```cpp
/*
 * Proyecto: ImplementacionDeUnaListaGenerica
 * Archivo:  main.cpp
 * Autor:    J. Miguel Guanira E.//miguel.guanira.
 *
 * Created on 25 de septiembre de 2024, 09:23 AM
 */

#include <iostream>
#include <iomanip>
using namespace std;
#include "BibliotecaListaGenerica.h"
#include "FunciionesDeEnteros.h"
#include "FuncionesDeCadenas.h"
#include "FuncionesDePunterosGenericos.h"

int main(int argc, char** argv) {
    void *lista;
//    creaLista("datos.txt",lista,leerInt,miCmpInt);
//
//    imprimirLista("reporte.txt",lista,imprimeInt);
//    eliminarLista(lista,eliminaInt);


//    creaLista("personas.txt",lista,leeCadena,miStrcmp);
//    imprimirLista("reporte.txt",lista,imprimeStr);
//    eliminarLista(lista,eliminaStr);

//    creaLista("personas.txt",lista,leeCadena,miStrcmp);
//    imprimirLista("reporte.txt",lista,imprimeStr);
//    eliminarLista(lista,eliminaStr);

//    creaLista("personal.csv",lista,leerRegistro,miVoidcmpNombres);
//    imprimirLista("reporteNom.txt",lista,imprimePersona);
//    eliminarLista(lista,eliminaReg);

    creaLista("personal.csv",lista,leerRegistro,miVoidcmpSueldos);
    imprimirLista("reporteSueldo.txt",lista,imprimePersona);
    eliminarLista(lista,eliminaReg);

    return 0;
}

/*
 * Proyecto: ImplementacionDeUnaListaGenerica
 * Archivo:  BibliotecaListaGenerica.h
 * Autor:    J. Miguel Guanira E. //miguel.guanira.
 *
 * Created on 25 de septiembre de 2024, 09:27 AM
 */

#ifndef BIBLIOTECALISTAGENERICA_H
#define BIBLIOTECALISTAGENERICA_H

#include <fstream>


void creaLista(const char*nombArch,void *&lista, void *(*lee)(ifstream&),
               int(*compara)(const void*,const void*));
void insertarEnLista(void *dato, void*&lista,
                     int(*compara)(const void*,const void*));
void imprimirLista(const char*nombArch,void *lista,
                                  void (*imprimeDato)(ofstream &arch, void*));
void eliminarLista(void *lista, void (*eliminarDato)(void*));

#endif /* BIBLIOTECALISTAGENERICA_H */
```

```cpp

/*
 * Proyecto: ImplementacionDeUnaListaGenerica
 * Archivo:  BibliotecaListaGenerica.cpp
 * Autor:    J. Miguel Gunira E//miguel.guanira.
 *
 * Created on 25 de septiembre de 2024, 09:27 AM
 */

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
#include "BibliotecaListaGenerica.h"
enum REG {DATO,SIG};

void creaLista(const char*nombArch,void *&lista, void *(*lee)(ifstream&),
               int(*compara)(const void*,const void*)){
    ifstream arch(nombArch,ios::in);
    if(not arch.is_open()){
        cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
        exit(1);
    }
    void *dato;
    lista = nullptr;
    while(true){
        dato = lee(arch);
        if(arch.eof())break;
        insertarEnLista(dato, lista, compara);
    }
}


void insertarEnLista(void *dato, void*&lista,
                     int(*compara)(const void*,const void*)){
    void **p=(void**)lista, **ant = nullptr, **nuevo;

    nuevo = new void*[2]{};
    nuevo[DATO] = dato;
    while(p){
        if(compara(p[DATO],nuevo[DATO])>0)break; //compara(p[DATO],dato);
        ant = p;
        p = (void**)p[SIG];
    }
    nuevo[SIG] = p;
    if(ant) ant[SIG] = nuevo;
    else lista = nuevo;
}

void imprimirLista(const char*nombArch,void *lst,
                           void (*imprimeDato)(ofstream &arch, void*)){
    ofstream arch(nombArch,ios::out);
    if(not arch.is_open()){
        cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
        exit(1);
    }

    void **lista = (void **)lst;
    while(lista){
        imprimeDato(arch,lista[DATO]);
        lista = (void**) lista[SIG];
    }
}

void eliminarLista(void *lst, void (*eliminarDato)(void*)){
    void **lista = (void **)lst, **sale;
```

```
133         while(lista){
134             sale = lista;
135             eliminarDato(sale[DATO]);
136             lista = (void **)lista[SIG];
137             delete sale;
138         }
139     }
140 
141     /*
142      * Proyecto: UsoDEQsortDeCstdlib
143      * Archivo:  FunciionesDeEnteros.h
144      * Autor:    J. Miguel Guanira E. //miguel.guanira.
145      *
146      * Created on 24 de septiembre de 2024, 09:07 AM
147      */
148 
149     #ifndef FUNCIIONESDEENTEROS_H
150     #define FUNCIIONESDEENTEROS_H
151 
152     int miCmpInt(const void *, const void *);
153     void *leerInt(ifstream &arch);
154     void imprimeInt(ofstream &arch,void*d);
155     void eliminaInt(void *d);
156 
157     #endif /* FUNCIIONESDEENTEROS_H */
158 
159     /*
160      * Proyecto: UsoDEQsortDeCstdlib
161      * Archivo:  FunciionesDeEnteros.cpp
162      * Autor:    J. Miguel Gunira E//miguel.guanira.
163      *
164      * Created on 24 de septiembre de 2024, 09:07 AM
165      */
166 
167     #include <iostream>
168     #include <fstream>
169     #include <iomanip>
170     using namespace std;
171 
172     int miCmpInt(const void *a, const void *b){ //Esto cambió respecto al qsortG
173         int *ai = (int*)a, *bi= (int*)b;
174         return *ai-*bi;
175     }
176 
177     void *leerInt(ifstream &arch){
178         int dato, *ptDato;
179         arch>>dato;
180         if(arch.eof())return nullptr;
181         ptDato = new int;
182         *ptDato = dato;
183         return ptDato;
184     }
185 
186     void imprimeInt(ofstream &arch,void*d){
187         int *dato = (int*)d;
188         arch<<setw(5)<<*dato;
189     }
190 
191     void eliminaInt(void *d){
192         int * dato = (int*)d;
193         delete dato;
194     }
195 
196     /*
197      * Proyecto: UsoDEQsortDeCstdlib
198      * Archivo:  FuncionesDeCadenas.h
```

```
199     * Autor:    J. Miguel Guanira E. //miguel.guanira.
200     *
201     * Created on 24 de septiembre de 2024, 09:13 AM
202     */
203
204    #ifndef FUNCIONESDECADENAS_H
205    #define FUNCIONESDECADENAS_H
206
207    void*leeCadena(ifstream &arch);
208    int miStrcmp(const void*, const void *);
209    void imprimeStr(ofstream &arch, void*);
210    void eliminaStr(void*dato);
211
212    #endif /* FUNCIONESDECADENAS_H */
213
214    /*
215     * Proyecto: UsoDEQsortDeCstdlib
216     * Archivo:  FuncionesDeCadenas.cpp
217     * Autor:    J. Miguel Gunira E//miguel.guanira.
218     *
219     * Created on 24 de septiembre de 2024, 09:13 AM
220     */
221
222    #include <iostream>
223    #include <fstream>
224    #include <iomanip>
225    using namespace std;
226    #include <cstring>
227    #include "FuncionesDeCadenas.h"
228
229    void*leeCadena(ifstream &arch){
230        char buffer[60], *cad;
231        arch.getline(buffer,60);
232        cad = new char[strlen(buffer)+1];
233        strcpy(cad, buffer);
234        return cad;
235    }
236
237    int miStrcmp(const void*a, const void *b){
238        char *ai = (char*)a,*bi = (char*)b;
239        return strcmp(ai,bi);
240    }
241
242    void imprimeStr(ofstream &arch, void*dato){
243        char * cadena = (char*)dato;
244        arch<<cadena<<endl;
245    }
246
247    void eliminaStr(void*dato){
248        char *cadena =(char*)dato;
249        delete cadena;
250    }
251
252    /*
253     * Proyecto: UsoDEQsortDeCstdlib
254     * Archivo:  FuncionesDePunterosGenericos.h
255     * Autor:    J. Miguel Guanira E. //miguel.guanira.
256     *
257     * Created on 24 de septiembre de 2024, 09:34 AM
258     */
259
260    #ifndef FUNCIONESDEPUNTEROSGENERICOS_H
261    #define FUNCIONESDEPUNTEROSGENERICOS_H
262    void *leerRegistro(ifstream &arch);
263    void imprimirDatos(void*per, int np, const char *nombArch);
264    void imprimePersona(ofstream &arch,void *per);
```

```
265  int miVoidcmpCodigos(const void *a, const void *b);
266  int miVoidcmpNombres(const void *a, const void *b);
267  int miVoidcmpSueldos(const void *a, const void *b);
268  void eliminaReg(void *dato);
269  char*leeCadena(ifstream &arch, char delimitador='\n');
270
271  #endif /* FUNCIONESDEPUNTEROSGENERICOS_H */
272
273  /*
274   * Proyecto: UsoDEQsortDeCstdlib
275   * Archivo:  FuncionesDePunterosGenericos.cpp
276   * Autor:    J. Miguel Gunira E//miguel.guanira.
277   *
278   * Created on 24 de septiembre de 2024, 09:34 AM
279   */
280
281  #include <iostream>
282  #include <fstream>
283  #include <iomanip>
284  using namespace std;
285  #include <cstring>
286  #include "FuncionesDeCadenas.h"
287  #include "FuncionesDePunterosGenericos.h"
288  #define INCREMENTO 5
289
290
291  void *leerRegistro(ifstream &arch){
292      void **registro;
293      int *codigo, cod;
294      char *nombre;
295      double *sueldo;
296      arch >> cod;
297      if(arch.eof()) return nullptr;
298      codigo = new int;
299      *codigo = cod;
300      arch.get();
301      nombre = leeCadena(arch,',');
302      sueldo = new double;
303      arch>>*sueldo;
304      registro = new void*[3]{};
305      registro[0] = codigo;
306      registro[1] = nombre;
307      registro[2] = sueldo;
308      return registro;
309  }
310
311  void imprimirDatos(void*per, int np, const char *nombArch){
312      ofstream arch(nombArch,ios::out);
313      if(not arch.is_open()){
314          cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
315          exit(1);
316      }
317      void **personal = (void **)per;
318      for (int i = 0; i<np-1 ; i++)
319          imprimePersona(arch,personal[i]);
320  }
321
322  void imprimePersona(ofstream &arch,void *per){
323      void **persona = (void **)per;
324      int *codigo = (int *)persona[0];
325      char*nombre = (char*)persona[1];
326      double *sueldo = (double*)persona[2];
327
328      arch.precision(2);
329      arch<<fixed;
330      arch<<right<<setw(10)<<*codigo<<"  "<<left<<setw(40)<<nombre<<right
```

```
331             <<setw(10)<<*sueldo <<endl;
332
333     }
334
335     int miVoidcmpCodigos(const void *a, const void *b){
336         void **regA = (void **)(a), **regB = (void**)(b);
337         int *codA = (int*)regA[0], *codB=(int*)regB[0];
338         return *codA-*codB;
339     }
340
341     int miVoidcmpSueldos(const void *a, const void *b){
342         void **regA = (void **)(a), **regB = (void**)(b);
343         double *codA = (double*)regA[2], *codB=(double*)regB[2];
344         return (int)(*codA-*codB);
345     }
346
347     int miVoidcmpNombres(const void *a, const void *b){
348         void **regA = (void **)(a), **regB = (void**)(b);
349         char *nombA = (char*)regA[1], *nombB=(char*)regB[1];
350         return strcmp(nombA,nombB);
351     }
352
353     void eliminaReg(void *dato){
354         void **reg = (void **)dato;
355         int *codigo = (int*)reg[0];
356         char *nombre = (char*)reg[1];
357         double *sueldo = (double*)reg[2];
358
359         delete codigo;
360         delete nombre;
361         delete sueldo;
362         delete reg;
363     }
364
365     char*leeCadena(ifstream &arch, char delimitador){
366         char buffer[60], *cad;
367         arch.getline(buffer,60,delimitador);
368         cad = new char[strlen(buffer)+1];
369         strcpy(cad, buffer);
370         return cad;
371     }
```