

```

1  #include <iostream>
2  #include <conio.h>
3  #include <cstdlib>
4  #include <cctype>
5  #include <cstdio>
6  #include <string>
7  #include "songs.h"
8  #include "list.h"
9  using namespace std;
10 int main() {
11     List<Songs> myLista;
12     Songs c;
13     List<Songs>::Node* pos;
14     string myStr;
15     int op,op2;
16     char opc,opc2;
17     do {
18         cout<<"###Lista de canciones###"<<endl;
19         cout << myLista.toString() << endl;
20         cout<<endl;
21         cout<<"####Bienvenido al menu del album####"<<endl<<endl;
22         cout<<"1. Insertar cancion"<<endl;
23         cout<<"2. Eliminar cancion"<<endl;
24         cout<<"3. Buscar una cancion (lineal)"<<endl;
25         cout<<"4. Otras opciones" <<endl;
26         cout<<"5. Salir"<<endl;
27         cout<<"Seleccione una opcion:";
28         cin>>op;
29         cin.get();
30         switch(op) {
31             case 1:
32                 do {
33                     cout << endl;
34                     cout << " ####INSERTAR CANCION#### " <<endl;
35                     cout << " Nombre de la cancion: " <<endl;
36                     getline(cin,myStr);
37                     c.setSongName(myStr);
38                     cout << "Nombre del interprete:" << endl;
39                     getline(cin, myStr);
40                     c.setSongAutor(myStr);
41                     cout << "Ranking de la cancion (1-10):" << endl;
42                     cin>>myStr;
43                     c.setSongRanking(myStr);
44                     try {
45                         pos = myLista.getLastPos();
46                         myLista.insertData(pos, c);
47                     }
48                     catch (List<Songs>::Exception ex) {
49                         cout << ex.what() << endl;
50                         break;
51                     }
52                     cout << endl << " cancion insertada exitosamente" << endl << endl;
53                     do {
54                         cout << " Insertar otro? (S/N): ";
55                         cin >> opc;
56                         cin.ignore();
57                         opc = toupper(opc);
58                     }
59                     while(opc != 'S' and opc != 'N');
60                 }
61                 while(opc == 'S');
62                 cout << endl << endl;
63                 system("pause");
64                 system("cls");
65                 break;
66             case 2:

```

```

67  cout << " #####ELIMINAR CANCION#### " << endl;
68  cout << "Nombre de la cancion a eliminar:" << endl;
69  getline(cin, myStr);
70  c.setSongName(myStr);
71  pos = myLista.findData(c);
72  try {
73  myLista.deleteData(pos);
74  }
75  catch (List<Songs>::Exception ex) {
76  cout << ex.what() << endl;
77  }
78  cout << "Cancion Eliminada con exito!!" << endl;
79  system("pause");
80  system("cls");
81  break;
82  case 3:
83  cout << " #####BUSCAR CANCION (LINEAL)#### " << endl;
84  cout << "Nombre de la cancion a buscar:" << endl;
85  getline(cin, myStr);
86  c.setSongName(myStr);
87  pos = myLista.findData(c);
88  cout << myLista.retrieve(pos).toString() << endl;
89  cout << endl;
90  system("pause");
91  system("cls");
92  break;
93  case 4:
94      cout << "1. Mostrar primera cancion de la lista" << endl;
95      cout << "2. Mostrar ultima cancion de la lista" << endl;
96      cout << "3. Mostrar cancion anterior" << endl;
97      cout << "4. Mostrar cancion siguiente" << endl;
98      cout << "5. Eliminar todo" << endl;
99      cout << "Seleccione una opcion" << endl;
100     cin >> op2;
101     switch (op2)
102     {
103     case 1:
104     cout << " #####PRIMERA CANCION#### " << endl;
105     cout << myLista.retrieve(myLista.getFirstPos()).toString() << endl;
106     cout << endl;
107     system("pause");
108     system("cls");
109     break;
110     case 2:
111     cout << " #####ULTIMA CANCION#### " << endl;
112     cout << myLista.retrieve(myLista.getLastPos()).toString() << endl;
113     cout << endl;
114     system("pause");
115     system("cls");
116     break;
117     case 3:
118     cout << " #####CANCION ANTERIOR#### " << endl;
119     cout << "Nombre de la cancion para mostrar su anterior:" << endl;
120     getline(cin, myStr);
121     c.setSongName(myStr);
122     pos = myLista.findData(c);
123     try {
124     cout << myLista.retrieve(myLista.getPrevPos(pos)).toString() << endl;
125     cout << endl;
126     }
127     catch (List<Songs>::Exception ex) {
128     cout << ex.what() << endl;
129     }
130     system("pause");
131     system("cls");
132     break;

```

```

133 case 4:
134 cout << " #####SIGUIENTE CANCION#### " << endl;
135 cout << "Nombre de la cancion para mostrar su siguiente:" << endl;
136 getline(cin, myStr);
137 c.setSongName(myStr);
138 pos = myLista.findData(c);
139 try {
140 cout << myLista.retrieve(myLista.getNextPos(pos)).toString() << endl;
141 }
142 catch (List<Songs>::Exception ex) {
143 cout << ex.what() << endl;
144 }
145 cout << endl;
146 system("pause");
147 system("cls");
148 break;
149
150 case 5:
151 cout << " #####ELIMINAR TODO#### " << endl;
152 do {
153 cout << " Seguro de que desea eliminar todo? (S/N): ";
154 cin >> opc2;
155 cin.ignore();
156 opc2 = toupper(opc2);
157 system("cls");
158 }
159 while(opc2 != 'S' and opc2 != 'N');
160 if(opc2 == 'S') {
161 try {
162 myLista.deleteAll();
163 }
164 catch (List<Songs>::Exception ex) {
165 cout << ex.what() << endl;
166 }
167 }
168 cout<<"SE HA ELIMINADO TODO"<<endl;
169 system("pause");
170 system("cls");
171 break;
172 default:
173 cout<<"Opcion invalida, vuelva a intentarlo!!"<<endl;
174 }
175 case 5:
176 cout<<"Nos vemos!!"<<endl;
177 break;
178 default:
179 cout<<"Opcion invalida, vuelva a intentarlo!!"<<endl;
180 break;
181 }
182 }
183 while (op!=5);
184 return 0;
185 }

```