

1. Includes, determinando ofstream e ifstream, determinando a quantidade máxima de livros a serem armazenados, determinando as estruturas que serão utilizadas ao longo do código.

```
1  #include <iostream>
2  #include <windows.h>
3  #include <string.h>
4  #include <fstream>
5
6  using namespace std;
7
8  ofstream write;
9  ifstream read;
10
11 const int maxBooks = 100;
12
13 struct Book
14 {
15     int code;
16     string name;
17     string edit;
18     int yearOfPublication;
19 };
20
21 struct listOfBooks
22 {
23     int countBooks = 0;
24     Book books[maxBooks];
25 };
```

2. Declarando a lista e colocando os cabeçalhos das funções.

```
27  listOfBooks list;
28
29  int menu();
30  int menuEdit();
31
32  void insertStart(Book newBook);
33  void insertEnd(Book newBook);
34  void insertInPosition(Book newBook, int position);
35
36  void removeStart();
37  void removeEnd();
38  void removeInPosition(int position);
39
40  void editBook(int code);
41  void changePosition(int p1, int p2);
42
43  void showList();
44
45  Book readKeyboard();
46
47  void importList();
48  void exportList();
```

3. Função principal que executa um menu com base nas opções escolhidas pelo usuário

```

50 int main()
51 {
52     UINT CPAGE_UTF8 = 65001;
53     UINT CPAGE_DEFAULT = GetConsoleOutputCP();
54     SetConsoleOutputCP(CPAGE_UTF8);
55
56     int option, position, code, p1, p2;
57
58     Book book;
59
60     do
61     {
62         system("cls");
63
64         option = menu();
65
66         switch (option)
67         {
68             case 0:
69                 // 00 - Sair do programa
70                 cout << "Saindo ... ";
71                 break;
72
73             case 1:
74                 // 01 - Inserir livro no início.
75                 book = readKeyboard();
76                 cout << "Inserindo livro ... \n";
77                 insertStart(book);
78                 break;
79
80             case 2:
81                 // 02 - Inserir livro no fim.
82                 book = readKeyboard();
83                 cout << "Inserindo livro ... \n";
84                 insertEnd(book);
85                 break;

```

```

87             case 3:
88                 // 03 - Inserir livro em uma posição.
89                 book = readKeyboard();
90                 cout << "Digite a posição: ";
91                 cin >> position;
92                 cout << "Inserindo livro ... \n";
93                 insertInPosition(book, position);
94                 break;
95             case 4:
96                 // 04 - Remover livro do início.
97                 cout << "Removendo livro ... \n";
98                 removeStart();
99                 break;
100            case 5:
101                cout << "Removendo livro ... \n";
102                removeEnd();
103                // 05 - Remover livro do fim.
104                break;
105            case 6:
106                cout << "Posição do livro: ";
107                cin >> position;
108                cout << "Removendo livro ... \n";
109                removeInPosition(position);
110                // 06 - Remover livro de uma posição.
111                break;
112            case 7:
113                // 07 - Atualizar dados de um livro.
114                showList();
115
116                cout << "Código do livro: ";
117                cin >> code;
118                editBook(code);
119                break;
120            case 8:
121                // 08 - Trocar posição de um livro.
122                cout << "Digite a posição do primeiro livro: ";
123                cin >> p1;
124                cout << "Digite a posição do segundo livro: ";
125                cin >> p2;
126                changePosition(p1, p2);
127                break;
128            case 9:
129                // 09 - Importar lista de livros.
130                importList();
131                break;

```

```

132     case 10:
133         // 10 - Exportar lista de livros.
134         exportList();
135         break;
136     case 11:
137         // 11 - Mostrar lista de livros.
138         showList();
139         break;
140     default:
141         cout << "Opção inválida!\n";
142         Sleep(1000);
143         break;
144     }
145 } while (option != 0);
146
147 cout << endl
148      << endl;
149 return 0;
150 }

```

4. Função que exibe o menu e retorna a opção inserida pelo usuário.

```

152 int menu()
153 {
154     int option;
155
156     cout << "»———— MENU —————«\n\n";
157     cout << "01 - Inserir livro no início\n";
158     cout << "02 - Inserir livro no fim\n";
159     cout << "03 - Inserir livro em uma posição\n";
160     cout << "04 - Remover livro do início\n";
161     cout << "05 - Remover livro do fim\n";
162     cout << "06 - Remover livro de uma posição\n";
163     cout << "07 - Atualizar dados de um livro\n";
164     cout << "08 - Trocar posição de um livro\n";
165     cout << "09 - Importar lista de livros\n";
166     cout << "10 - Exportar lista de livros\n";
167     cout << "11 - Mostrar lista de livros\n";
168     cout << "00 - SAIR\n";
169     cout << "\n\n»————«\n\n";
170     cout << "Digite: ";
171     cin >> option;
172
173     system("cls");
174
175     return option;
176 }

```

5. Menu para edição das propriedades do livro.

```

178 int menuEdit()
179 {
180     int option;
181
182     cout << "»———— EDITAR —————«\n\n";
183     cout << "01 - Editar código\n";
184     cout << "02 - Editar nome\n";
185     cout << "03 - Editar editora\n";
186     cout << "04 - Editar ano de publicação\n";
187     cout << "00 - SAIR\n";
188     cout << "\n\n»————«\n\n";
189     cout << "Digite: ";
190     cin >> option;
191
192     system("cls");
193     return option;
194 }

```

6. Insere um livro no início da lista

```
196 void insertStart(Book newBook)
197 {
198     if (list.countBooks < maxBooks)
199     {
200         for (int i = list.countBooks; i > 0; i--)
201             list.books[i] = list.books[i - 1];
202         list.books[0] = newBook;
203         list.countBooks = list.countBooks + 1;
204         cout << "Livro inserido com sucesso!";
205         Sleep(1000);
206     }
207 }
```

7. Insere um livro no fim da lista

```
209 void insertEnd(Book newBook)
210 {
211     if (list.countBooks < maxBooks)
212     {
213         list.books[list.countBooks] = newBook;
214         list.countBooks = list.countBooks + 1;
215         cout << "Livro inserido com sucesso!";
216         Sleep(1000);
217     }
218 }
```

8. Insere um livro em uma determinada posição

```
220 void insertInPosition(Book newBook, int position)
221 {
222     if (list.countBooks < maxBooks)
223     {
224         for (int i = list.countBooks; i > 0; i--)
225             list.books[i] = list.books[i - 1];
226         list.books[position] = newBook;
227         list.countBooks = list.countBooks + 1;
228         cout << "Livro inserido com sucesso!";
229         Sleep(1000);
230     }
231 }
```

9. Remove o primeiro livro

```

233 void removeStart()
234 {
235     if (list.countBooks < maxBooks)
236     {
237         cout << "Removendo o livro: ";
238         cout << list.books[0].code << " ";
239         cout << list.books[0].name << endl;
240         Sleep(1000);
241
242         for (int i = 0; i < list.countBooks; i++)
243             list.books[i] = list.books[i + 1];
244         list.countBooks = list.countBooks - 1;
245         cout << "Livro removido com sucesso!";
246         Sleep(1000);
247     }
248 }

```

10. Remove o último livro

```

250 void removeEnd()
251 {
252     if (list.countBooks > 0)
253     {
254         cout << "Removendo o livro: ";
255         cout << list.books[list.countBooks - 1].code << " ";
256         cout << list.books[list.countBooks - 1].name << endl;
257         list.countBooks = list.countBooks - 1;
258         cout << "Livro removido com sucesso!";
259         Sleep(1000);
260     }
261 }

```

11. Remove um livro de uma determinada posição

```

263 void removeInPosition(int position)
264 {
265     if (list.countBooks > 0 && position < list.countBooks)
266     {
267         cout << "Removendo o livro: ";
268         cout << list.books[position].code << " ";
269         cout << list.books[position].name << endl;
270
271         for (int i = position; i < list.countBooks; i++)
272             list.books[i] = list.books[i - 1];
273
274         list.countBooks = list.countBooks - 1;
275         cout << "Livro removido com sucesso!";
276         Sleep(1000);
277     }
278 }

```

12. Mostra a lista de livros enquanto o usuário não quiser voltar ao menu

```

280 void showList()
281 {
282     int option;
283
284     do
285     {
286         cout << "\n»————— LISTA —————«\n\n";
287         for (int i = 0; i < list.countBooks; i++)
288         {
289             cout << list.books[i].code << " - ";
290             cout << list.books[i].name << endl;
291         }
292         cout << "\n»—————«\n\n";
293
294         cout << "Digite (1) para voltar ao menu: ";
295         cin >> option;
296     } while (option != 1);
297 }

```

13. Lê os dados do livro, utilizada para inserir os dados de um livro pelo teclado

```

299 ~ Book readKeyboard()
300 {
301     Book book;
302
303     cout << "»————— LIVRO —————«\n\n";
304     cout << "Código: ";
305     cin >> book.code;
306     cout << "Nome: ";
307     cin.ignore();
308     getline(cin, book.name);
309     cout << "Editora: ";
310     getline(cin, book.edit);
311     cout << "Ano de publicação: ";
312     cin >> book.yearOfPublication;
313
314     return book;
315 }

```

14. Edita os dados de um livro, busca feita pelo código do livro

```

317 void editBook(int code)
318 {
319     int option;
320
321     for (int i = 0; i < list.countBooks; i++)
322     {
323         if (list.books[i].code == code)
324         {
325             option = menuEdit();
326
327             do
328             {
329                 switch (option)
330                 {
331                     case 0:
332                         // sair
333                         cout << "Saindo ... ";
334                         Sleep(1000);
335                         break;
336                     case 1:
337                         // editar código
338                         cout << "Digite o código: ";
339                         cin >> list.books[i].code;
340                         break;
341                     case 2:
342                         // editar editora
343                         cout << "Digite o nome: ";
344                         cin >> list.books[i].name;
345                         break;
346                     case 3:
347                         // editar ano de publicação
348                         cout << "Digite o ano de publicação: ";
349                         cin >> list.books[i].yearOfPublication;
350                         break;
351                     default:
352                         cout << "Opção inválida!\n";
353                         Sleep(1000);
354                         system("cls");
355                         break;
356                 }
357             } while (option != 0);
358         }
359     }
360 }

```

15. Troca a posição dos livros

```

362 void changePosition(int p1, int p2)
363 {
364     Book aux;
365     aux = list.books[p1];
366     list.books[p1] = list.books[p2];
367     list.books[p2] = aux;
368 }

```

16. Importa os livros de um arquivo

```
370 void importList()
371 {
372     read.open("booksList.txt");
373
374     if (read.fail())
375         cout << "Não foi possível abrir o arquivo ... \n";
376     else
377     {
378         read >> list.countBooks;
379         for (int i = 0; i < list.countBooks; i++)
380         {
381             read >> list.books[i].code;
382             read.ignore();
383             getline(read, list.books[i].name);
384             read.ignore();
385             getline(read, list.books[i].edit);
386             read.ignore();
387             read >> list.books[i].yearOfPublication;
388         }
389
390         cout << "Importação realizada!";
391         Sleep(1000);
392         read.close();
393     }
394 }
```

17. Exporta os livros cadastrados para um arquivo

```
396 void exportList()
397 {
398     write.open("booksList.txt");
399
400     if (!write.is_open())
401         cout << "Não foi possível abrir o arquivo ... \n";
402     else
403     {
404         write.clear();
405
406         write << list.countBooks;
407         for (int i = 0; i < list.countBooks; i++)
408         {
409             write << list.books[i].code << endl;
410             write << list.books[i].name << endl;
411             write << list.books[i].edit << endl;
412             write << list.books[i].yearOfPublication << endl;
413         }
414
415         cout << "Exportação realizada!";
416         Sleep(1000);
417         write.close();
418     }
419 }
```