

UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS E INGENIERIAS



"Indices Simples"

JANETTE ARACELI CASTELLANOS BARAJAS

Ceballos Jimenez Christopher

Ingeniería Informática

Codigo: 219750442

## Main.cpp

```
#include <iostream>
```

```
#include <string.h>
```

```
#include <fstream>
```

```
#include <cstdlib>
```

```
#define TAM_LARGO 35;
```

```
using namespace std;
```

```
void pausa();
```

```
class Detalles_Lista
```

```
{
```

```
public:
```

```
    char Nombre_cancion[TAM_LARGO], Artista[TAM_LARGO], Banda[TAM_LARGO];
```

```
} Detalles;
```

```
class Lista_Reproduccion
```

```
{
```

```
public:
```

```
    char Nombre_cancion[TAM_LARGO];
```

```
    long int posicion;
```

```
} ObjINDI;
```

```
class nodo
```

```
{
```

```
public:
```

```
    char Nombre_cancion[TAM_LARGO], Artista[TAM_LARGO], Banda[TAM_LARGO];
```

```
    nodo *siguiente;
```

```
    nodo *anterior;
```

```
};
```

```
typedef nodo *pnodo;
```

```
class Lista
```

```
{
```

```
public:
```

```
    pnodo inicio,fin;
```

```
    int tam;
```

```
    void Inicializar();
```

```
    void Agregar();
```

```
    void Mostrar();
```

```
    void Modificar();
```

```
    void Eliminar();
```

```
    void Buscar();
```

```
};
```

```
void Lista::Inicializar()
```

```
{
```

```
    inicio=NULL;
```

```
    fin=NULL;
```

```
    tam=0;
```

```
}
```

```

void Lista::Agregar()
{
    ofstream DocumentoGlobal("DocPrincipal.txt",ios::app);
    ofstream DocumentoIndice("Indice.txt",ios::app);
    if(tam==0)
    {
        system("cls");
        cout << endl << endl;

        cout << "\t\t\t *****" << endl;
        cout << "\t\t\t ** DETALLES DE LA LISTA DE REPRODUCCION **" << endl;
        cout << "\t\t\t **      [AGREGAR]      *****" << endl;
        cout << "\t\t\t *****" << endl << endl
        << endl;

        pnodo nuevo=new nodo;
        fflush(stdin);
        cout<<"\n\t\t[+] Nombre: ";
        cin.getline(Detalles.Nombre_cancion,TAM_LARGO);
        cout<<"\n\t\t[+] Artista: ";
        cin.getline(Detalles.Artista,TAM_LARGO);
        cout<<"\n\t\t[+] Banda o grupo: ";
        cin.getline(Detalles.Banda,TAM_LARGO);
        strcpy(nuevo->Nombre_cancion,Detalles.Nombre_cancion);
        strcpy(nuevo->Artista,Detalles.Artista);
        strcpy(nuevo->Banda,Detalles.Banda);
        DocumentoGlobal.write((char *)&Detalles,sizeof(Detalles));
        inicio=nuevo;
    }
}

```

```

        inicio->anterior=NULL;

        fin=nuevo;

        fin->siguiente=NULL;

        strcpy(ObjINDI.Nombre_cancion,Detalles.Nombre_cancion);

        DocumentoGlobal.seekp(0,DocumentoGlobal.end);

        ObjINDI.posicion=DocumentoGlobal.tellp();

        ObjINDI.posicion=ObjINDI.posicion-sizeof(Detalles);

        DocumentoIndice.write((char *)&ObjINDI,sizeof(ObjINDI));

        tam++;
    }

    else
    {

        pnodo nuevo=new nodo;

        system("cls");

        cout << endl << endl;

        cout << "\t\t\t *****" << endl;

        cout << "\t\t\t ** DETALLES DE LA LISTA DE REPRODUCCION **" << endl;

        cout << "\t\t\t **          [AGREGAR]          **" << endl;

        cout << "\t\t\t *****" << endl << endl
    << endl;

        fflush(stdin);

        cout<<"\n\t\t[+] Nombre: ";

        cin.getline(Detalles.Nombre_cancion,TAM_LARGO);

        cout<<"\n\t\t[+] Artista: ";

        cin.getline(Detalles.Artista,TAM_LARGO);

        cout<<"\n\t\t[+] Banda o grupo: ";

```

```
cin.getline(Detalles.Banda,TAM_LARGO);
```

```
strcpy(nuevo->Nombre_cancion,Detalles.Nombre_cancion);
```

```
strcpy(nuevo->Artista,Detalles.Artista);
```

```
strcpy(nuevo->Banda,Detalles.Banda);
```

```
DocumentoGlobal.write((char *)&Detalles,sizeof(Detalles));
```

```
fin->siguiente=nuevo;
```

```
nuevo->anterior=fin;
```

```
fin=nuevo;
```

```
fin->siguiente=NULL;
```

```
strcpy(ObjINDI.Nombre_cancion,Detalles.Nombre_cancion);
```

```
DocumentoGlobal.seekp(0,DocumentoGlobal.end);
```

```
ObjINDI.posicion=DocumentoGlobal.tellp();
```

```
ObjINDI.posicion=ObjINDI.posicion-sizeof(Detalles);
```

```
DocumentoIndice.write((char *)&ObjINDI,sizeof(ObjINDI));
```

```
tam++;
```

```
}
```

```
cout << "\n\n\t\t Datos agregados con exito ...";
```

```
DocumentoGlobal.close();
```

```
DocumentoIndice.close();
```

```
}
```

```

void Lista::Mostrar()
{
    pnode actual=inicio;

    if(tam==0)
        cout<<"No hay elementos para mostrar... "<<endl;
    else
    {
        system("cls");
        cout << endl << endl;

        cout << "\t\t\t *****" << endl;
        cout << "\t\t\t ** [LISTA DE REPRODUCCION] **" << endl;
        cout << "\t\t\t *****" << endl << endl << endl;

        while(actual!=NULL)
        {
            cout << endl;

            cout << "[-]Nombre : " << actual->Nombre_cancion << endl;
            cout << "[-]Artista : " << actual->Artista << endl;
            cout << "[-]Banda o grupo: " << actual->Banda << endl;

            cout<<endl;

            actual=actual->siguiente;
        }
    }
}

```

```

void Lista::Modificar()
{
    if(tam==0)
        cout<<"No hay nada que modificar "<<endl;
    else
    {
        ofstream DocumentoAuxGlobal("AuxDocPrincipal.txt",ios::app);
        ofstream DocumentoAuxIndice("AuxDocIndice.txt",ios::app);
        ifstream LecturaDocGlobal("DocPrincipal.txt");
        ifstream LecturaDocIndice("Indice.txt");
        string comparaNombre;
        bool flag=false;
        pnode actual=inicio;
        if(!LecturaDocIndice.good())
            cout<<"No Existe el Archivo "<<endl;
        else
        {
            system("cls");
            cout << endl << endl;
            cout << "\t\t\t *****" << endl;
            cout << "\t\t\t ** MODIFICAR LISTA DE REPRODUCCION **" << endl;
            cout << "\t\t\t *****" << endl << endl;
            fflush(stdin);
            cout << "[?] Ingrese el Nombre a Modificar: ";
            fflush(stdin);
            getline(cin,comparaNombre);

```



```
while(actual!=NULL)
{
    LecturaDocIndice.read((char *)&ObjINDI,sizeof(ObjINDI));
    LecturaDocGlobal.seekg(ObjINDI.posicion);
    LecturaDocGlobal.read((char *)&Detalles,sizeof(Detalles));
    if(LecturaDocIndice.eof())
        break;
    if(comparaNombre==actual->Nombre_cancion)
    {
        flag=true;

        fflush(stdin);
        cout<<"\n\t\t[+]INGRESE Nombre: ";
        cin.getline(Detalles.Nombre_cancion,TAM_LARGO);
        cout<<"\n\t\t[+]INGRESE Artista: ";
        cin.getline(Detalles.Artista,TAM_LARGO);
        cout<<"\n\t\t[+]INGRESE Banda o grupo: ";
        cin.getline(Detalles.Banda,TAM_LARGO);

        strcpy(actual->Nombre_cancion,Detalles.Nombre_cancion);
        strcpy(actual->Artista,Detalles.Artista);
        strcpy(actual->Banda,Detalles.Banda);

        DocumentoAuxGlobal.write((char *)&Detalles,sizeof(Detalles));
        strcpy(ObjINDI.Nombre_cancion,Detalles.Nombre_cancion);
        DocumentoAuxIndice.write((char *)&ObjINDI,sizeof(ObjINDI));
    }
}
```

```

    }
    else
    {
        DocumentoAuxGlobal.write((char *)&Detalles,sizeof(Detalles));
        DocumentoAuxIndice.write((char *)&ObjINDI,sizeof(ObjINDI));
    }
    actual=actual->siguiente;
}
}
if(flag==false)
    cout<<"No se encontro ninguna Nombre con ese indice"<<endl;

DocumentoAuxGlobal.close();
DocumentoAuxIndice.close();
LecturaDocGlobal.close();
LecturaDocIndice.close();
remove("Indice.txt");
rename("AuxDocIndice.txt","Indice.txt");
remove("DocPrincipal.txt");
rename("AuxDocPrincipal.txt","DocPrincipal.txt");
}
}

void Lista::Eliminar()
{
    if(tam==0)
        cout<<"No hay nada que eliminar "<<endl;

```

```

else
{
    string comparaNombre;
    bool flag=false;
    pnode actual=inicio,sup;
    ofstream DocumentoAuxGlobal("AuxDocPrincipal.txt",ios::app);
    ofstream DocumentoAuxIndice("AuxDocIndice.txt",ios::app);
    ifstream LecturaDocGlobal("DocPrincipal.txt");
    ifstream LecturaDocIndice("Indice.txt");
    if(!LecturaDocIndice.good())
        cout<<"No Existe el Archivo "<<endl;
    else
    {
        system("cls");
        cout << endl << endl;
        cout << "\t\t\t *****" << endl;
        cout << "\t\t\t ** ELIMINAR LISTA DE REPRODUCCION **" << endl;
        cout << "\t\t\t *****" << endl << endl;
        cin.ignore();
        cout << "[?] Ingrese el Nombre para Eliminar: ";
        getline(cin,comparaNombre);
        while(actual!=NULL)
        {
            LecturaDocIndice.read((char *)&ObjINDI,sizeof(ObjINDI));
            LecturaDocGlobal.seekg(ObjINDI.posicion);
            LecturaDocGlobal.read((char *)&Detalles,sizeof(Detalles));
            if(LecturaDocIndice.eof())

```

```
        break;
    if(comparaNombre==actual->Nombre_cancion)
    {
        flag=true;
        sup=actual;
        if(actual==inicio)
        {
            if(tam==1)
            {
                cout << endl;
                cout << "[-]Nombre : " << actual->Nombre_cancion << endl;
                cout << "[-]Artista : " << actual->Artista << endl;
                cout << "[-]Banda o grupo: " << actual->Banda << endl;

                cout<<endl;
            }
            else
            {
                inicio=actual->siguiente;
                inicio->anterior=NULL;

                cout << endl;
                cout << "[-]Nombre : " << actual->Nombre_cancion << endl;
                cout << "[-]Artista : " << actual->Artista << endl;
                cout << "[-]Banda o grupo: " << actual->Banda << endl;
                cout<<endl;
            }
        }
    }
```

```
}  
else  
{  
    if(actual==fin)  
    {  
        fin=fin->anterior;  
        fin->siguiente=NULL;  
        cout << endl;  
        cout << "[-]Nombre : " << actual->Nombre_cancion << endl;  
        cout << "[-]Artista : " << actual->Artista << endl;  
        cout << "[-]Banda o grupo: " << actual->Banda << endl;  
        cout<<endl;  
    }  
    else  
    {  
        actual->siguiente->anterior=actual->anterior;  
        actual->anterior->siguiente=actual->siguiente;  
    }  
}  
delete(sup);  
tam--;  
cout<<"Eliminando Registro...\nListo"<<endl;  
  
}  
else  
{
```

```

        DocumentoAuxGlobal.write((char *)&Detalles,sizeof(Detalles));
        strcpy(ObjINDI.Nombre_cancion,actual->Nombre_cancion);
        DocumentoAuxGlobal.seekp(0,DocumentoAuxGlobal.end);
        ObjINDI.posicion=DocumentoAuxGlobal.tellp();
        ObjINDI.posicion=ObjINDI.posicion-sizeof(Detalles);
        DocumentoAuxIndice.write((char*)&ObjINDI,sizeof(ObjINDI));
    }
    actual=actual->siguiente;
}
}
if(flag==false)
    cout<<"No se encontro ninguna Nombre en el indice"<<endl;

DocumentoAuxGlobal.close();
DocumentoAuxIndice.close();
LecturaDocGlobal.close();
LecturaDocIndice.close();
remove("Indice.txt");
rename("AuxDocIndice.txt","Indice.txt");
remove("DocPrincipal.txt");
rename("AuxDocPrincipal.txt","DocPrincipal.txt");
}
}

void Lista::Buscar()
{
    if(tam==0)

```

```

        cout<<"No hay nada que Buscar"<<endl;
else
{
    string comparaNombre;

    bool flag=false;

    pnodo actual=inicio;

    system("cls");

    cout << endl << endl;

    cout << "\t\t\t *****" << endl;
    cout << "\t\t\t ** [BUSCAR LISTA DE REPRODUCCION] **" << endl;
    cout << "\t\t\t *****" << endl << endl;

    cout << "\t\t[?] Ingrese el Nombre a buscar: ";

    cin.ignore();

    getline(cin,comparaNombre);

    while(actual!=NULL)
    {
        if(comparaNombre==actual->Nombre_cancion)
        {
            flag=true;

            cout << endl;

            cout << "[-]Nombre : " << actual->Nombre_cancion << endl;

            cout << "[-]Artista : " << actual->Artista << endl;

            cout << "[-]Banda o grupo: " << actual->Banda << endl;

            cout<<endl;

        }

        actual=actual->siguiente;
    }
}

```

```

        if(flag==false)
            cout<<"No se encontro ningun Nombre con ese indice"<<endl;
    }
}

int main()
{
    char opcion[5];
    do
    {
        system("cls");
        cout << endl << endl;
        cout << "\t\t....." << endl;
        cout << "\t\t |      INDIZACION      |" << endl;
        cout << "\t\t....." << endl << endl;

        cout << "\t\t MENU LISTA DE REPRODUCCION" << endl;
        cout << "\t\t [1] CAPTURAR\n"
            << "\t\t [2] MOSTRAR\n"
            << "\t\t [3] MODIFICAR\n"
            << "\t\t [4] ELIMINAR\n"
            << "\t\t [5] BUSCAR\n"
            << "\t\t [6] SALIR" << endl;
        cout << "\t\t [/] Ingrese el numero deseado: ";
        cin>>opcion;

        if(strcmp(opcion,"1")== 0)

```



```
{  
    system("cls");  
    li.Agregar();  
    system("pause>>cls");  
}
```

```
if(strcmp(opcion,"2")== 0)  
{  
    system("cls");  
    li.Mostrar();  
    system("pause>>cls");  
}
```

```
if(strcmp(opcion,"3")== 0)  
{  
    system("cls");  
    li.Modificar();  
    system("pause>>cls");  
}
```

```
if(strcmp(opcion,"4")== 0)  
{  
    system("cls");  
    li.Eliminar();  
    system("pause>>cls");  
}
```

```
        if(strcmp(opcion,"5")== 0)
        {
            system("cls");
            li.Buscar();
            system("pause>>cls");
        }

        if(strcmp(opcion,"6")== 0)
        {
            break;
        }

    }
    while(opcion != "6");

    system("cls");
    cout << "\n\n\t\t Saliendo del sistema....";
    system("pause>>cls");
}

void pausa()
{
    system("PAUSE");
}
```

## Capturas de ejecución

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
|
|                                     INDIZACION
|-----|
|
| MENU LISTA DE REPRODUCCION
| [1] CAPTURAR
| [2] MOSTRAR
| [3] MODIFICAR
| [4] ELIMINAR
| [5] BUSCAR
| [6] SALIR
| [/] Ingrese el numero deseado: 1_
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
*****
** DETALLES DE LA LISTA DE REPRODUCCION **
** [AGREGAR] ****
*****

[+] Nombre de la cancion: La mejor solucion

[+] Artista: Memo garza

[+] Banda o grupo: Adictiva
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
|
|                                     INDIZACION
|-----|
|
| MENU LISTA DE REPRODUCCION
| [1] CAPTURAR
| [2] MOSTRAR
| [3] MODIFICAR
| [4] ELIMINAR
| [5] BUSCAR
| [6] SALIR
| [/] Ingrese el numero deseado: 2_
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
*****
** [LISTA DE REPRODUCCION] **
*****

-]Nombre de la cancion : La mejor solucion
-]Artista   : Memo garza
-]Banda o grupo: Adictiva

-]Nombre de la cancion : Quitame este miedo
-]Artista   : Mike miramontes
-]Banda o grupo: Imponente

-]Nombre de la cancion : Te presumo
-]Artista   : Yaki
-]Banda o grupo: Recodo
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
-----
|               INDIZACION               |
|-----|
```

```
MENU LISTA DE REPRODUCCION
[1] CAPTURAR
[2] MOSTRAR
[3] MODIFICAR
[4] ELIMINAR
[5] BUSCAR
[6] SALIR
[/] Ingrese el numero deseado: 3_
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
*****
** MODIFICAR LISTA DE REPRODUCCION **
*****

[?] Ingrese el Nombre a Modificar: Te presumo

[+]INGRESE Nombre de la cancion: Irreversible

[+]INGRESE Artista: Josie Cuen

[+]INGRESE Banda o grupo: Arrolladora banda el limon_
```

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
-----  
|                               |  
-----  
INDIZACION
```

MENU LISTA DE REPRODUCCION

- [1] CAPTURAR
- [2] MOSTRAR
- [3] MODIFICAR
- [4] ELIMINAR
- [5] BUSCAR
- [6] SALIR
- [/] Ingrese el numero deseado: 4\_

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
*****  
**  ELIMINAR LISTA DE REPROCUCCION  **  
*****
```

[?] Ingrese el Nombre para Eliminar: Quitame este miedo  
Eliminando Registro...  
Listo  
\_

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
-----  
|                               |  
-----  
INDIZACION
```

MENU LISTA DE REPRODUCCION

- [1] CAPTURAR
- [2] MOSTRAR
- [3] MODIFICAR
- [4] ELIMINAR
- [5] BUSCAR
- [6] SALIR
- [/] Ingrese el numero deseado: 5\_

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

```
*****  
** [BUSCAR LISTA DE REPRODUCCION] **  
*****
```

[?] Ingrese el Nombre a buscar: Irreversible

[-]Nombre de la cancion : Irreversible  
[-]Artista : Josie Cuen  
[-]Banda o grupo: Arrolladora banda el limon

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

-----  
INDIZACION

MENU LISTA DE REPRODUCCION

[1] CAPTURAR  
[2] MOSTRAR  
[3] MODIFICAR  
[4] ELIMINAR  
[5] BUSCAR  
[6] SALIR  
[/] Ingrese el numero deseado: 6

C:\Users\Admin\Downloads\Indices\bin\Debug\Indices.exe

Saliendo del sistema....  
process returned 0 (0x0) execution time : 2.410 s  
press any key to continue.