

Evaluación Unidad 3.

Código:

```
ListaCircular.cpp x
ListaCircular.cpp > ...
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std;
4
5  struct Cancion
6  {
7      string titulo;
8      string artista;
9      Cancion *siguiente;
10 };
11
12 class ListaCircular
13 {
14 private:
15     Cancion *ultimo;
16
17 public:
18     ListaCircular() : ultimo(nullptr) {}
19
20     bool estaVacia()
21     {
22         return ultimo == nullptr;
23     }
24
25     void agregarCancion(const string &titulo, const string &artista)
26     {
27         Cancion *nueva = new Cancion{titulo, artista, nullptr};
28
29         if (estaVacia())
30         {
31             nueva->siguiente = nueva;
32
33             ultimo = nueva;
34         }
35         else
36         {
37             nueva->siguiente = ultimo->siguiente;
38             ultimo->siguiente = nueva;
39             ultimo = nueva;
40         }
41         cout << "Canción \"" << titulo << "\" agregada.\n";
42     }
43
44     void mostrarCanciones()
45     {
46         if (estaVacia())
47         {
48             cout << "La lista está vacía.\n";
49         }
50         else
51         {
52             Cancion *actual = ultimo->siguiente;
53             do
54             {
55                 cout << "Título: " << actual->titulo << ", Artista: " << actual->artista << '\n';
56                 actual = actual->siguiente;
57             }
```

```

58         } while (actual != ultimo->siguiente);
59     }
60 }
61
62 void modificarCancion(const string &tituloActual, const string &nuevoTitulo, const string &nuevoArtista)
63 {
64     if (estaVacia())
65     {
66         cout << "La lista está vacía.\n";
67     }
68     else
69     {
70         Cancion *actual = ultimo->siguiente;
71         bool encontrado = false;
72         do
73         {
74             if (actual->titulo == tituloActual)
75             {
76                 actual->titulo = nuevoTitulo;
77                 actual->artista = nuevoArtista;
78                 cout << "Canción modificada exitosamente.\n";
79                 encontrado = true;
80                 break;
81             }
82             actual = actual->siguiente;
83         } while (actual != ultimo->siguiente);

```

```

84
85         if (!encontrado)
86         {
87             cout << "Canción no encontrada.\n";
88         }
89     }
90 }
91
92 }
93
94 void eliminarCancion(const string &titulo)
95 {
96     if (estaVacia())
97     {
98         cout << "La lista está vacía.\n";
99     }
100    else
101    {
102        Cancion *actual = ultimo->siguiente;
103        Cancion *anterior = ultimo;
104        bool encontrado = false;
105
106        do
107        {
108            if (actual->titulo == titulo)

```

```

109     {
110     if (actual == ultimo)
111     {
112         if (actual->siguiente == actual)
113         {
114             ultimo = nullptr;
115         }
116         else
117         {
118             anterior->siguiente = actual->siguiente;
119             ultimo = anterior;
120         }
121     }
122     else
123     {
124         anterior->siguiente = actual->siguiente;
125         if (actual == ultimo->siguiente)
126         {
127             ultimo->siguiente = actual->siguiente;
128         }
129     }
130     delete actual;
131     cout << "Canción eliminada.\n";
132     encontrado = true;
133     break;

```

```

134     }
135     anterior = actual;
136     actual = actual->siguiente;
137 } while (actual != ultimo->siguiente);
138
139 if (!encontrado)
140 {
141     cout << "Canción no encontrada.\n";
142 }
143 }
144
145
146 }
147
148 void navegar()
149 {
150     if (estaVacia())
151     {
152         cout << "La lista está vacía.\n";
153     }
154     else
155     {
156         char opcion;
157         Cancion *actual = ultimo->siguiente;

```

```

158
159         do
160         {
161             cout << "Título: " << actual->titulo << ", Artista: " << actual->artista << '\n';
162             cout << "Presiona '1' para la siguiente canción o '2' para salir: ";
163             cin >> opcion;
164
165             if (opcion == '1')
166             {
167                 actual = actual->siguiente;
168             }
169         } while (opcion != '2');
170     }
171
172
173     }
174 };
175
176 int main()
177 {
178     ListaCircular lista;
179     int opcion;
180     string titulo, artista, tituloActual, nuevoTitulo, nuevoArtista;
181
182     do
183     {
184         cout << "\n--- Menú ---\n";
185         cout << "1. Agregar canción\n";

```

```

186         cout << "2. Mostrar canciones\n";
187         cout << "3. Modificar canción\n";
188         cout << "4. Eliminar canción\n";
189         cout << "5. Navegar entre canciones\n";
190         cout << "6. Salir\n";
191         cout << "Elige una opción: ";
192         cin >> opcion;
193         cin.ignore();
194
195         switch (opcion)
196         {
197             case 1:
198                 cout << "Ingresa el título de la canción: ";
199                 getline(cin, titulo);
200                 cout << "Ingresa el artista: ";
201                 getline(cin, artista);
202                 lista.agregarCancion(titulo, artista);
203                 break;
204
205             case 2:
206                 lista.mostrarCanciones();
207                 break;
208
209             case 3:
210                 cout << "Ingresa el título actual de la canción a modificar: ";
211                 getline(cin, tituloActual);
212                 cout << "Ingresa el nuevo título: ";
213                 getline(cin, nuevoTitulo);

```

```
214     cout << "Ingresa el nuevo artista: ";
215     getline(cin, nuevoArtista);
216     lista.modificarCancion(tituloActual, nuevoTitulo, nuevoArtista);
217     break;
218
219     case 4:
220         cout << "Ingresa el título de la canción a eliminar: ";
221         getline(cin, titulo);
222         lista.eliminarCancion(titulo);
223         break;
224
225     case 5:
226         lista.navegar();
227         break;
228
229     case 6:
230         cout << "Saliendo...\n";
231         break;
232
233     default:
234         cout << "Opción no válida.\n";
235     }
236 } while (opcion != 6);
237
238 return 0;
239 }
```

Resultados

- **Agregar canción**

```
→ PracticasU3 ./ListaCircular

--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 1
Ingresa el título de la canción: Red
Ingresa el artista: Taylor Swift
Canción "Red" agregada.

--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 1
Ingresa el título de la canción: Hasta que me olvides
Ingresa el artista: Luis Miguel
Canción "Hasta que me olvides" agregada.
```

- **Mostrar Canción**

```
--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 2
Título: Red, Artista: Taylor Swift
Título: Hasta que me olvides, Artista: Luis Miguel
```

- **Modificar canción.**

```
--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 3
Ingresa el título actual de la canción a modificar: Hasta que me olvides
Ingresa el nuevo título: Hasta Que Me Olvides
Ingresa el nuevo artista: Juan Luis Guerra
Canción modificada exitosamente.
```

- **Eliminar canción**

```
--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 4
Ingresa el título de la canción a eliminar: Red
Canción eliminada.
```

- **Navegar entre canciones**

```
--- Menú ---
1. Agregar canción
2. Mostrar canciones
3. Modificar canción
4. Eliminar canción
5. Navegar entre canciones
6. Salir
Elige una opción: 5
Título: Hasta que me olvides, Artista: Luis Miguel
Presiona '1' para la siguiente canción o '2' para salir: 1
Título: Aquel Señor, Artista: Armando Manzanero
Presiona '1' para la siguiente canción o '2' para salir: 1
Título: Red, Artista: Taylor Swift
Presiona '1' para la siguiente canción o '2' para salir: 2
```

La lista es circular porque, al realizar el registro de las canciones:

Enlace del primer nodo a sí mismo: Cuando se agrega la primera canción, se establece que el puntero siguiente de ese nodo apunta a sí mismo, formando un ciclo con un solo elemento.

El último nodo apunta al primero: Cada vez que se agrega una nueva canción, el nodo recién agregado se conecta al inicio de la lista, y luego ultimo se actualiza para que apunte al nuevo nodo, manteniendo el ciclo.