```cpp
/*
 * Proyecto: PunterosGenericos-Aplicacion
 * Archivo:  main.cpp
 * Autor:    J. Miguel Guanira E.//miguel.guanira.
 *
 * Created on 9 de abril de 2024, 09:45 AM
 */

#include <iostream>
#include <iomanip>
using namespace std;
#include "FuncionesAuxiliares.h"

int main(int argc, char** argv) {
    void *alumnos;

    cargarAlumnos(alumnos,"Alumnos.csv");
    cargarNotas(alumnos,"CursosNotas.csv");

    probarCarga(alumnos);

    return 0;
}

/*
 * Proyecto: PunterosGenericos-Aplicacion
 * Archivo:  FuncionesAuxiliares.h
 * Autor:    J. Miguel Guanira E. //miguel.guanira.
 *
 * Created on 9 de abril de 2024, 09:47 AM
 */

#ifndef FUNCIONESAUXILIARES_H
#define FUNCIONESAUXILIARES_H

void cargarAlumnos(void *&alumnos,const char*nombArch);
void probarCarga(void *alumnos);
void *leerRegistro(ifstream &arch);
char *leeCadena(ifstream &arch, char='\n');
void incrementarEspacios(void **&alumnos,int &nd, int &cap);
void imprimirAlumno(void *al);
void cargarNotas(void *al,const char*nombArch);
int buscar(int cod,void **alumnos);
bool sonIguales(int cod,void*al);
void colocarCurso(ifstream &arch,void *al,int &nd,int &cap);
void incrementarEspacios(void *&al,int &nd, int &cap);
void *leeRCurso(ifstream &arch);
void agregarCurso(void*cur,void *alNota, int nd);
void imprimirCursos(void *cur);
void imprimirCurso(void *cur);

#endif /* FUNCIONESAUXILIARES_H */

/*
 * Proyecto: PunterosGenericos-Aplicacion
 * Archivo:  FuncionesAuxiliares.cpp
 * Autor:    J. Miguel Gunira E//miguel.guanira.
 *
 * Created on 9 de abril de 2024, 09:47 AM
 */

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
#include <cstring>
```

```cpp
#include "FuncionesAuxiliares.h"
#define INCREMENTO 5
enum Reg {COD,NOM,NOT,PRO};

void cargarAlumnos(void *&al,const char*nombArch){
    ifstream arch(nombArch,ios::in);
    if(not arch.is_open()){
        cout<<"El archivo "<<nombArch<<"no se abrio"<<endl;
        exit(1);
    }
    void **alumnos, *reg;
    int nd=0, cap=0;
    alumnos = nullptr;
    while(true){
        reg = leerRegistro(arch);
        if(arch.eof())break;
        if(nd==cap) incrementarEspacios(alumnos,nd,cap);
        alumnos[nd-1] = reg;
        nd++;
    }
    al = alumnos;
}

void *leerRegistro(ifstream &arch){
    void**registro;
    int *codigo, cod;
    char*nombre;

    arch>>cod;
    if(arch.eof()) return nullptr;
    codigo = new int;
    *codigo = cod;
    arch.get();
    nombre = leeCadena(arch);
    registro = new void*[4]{};
    registro[COD]=codigo;
    registro[NOM]=nombre;
//    registro[NOT]=nullptr;
//    registro[PRO]=nullptr;
    return registro;
}

char *leeCadena(ifstream &arch, char car){
    char cadena[60], *cad;
    arch.getline(cadena,60,car);
    cad = new char [strlen(cadena)+1];
    strcpy(cad,cadena);
    return cad;
}

void incrementarEspacios(void **&alumnos,int &nd, int &cap){
    void **aux;

    cap += INCREMENTO;
    if (alumnos == nullptr){
        alumnos = new void*[cap]{}; // No olvidar las {}
        nd = 1;
    }
    else{
        aux = new void* [cap]{};
        for (int i=0; i<nd; i++)
            aux[i] = alumnos[i];
        delete alumnos;
        alumnos = aux;
    }
}
```

```
133
134
135 void probarCarga(void *al){
136     void **alumnos = (void **)al;
137     for(int i=0; alumnos[i]; i++)
138         imprimirAlumno(alumnos[i]);
139 }
140
141 void imprimirAlumno(void *al){
142     void **alumno = (void**)al;
143     int *codigo;
144     char *nombre;
145
146     codigo = (int*)(alumno[COD]);
147     nombre = (char*)(alumno[NOM]);
148     cout<<left<<setw(10)<<*codigo<<setw(50)<<nombre<<endl;
149     if(alumno[NOT])
150         imprimirCursos(alumno[NOT]);
151 }
152
153 void imprimirCursos(void *cur){
154     void **cursos = (void**)cur;
155     for(int i=0; cursos[i]; i++)
156         imprimirCurso(cursos[i]);
157 }
158
159 void imprimirCurso(void *cur){
160     void **curso= (void**)cur;
161     char*codigo= (char*)(curso[0]);
162     int*nota = (int*)(curso[1]);
163     cout<<right<<setw(10)<<codigo<<setw(5)<<*nota<<endl;
164 }
165
166 void cargarNotas(void *al,const char*nombArch){
167     ifstream arch(nombArch,ios::in);
168     if(not arch.is_open()){
169         cout<<"El archivo "<<nombArch<<"no se abrio"<<endl;
170         exit(1);
171     }
172     void **alumnos=(void**)al;
173     int nd[50]{}, cap[50]{}, cod, pos;
174
175     while(true){
176         arch>>cod;
177         if(arch.eof())break;
178         arch.get();
179         pos = buscar(cod,alumnos);
180         if(pos!= -1)
181             colocarCurso(arch,alumnos[pos],nd[pos],cap[pos]);
182         else
183             while(arch.get()!='\n');
184     }
185 }
186
187 int buscar(int cod,void **alumnos){
188     for(int i=0; alumnos[i]; i++)
189         if(sonIguales(cod,alumnos[i])) return i;
190     return -1;
191 }
192
193 bool sonIguales(int cod,void*al){
194     void **alumno = (void**)al;
195     int *codigo = (int*)alumno[COD];
196     return cod==*codigo;
197 }
198
```

```
199  void colocarCurso(ifstream &arch,void *al,int &nd,int &cap){
200      void **alumno=(void**)al;
201      void *cur;
202      cur = leeRCurso(arch);
203      if(nd == cap)
204          incrementarEspacios(alumno[NOT],nd,cap);
205      agregarCurso(cur,alumno[NOT],nd);
206      nd++;
207  }
208
209  void incrementarEspacios(void *&al,int &nd, int &cap){
210      void **aux, **alumnos = (void**)al;
211
212      cap += INCREMENTO;
213      if (alumnos == nullptr){
214          alumnos = new void*[cap]{}; // No olvidar las {}
215          nd = 1;
216      }
217      else{
218          aux = new void* [cap]{};
219          for (int i=0; i<nd; i++)
220              aux[i] = alumnos[i];
221          delete alumnos;
222          alumnos = aux;
223      }
224      al = alumnos;
225  }
226
227  void *leeRCurso(ifstream &arch){
228      char*codigo;
229      int*nota = new int;
230      void**registro = new void*[2];
231      codigo = leeCadena(arch,',');
232      arch>>*nota;
233      registro[0]=codigo;
234      registro[1]=nota;
235      return registro;
236  }
237
238  void agregarCurso(void*cur,void *alNota, int nd){
239      void **alumnoNota = (void**)alNota;
240      alumnoNota[nd-1] = cur;
241  }
242
```