

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

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

```

133         exit(1);
134     }
135     class Nodo *p = lista;
136     while(p) {
137         arch<<p->dato;
138         p = p->sig;
139     }
140     // arch<<endl;
141 }
142
143 void Lista::elimina() {
144     class Nodo *sale;
145     while(lista){
146         sale = lista;
147         lista = lista->sig;
148         delete sale;
149     }
150     //lista = nullptr; //Esto es por si se quiere luego llenar otra lista
151 }
152
153 void Lista::imprimeRecursoivo(const char* nombArch) {
154     ofstream arch (nombArch,ios::out);
155     if(not arch.is_open()){
156         cout<<"ERROR: no se pudo abrie el archivo "<<nombArch<<endl;
157         exit(1);
158     }
159     imprimeR(arch, lista);
160     arch<<endl;
161 }
162
163 void Lista::imprimeR(ofstream& arch, Nodo* lista) {
164     if(lista==nullptr) return;
165     imprimeR(arch,lista->sig);
166     arch<<lista->dato;
167 }
168
169 /*
170 * Proyecto: ListaLigadaConClasePersona
171 * Archivo:  Persona.h
172 * Autor:    J. Miguel Guanira E. //miguel.guanira.
173 *
174 * Created on 5 de noviembre de 2024, 09:34 AM
175 */
176
177
178 #ifndef PERSONA_H
179 #define PERSONA_H
180
181 class Persona {
182 private:
183     int dni;
184     char *nombre;
185     double sueldo;
186 public:
187     Persona();
188     Persona(const Persona& orig);
189     virtual ~Persona();
190     void SetSueldo(double sueldo);
191     double GetSueldo() const;
192     void SetNombre(const char* nombre);
193     void GetNombre(char*) const;
194     void SetDni(int dni);
195     int GetDni() const;
196     void operator = (const Persona& orig);
197     bool operator >(const Persona& orig);
198 };

```

```
199
200 #endif /* PERSONA_H */
201
202 void operator>>(ifstream &, class Persona &per);
203 void operator<<(ofstream &, class Persona &per);
204
205 /*
206  * Proyecto: ListaLigadaConClasePersona
207  * Archivo: Persona.cpp
208  * Autor: J. Miguel Guanira E. //miguel.guanira.
209  *
210  * Created on 5 de noviembre de 2024, 09:34 AM
211  */
212
213 #include <iostream>
214 #include <fstream>
215 #include <iomanip>
216 using namespace std;
217 #include <cstring>
218 #include "Persona.h"
219
220 Persona::Persona() {
221     nombre = nullptr;
222 }
223
224 Persona::Persona(const Persona& orig) {
225     nombre = nullptr;
226     *this = orig;
227 }
228
229 Persona::~~Persona() {
230     if (nombre!=nullptr) delete nombre;
231 }
232
233 void Persona::SetSueldo(double sueldo) {
234     this->sueldo = sueldo;
235 }
236
237 double Persona::GetSueldo() const {
238     return sueldo;
239 }
240
241 void Persona::SetNombre(const char* nomb) {
242     if(nombre) delete nombre;
243     nombre = new char [strlen(nomb)+1];
244     strcpy(nombre,nomb);
245 }
246
247 void Persona::GetNombre(char*nomb) const {
248     if(nombre == nullptr) nomb[0] = 0;
249     else strcpy(nomb,nombre);
250 }
251
252 void Persona::SetDni(int dni) {
253     this->dni = dni;
254 }
255
256 int Persona::GetDni() const {
257     return dni;
258 }
259
260 void Persona::operator = (const Persona& orig){
261     char aux[60];
262     dni = orig.dni;
263     sueldo = orig.sueldo;
264     orig.GetNombre(aux);
```

```
265         SetNombre(aux);
266     }
267
268     bool Persona::operator>(const class Persona& orig) {
269         return dni > orig.dni;
270     }
271
272
273     void operator>>(ifstream &arch, class Persona &per){
274         int dni;
275         char nomb[60];
276         double sueldo;
277         arch>>dni;
278         if(arch.eof())return;
279         arch.get();
280         arch.getline(nomb,60,',');
281         arch>>sueldo;
282         per.SetDni(dni);
283         per.SetNombre(nomb);
284         per.SetSueldo(sueldo);
285     }
286
287     void operator<<(ofstream &arch, class Persona &per){
288         arch.precision(2);
289         arch<<fixed;
290         char cad[60];
291         per.GetNombre(cad);
292         arch<<right<<setw(10)<<per.GetDni()<<" " <<left<<setw(45)<<cad
293             <<right<<setw(10)<<per.GetSueldo()<<endl;
294     }
295
296     /*
297     * Proyecto: ListaLigadaConClases
298     * Archivo: main.cpp
299     * Autor: J. Miguel Guanira E.//miguel.guanira.
300     *
301     * Created on 5 de noviembre de 2024, 08:26 AM
302     */
303
304     #include <iostream>
305     #include <iomanip>
306     using namespace std;
307     #include "Lista.h"
308
309     int main(int argc, char** argv) {
310         class Lista listaLigada;
311         listaLigada.crear("personal.csv");
312         listaLigada.imprimir("reporteDePersonas.txt");
313
314         listaLigada.imprimeRecurso("reporteRecDePersonas.txt");
315
316
317         return 0;
318     }
```