

```

1  /*
2  * Proyecto: ImplementacionDeUnaListaGenerica
3  * Archivo:  BibliotecaListaGenerica.h
4  * Autor:    J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 25 de septiembre de 2024, 09:27 AM
7  */
8
9  #ifndef BIBLIOTECALISTAGENERICA_H
10 #define BIBLIOTECALISTAGENERICA_H
11
12 #include <fstream>
13
14
15 void creaLista(const char*nombArch,void *&lista, void *(*lee)(ifstream&),
16               int(*compara)(const void*,const void*));
17 void insertarEnLista(void *dato, void*&lista,
18                     int(*compara)(const void*,const void*));
19 void imprimirLista(const char*nombArch,void *lista,
20                   void (*imprimeDato)(ofstream &arch, void*));
21                                     //EL error estaba aquí en el encabezado
22 #endif /* BIBLIOTECALISTAGENERICA_H */
23
24 /*
25 * Proyecto: ImplementacionDeUnaListaGenerica
26 * Archivo:  BibliotecaListaGenerica.cpp
27 * Autor:    J. Miguel Gunira E//miguel.guanira.
28 *
29 * Created on 25 de septiembre de 2024, 09:27 AM
30 */
31
32 #include <iostream>
33 #include <fstream>
34 #include <iomanip>
35 using namespace std;
36 #include "BibliotecaListaGenerica.h"
37 enum REG {DATO,SIG};
38
39 void creaLista(const char*nombArch,void *&lista, void *(*lee)(ifstream&),
40               int(*compara)(const void*,const void*)){
41     ifstream arch(nombArch,ios::in);
42     if(not arch.is_open()){
43         cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
44         exit(1);
45     }
46     void *dato;
47     lista = nullptr;
48     while(true){
49         dato = lee(arch);
50         if(arch.eof())break;
51         insertarEnLista(dato, lista, compara);
52     }
53 }
54
55 void insertarEnLista(void *dato, void*&lista,
56                     int(*compara)(const void*,const void*)){
57     void **p=(void**)lista, **ant = nullptr, **nuevo;
58
59     nuevo = new void*[2]{};
60     nuevo[DATO] = dato;
61     while(p){
62         if(compara(p[DATO],nuevo[DATO])>0)break; //compara(p[DATO],dato);
63         ant = p;
64         p = (void**)p[SIG];
65     }
66     nuevo[SIG] = p;

```

```

67     if (ant) ant[SIG] = nuevo;
68     else lista = nuevo;
69 }
70
71 void imprimirLista(const char*nombArch,void *lst,
72                  void (*imprimeDato)(ofstream &arch, void*)){
73     ofstream arch(nombArch,ios::out);
74     if(not arch.is_open()){
75         cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
76         exit(1);
77     }
78
79     void **lista = (void **)lst;
80     while(lista){
81         imprimeDato(arch,lista[DATO]);
82         lista = (void**) lista[SIG];
83     }
84 }
85
86 /*
87  * Proyecto: ImplementacionDeUnaListaGenerica
88  * Archivo:  main.cpp
89  * Autor:    J. Miguel Guanira E.//miguel.guanira.
90  *
91  * Created on 25 de septiembre de 2024, 09:23 AM
92  */
93
94 #include <iostream>
95 #include <iomanip>
96 using namespace std;
97 #include "BibliotecaListaGenerica.h"
98 #include "FuncionesDeEnteros.h"
99
100 int main(int argc, char** argv) {
101     void *lista;
102     creaLista("datos.txt",lista,leerInt,miCmpInt);
103
104     imprimirLista("reporte.txt",lista,imprimeInt);
105     // eliminarLista(lista,eliminarDato);
106     return 0;
107 }
108
109 /*
110  * Proyecto: UsoDEQsortDeCstdlib
111  * Archivo:  FuncionesDeEnteros.h
112  * Autor:    J. Miguel Guanira E. //miguel.guanira.
113  *
114  * Created on 24 de septiembre de 2024, 09:07 AM
115  */
116
117 #ifndef FUNCIONESDEENTEROS_H
118 #define FUNCIONESDEENTEROS_H
119
120 int miCmpInt(const void *, const void *);
121 void *leerInt(ifstream &arch);
122 void imprimeInt(ofstream &arch,void*d);
123
124 #endif /* FUNCIONESDEENTEROS_H */
125
126 /*
127  * Proyecto: UsoDEQsortDeCstdlib
128  * Archivo:  FuncionesDeEnteros.cpp
129  * Autor:    J. Miguel Gunira E//miguel.guanira.
130  *
131  * Created on 24 de septiembre de 2024, 09:07 AM
132  */

```

```
133
134     #include <iostream>
135     #include <fstream>
136     #include <iomanip>
137     using namespace std;
138
139     int miCmpInt(const void *a, const void *b){ //Esto cambió respecto al qsortG
140         int *ai = (int*)a, *bi= (int*)b;
141         return *ai-*bi;
142     }
143
144     void *leerInt(ifstream &arch){
145         int dato, *ptDato;
146         arch>>dato;
147         if(arch.eof())return nullptr;
148         ptDato = new int;
149         *ptDato = dato;
150         return ptDato;
151     }
152
153     void imprimeInt(ofstream &arch,void*d){
154         int *dato = (int*)d;
155         arch<<setw(5)<<*dato;
156     }
```