

```

1  /*
2   * Proyecto: ImplementacionDeUnaListaGenerica
3   * Archivo:  main.cpp
4   * Autor:    J. Miguel Guanira Erazo (Juan Miguel)
5   * Creado el 6 de junio de 2020, 05:57 PM
6   */
7  #include <iostream>
8  #include <fstream>
9  using namespace std;
10 #include "BibliotecaListaGenerica.h"
11 #include "ListaConEnteros.h"
12 #include "ListaConCadenas.h"
13 #include "ListaConRegistroVoid.h"
14 int main(int argc, char** argv) {
15     void *lista;
16     ///*****
17     ///    LISTA CON ENTEROS
18     ///*****
19     //    ifstream arch("datos.txt",ios::in);
20     //    if(!arch){
21     //        cout<<"ERROR: No se pudo abrir el archivo datos.txt"<<endl;
22     //        exit(1);
23     //    }
24     //    crearLista(arch,lista,leeEntero,cmpEntero);
25     //    imprimirLista(lista,imprimeEntero);
26     //    eliminarLista(lista,eliminaEntero);
27     {}
28     ///*****
29     ///    LISTA CON NOMBRES
30     ///*****
31     //    ifstream arch("personas.txt",ios::in);
32     //    if(!arch){
33     //        cout<<"ERROR: No se pudo abrir el archivo personas.txt"<<endl;
34     //        exit(1);
35     //    }
36     //    crearLista(arch, lista, leeNombre, cmpCadena);
37     //    imprimirLista(lista,imprimeNombre);
38     //    eliminarLista(lista,eliminaNombre);
39     {}
40     ///*****
41     //    LISTA CON REGISTROS
42     ifstream arch("personal.csv",ios::in);
43     if(!arch){
44         cout<<"ERROR: No se pudo abrir el archivo personal.csv"<<endl;
45         exit(1);
46     }
47     //    crearLista(arch,lista,leeReg,cmpRegCodigo);
48     //    crearLista(arch,lista,leeReg,cmpRegNombre);
49     crearLista(arch,lista,leeReg,cmpRegSueldo);
50     imprimirLista(lista,imprimeReg);
51     eliminarLista(lista,eliminaReg);
52     return 0;
53 }

```

**PUNTEROS A FUNCIONES
IMPLEMENTACIÓN DE UNA
LISTA GENÉRICA**

```

54  /*
55  * Proyecto:  ImplementacionDeUnaListaGenerica
56  * Archivo:  BibliotecaListaGenerica.h
57  * Autor:    J. Miguel Guanira Erazo (Juan Miguel)
58  *
59  * Creado el 6 de junio de 2020, 05:58 PM
60  */
61  #ifndef BIBLIOTECALISTAGENERICA_H
62  #define BIBLIOTECALISTAGENERICA_H
63  #include <fstream>
64  using namespace std;
65
66  void crearLista(ifstream &,void *&,void*(*)(ifstream &),
67                int(*)(const void*,const void*));
68  void imprimirLista(void *,void(*)(void*));
69  void eliminarLista(void *,void(*)(void*));
70  void insertarLista(void *&, void *,
71                    int(*)(const void*,const void*));
72
73  #endif /* BIBLIOTECALISTAGENERICA_H */
74
75  /*
76  * Proyecto:  ImplementacionDeUnaListaGenerica
77  * Archivo:  BibliotecaListaGenerica.cpp
78  * Autor:    J. Miguel Guanira Erazo (Juan Miguel)
79  *
80  * Creado el 6 de junio de 2020, 05:58 PM
81  */
82  #include <iostream>
83  #include <fstream>
84  #include <iomanip>
85  using namespace std;
86  #include "BibliotecaListaGenerica.h"
87  enum Reg {DATO,SIG};
88
89  void crearLista(ifstream &arch, void *&lista, void*(*lee)(ifstream &),
90                int(*comp)(const void*,const void*)){
91      void *dato;
92      lista = nullptr;
93      while(1){
94          dato = lee(arch);
95          if(dato == nullptr) break;
96          insertarLista(lista, dato, comp);
97      }
98  }
99
100 void imprimirLista(void *lista, void(*imprime)(void*)){
101     void **ls=(void**)lista;
102     while(ls){
103         imprime(ls[DATO]);
104         ls = (void**)(ls[SIG]);
105     }
106 }

```

```

107 void insertarLista(void *&lista, void *dato,
108                  int(*comp)(const void*,const void*)){
109     void **p = (void **)lista, **ant=NULLPTR, **nuevo;
110     nuevo = new void*[2];
111     nuevo[DATO] = dato;
112     while(p){
113         if(comp(p[DATO], dato)>0) break;
114         ant = p;
115         p = (void**)(p[SIG]);
116     }
117     nuevo[SIG] = p;
118     if(ant == NULLPTR) lista = nuevo;
119     else ant[SIG] = nuevo;
120 }
121
122 void eliminarLista(void *lista,void(*elimina)(void *)){
123     void **ls=(void**)lista, **sale;
124     while(ls){
125         sale = ls;
126         ls = (void**)(ls[SIG]);
127         elimina(sale[DATO]);
128         delete sale;
129     }
130 }
131
132 /*
133  * Proyecto: ImplementacionDeUnaListaGenerica
134  * Archivo: ListaConEnteros.h
135  * Autor: J. Miguel Guanira Erazo (Juan Miguel)
136  *
137  * Creado el 6 de junio de 2020, 07:41 PM
138  */
139
140 #ifndef LISTACONENTEROS_H
141 #define LISTACONENTEROS_H
142 #include <fstream>
143 using namespace std;
144
145 void* leeEntero(istream &);
146 int cmpEntero(const void*,const void*);
147 void imprimeEntero(void*);
148 void eliminaEntero(void *);
149
150 #endif /* LISTACONENTEROS_H */
151
152 /*
153  * Proyecto: ImplementacionDeUnaListaGenerica
154  * Archivo: ListaConEnteros.cpp
155  * Autor: J. Miguel Guanira Erazo (Juan Miguel)
156  *
157  * Creado el 6 de junio de 2020, 07:41 PM
158  */
159

```

```

160  #include <iostream>
161  #include <fstream>
162  #include <iomanip>
163  using namespace std;
164
165  void* leeEntero(ifstream &arch){
166      int dato, *d;
167      arch>>dato;
168      if(arch.eof()) return nullptr;
169      d = new int;
170      *d = dato;
171      return d;
172  }
173
174  int cmpEntero(const void*d1,const void*d2){
175      int *dato1 = (int*)d1, *dato2=(int*)d2;
176      return *dato1-*dato2;
177  }
178
179  void imprimeEntero(void*d){
180      int *dato = (int*)d;
181      cout<<setw(4)<<*dato<<endl;
182  }
183
184  void eliminaEntero(void *d){
185      int *dato = (int*)d;
186      delete dato;
187  }
188
189  /*
190   * Proyecto:  ImplementacionDeUnaListaGenerica
191   * Archivo: ListaConCadenas.h
192   * Autor:    J. Miguel Guanira Erazo (Juan Miguel)
193   *
194   * Creado el 6 de junio de 2020, 08:21 PM
195   */
196  #ifndef LISTACONCADENAS_H
197  #define LISTACONCADENAS_H
198
199  void* leeNombre(ifstream&);
200  int cmpCadena(const void*,const void*);
201  void imprimeNombre(void*);
202  void eliminaNombre(void *);
203
204  #endif /* LISTACONCADENAS_H */
205
206  /*
207   * Proyecto:  ImplementacionDeUnaListaGenerica
208   * Archivo: ListaConRegistroVoid.h
209   * Autor:    J. Miguel Guanira Erazo (Juan Miguel)
210   *
211   * Creado el 6 de junio de 2020, 08:39 PM
212   */

```

```

213 #ifndef LISTACONREGISTROVOID_H
214 #define LISTACONREGISTROVOID_H
215 #include <fstream>
216 using namespace std;
217
218 void *leeReg(ifstream&);
219 int cmpRegCodigo(const void *, const void *);
220 int cmpRegNombre(const void *, const void *);
221 int cmpRegSueldo(const void *, const void *);
222 void imprimeReg(void*);
223 void eliminaReg(void*);
224
225 #endif /* LISTACONREGISTROVOID_H */
226
227 /*
228  * Proyecto: ImplementacionDeUnaListaGenerica
229  * Archivo: ListaConRegistroVoid.cpp
230  * Autor: J. Miguel Guanira Erazo (Juan Miguel)
231  *
232  * Creado el 6 de junio de 2020, 08:39 PM
233  */
234 #include <iostream>
235 #include <fstream>
236 #include <iomanip>
237 using namespace std;
238 #include <cstring>
239 enum Reg {CODIGO, NOMBRE, SUELDO};
240
241 void *leeReg(ifstream &arch){
242     int *codigo, cod;
243     char*nombre, buff[100];
244     double * sueldo;
245     void **registro;
246
247     arch>>cod;
248     if(arch.eof()) return nullptr;
249     codigo = new int;
250     *codigo = cod;
251     arch.get(); //Sacamos la coma
252
253     arch.getline(buff,100,',');
254     nombre = new char[strlen(buff)+1];
255     strcpy(nombre, buff);
256
257     sueldo = new double;
258     arch>>*sueldo;
259
260     registro = new void*[3];
261     registro[CODIGO] = codigo;
262     registro[NOMBRE] = nombre;
263     registro[SUELDO] = sueldo;
264     return registro;
265 }

```

```

266     int cmpRegCodigo(const void *perI, const void *perJ){
267         void **pI = (void**)perI, **pJ = (void**)perJ;
268         int *cI = (int *)pI[CODIGO], *cJ = (int *)pJ[CODIGO];
269         return *cI - *cJ;
270     }
271
272     int cmpRegNombre(const void *perI, const void *perJ){
273         void **pI = (void**)perI, **pJ = (void**)perJ;
274         char *nI = (char*)pI[NOMBRE], *nJ = (char*)pJ[NOMBRE] ;
275         return strcmp(nI,nJ);
276     }
277
278     int cmpRegSueldo(const void *perI, const void *perJ){
279         void **pI = (void**)perI, **pJ = (void**)perJ;
280         double *sI = (double *)pI[SUELDO], *sJ = (double *)pJ[SUELDO];
281         return *sJ - *sI;
282     }
283
284     void imprimeReg(void*reg){
285         void **registro = (void**)reg;
286         int *codigo = (int*)(registro[CODIGO]);
287         char *nombre = (char*)(registro[NOMBRE]);
288         double *sueldo = (double*)(registro[SUELDO]);
289
290         cout.precision(2);
291         cout<<fixed;
292         cout<<left<<setw(10)<<*codigo
293             <<setw(50)<<nombre
294             <<right<<setw(10)<<*sueldo<<endl;
295     }
296
297     void eliminaReg(void*reg){
298         void **registro = (void**)reg;
299         int *codigo = (int*)(registro[CODIGO]);
300         char *nombre = (char*)(registro[NOMBRE]);
301         double *sueldo = (double*)(registro[SUELDO]);
302         delete codigo;
303         delete nombre;
304         delete sueldo;
305         delete registro;
306     }
307

```