

```
1  /*
2  * Proyecto: ListaLigadaConClases
3  * Archivo:  Nodo.h
4  * Autor:    J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 5 de noviembre de 2024, 08:27 AM
7  */
8
9
10 #ifndef NODO_H
11 #define NODO_H
12 #include "Lista.h"
13
14 class Nodo {
15 private:
16     int dato;
17     class Nodo *sig;
18 public:
19     Nodo();
20     friend class Lista;
21 };
22
23 #endif /* NODO_H */
24
25 /*
26 * Proyecto: ListaLigadaConClases
27 * Archivo:  Nodo.cpp
28 * Autor:    J. Miguel Guanira E. //miguel.guanira.
29 *
30 * Created on 5 de noviembre de 2024, 08:27 AM
31 */
32
33 #include <iostream>
34 #include <iomanip>
35 using namespace std;
36
37 #include "Nodo.h"
38
39 Nodo::Nodo() {
40     sig = nullptr;
41 }
42
43 /*
44 * Proyecto: ListaLigadaConClases
45 * Archivo:  Lista.h
46 * Autor:    J. Miguel Guanira E. //miguel.guanira.
47 *
48 * Created on 5 de noviembre de 2024, 08:30 AM
49 */
50
51
52 #ifndef LISTA_H
53 #define LISTA_H
54 #include <fstream>
55
56 #include "Nodo.h"
57
58 class Lista {
59 private:
60     class Nodo *lista;
61     void imprimeR(ofstream &arch, class Nodo *lista);
62 public:
63     Lista();
64     virtual ~Lista();
65     void crear(const char*nombArch);
66     void insertar(int d);
```

```
67     void imprimir(const char*nombArch);
68     void elimina();
69     void imprimeRecursivo(const char*nombArch);
70 };
71
72 #endif /* LISTA_H */
73
74 /*
75  * Proyecto: ListaLigadaConClases
76  * Archivo:  Lista.cpp
77  * Autor:    J. Miguel Guanira E. //miguel.guanira.
78  *
79  * Created on 5 de noviembre de 2024, 08:30 AM
80  */
81
82 #include <iostream>
83 #include <iomanip>
84 #include <fstream>
85 using namespace std;
86
87 #include "Lista.h"
88
89 Lista::Lista() {
90     lista = nullptr;
91 }
92
93 Lista::~Lista() {
94     elimina();
95 }
96
97 void Lista::crear(const char* nombArch) {
98     ifstream arch (nombArch,ios::in);
99     if(not arch.is_open()){
100         cout<<"ERROR: no se pudo abrie el archivo "<<nombArch<<endl;
101         exit(1);
102     }
103     int d;
104     while(true){
105         arch>>d;
106         if(arch.eof())break;
107         insertar(d);
108     }
109 }
110
111 void Lista::insertar(int d) {
112     class Nodo *p=lista, *ant=nullptr, *nuevo;
113
114     nuevo = new class Nodo;
115     nuevo->dato = d;
116     while(p){
117         if(p->dato > d) break;
118         ant = p;
119         p = p->sig;
120     }
121     nuevo->sig = p;
122     if(ant != nullptr) ant->sig = nuevo;
123     else lista = nuevo;
124 }
125
126 void Lista::imprimir(const char* nombArch) {
127     ofstream arch (nombArch,ios::out);
128     if(not arch.is_open()){
129         cout<<"ERROR: no se pudo abrie el archivo "<<nombArch<<endl;
130         exit(1);
131     }
132     class Nodo *p = lista;
```

```
133     while(p) {
134         arch<<setw(4)<<p->dato;
135         p = p->sig;
136     }
137     arch<<endl;
138 }
139
140 void Lista::elimina() {
141     class Nodo *sale;
142     while(lista){
143         sale = lista;
144         lista = lista->sig;
145         delete sale;
146     }
147     //lista = nullptr; //Esto es por si se quiere luego llenar otra lista
148 }
149
150 void Lista::imprimeRecursoivo(const char* nombArch) {
151     ofstream arch (nombArch,ios::out);
152     if(not arch.is_open()){
153         cout<<"ERROR: no se pudo abrie el archivo "<<nombArch<<endl;
154         exit(1);
155     }
156     imprimeR(arch, lista);
157     arch<<endl;
158 }
159
160 void Lista::imprimeR(ofstream& arch, Nodo* lista) {
161     if(lista==nullptr) return;
162     imprimeR(arch,lista->sig);
163     arch<<setw(4)<<lista->dato;
164 }
165
166 /*
167  * Proyecto: ListaLigadaConClases
168  * Archivo:  main.cpp
169  * Autor:    J. Miguel Guanira E.//miguel.guanira.
170  *
171  * Created on 5 de noviembre de 2024, 08:26 AM
172  */
173
174 #include <iostream>
175 #include <iomanip>
176 using namespace std;
177 #include "Lista.h"
178
179 int main(int argc, char** argv) {
180     class Lista listaLigada;
181     listaLigada.crear("personal.csv");
182     listaLigada.imprimir("reporteDePersonas.txt");
183
184     listaLigada.imprimeRecursoivo("reporteRecDePersonas.txt");
185
186     return 0;
187 }
188
```