Holland	Lamont Dozier	8	HistoriaDeUnAmorByE.H.mp3
4	Guantanamera	Celia Cruz	Pete Seeger	7	GuantanameraByC.C.mp3
5	Gasolina	Mike Stoller	Daddy Yankee	6	GasolinaByM.S.mp3
6	Eres tu	Ellie Greenwich	Jeff Barry	23	EresTuByE.G.mp3
7	El gran varon	Willie Colón	Neil Young	17	ElGranVaronByW.C.mp3
8	El rey	José Jiménez	Vicente Fernández	19	ElReyByJ.J.mp3
9	El perdon	Enrique Iglesias	Nicky Jam	22	ElPerdonByE.I.mp3
10	El farsante	Leonard Cohen	Brian Holland	25	ElFarsanteByL.C.mp3
11	El dia que me quieras	Hank Williams	Bruce Springsteen	21	ElDiaQueMeQuieresByH.W.mp3
12	El buen perdedor	Brian Wilson	Franco De Vita	9	ElBuenPerdedorByB.W.mp3
13	Despacito	Daddy Yankee	Luis Fonsi	8	DespacitoByD.Y.mp3
14	Danza Kuduro	Bob Marley	Michell Omar	5	DanzaKuduroByB.M.mp3
15	Corazon partido	Stevie Wonder	Alejandro Sanz	3	CorazonPartidoByS.W.mp3
16	Conga	Gloria Estefan	Joni Mitchell	20	CongaByG.E.mp3
17	Color Esperanza	Diego Torres	Paul Simon	16	ColorEsperanzaByD.T.mp3
18	Cali Pachanguero	Mick Jagger	Keith Richards	15	CaliPachangueroByM.J.mp3
19	Burbujas de amor	Juan Luis	Smokey Robinson	18	BurbujasDeAmorByJ.L.mp3
20	Besame mucho	Chuck Berry	Mark Antony	10	BesameMuchoByC.B.mp3
21	Bailando	John Lennon	Enrique Iglesias	11	BailandoByJ.L.mp3
22	Amor prohibido	Paul McCartney	Salena Gomez	13	AmorProhibidoByP.M.mp3
23	Amigo	Bob Dylan	Roberto Carlos	8	AmigoByB.D.mp3
24	Ahora quien	John Sebastian	Marc Anthony	4	AhoraQuienByJ.S.mp3
25	Dios le pido	Jose Gonzalez	Juanes Vasquez	2	ADiosLePidoByJ.G.mp3
26	A puro dolor	Marco Luna	Son by Four	1	APuroDolorByM.L.mp3

MENU

[A]adir nuevos canciones:

[E]liminar cancion:

[M]ostrar toda la lista

[B]usqueda

[S]alir

Selecciona una opcion (A,E,M,S): s

FIN del programa

Process returned 0 (0x0) execution time : 124.735 s
Press any key to continue.

Conclusión:

Esta práctica fue sencilla al principio debido a que solo tuve que reutilizar el programa a la anterior actividad de la lista y cambie algunas formas del programa para hacerlo más eficiente como funciones para imprimir los datos con cierta cantidad de espacios, y se le agrego un atributo(archivo MP3) en la clase Canción y en las otras subclases del programa se le agrego otros métodos y sobrecargas de operadores en donde se harán las comparaciones correctamente de cada objeto, también tuve que implementar los métodos deseados como por canción o por interprete que vimos en clase como el static compareBy, también me costó un poco de trabajo utilizarlo en cada método de búsqueda para esta actividad, pero con ayuda del profesor y muchas pruebas salió la actividad, así que aún no se me queda claro de cómo funcionan, pero con más practica me podrían funcionar. Para finalmente esta actividad, tuve que ir paso por paso y estar verificando que funcione adecuadamente la actividad, además de investigar y utilizar nuevas funciones para funcionar el programa como quitar todos los espacios de cada cadena y ponerlo en mayúsculas para un correcto análisis de datos. Siento que ya puedo utilizar la programación orientada a objetos mejor que al principio debido a que he estado practicando mucho en esta clase y en las otras clases. Pero aun me queda mucho por aprender debido a que hay muchos métodos para obtener un buen funcionamiento de un programa, tomando en cuenta su eficiencia y memoria.