```cpp
/*
 * Proyecto: AlumnosCursos_DosArchivosPorIncrementos
 * Archivo:  main.cpp
 * Autor:    J. Miguel Guanira E.//miguel.guanira.
 *
 * Created on 3 de abril de 2024, 08:56 AM
 */

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

int main(int argc, char** argv) {
    int *codigo;
    char **nombre, ***cursos;

    cargarAlumnos(codigo,nombre,"Alumnos.csv");
    cargarCursos(codigo,cursos,"Cursos.csv");
    reporte(codigo,nombre,cursos);

    return 0;
}

/*
 * Proyecto: AlumnosCursos_DosArchivosPorIncrementos
 * Archivo:  FuncionesAuxiliares.h
 * Autor:    J. Miguel Guanira E. //miguel.guanira.
 *
 * Created on 3 de abril de 2024, 08:57 AM
 */

#ifndef FUNCIONESAUXILIARES_H
#define FUNCIONESAUXILIARES_H

void cargarAlumnos(int *&codigo,char **&nombre,const char *nombArch);
void reporte(int *codigo,char **nombre,char ***cursos);
void reporteCursos(char**cursos);
void aumentarEspacios(int *&codigo, char**&nombre,int &nd, int &cap);
char *leeCadena(ifstream &arch);
void colocarCurso(ifstream &arch, char**&cursos,int &nd,int &cap);
void aumentarEspacios(char **&cursos,int &nd,int &cap);
int buscarAlumno(int cod,int *codigo);
void cargarCursos(int *codigo,char ***&cursos,const char *nombArch);
#endif /* FUNCIONESAUXILIARES_H */

/*
 * Proyecto: AlumnosCursos_DosArchivosPorIncrementos
 * Archivo:  FuncionesAuxiliares.cpp
 * Autor:    J. Miguel Gunira E//miguel.guanira.
 *
 * Created on 3 de abril de 2024, 08:57 AM
 */

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;
#include <cstring>
#include "FuncionesAuxiliares.h"
#define INCREMENTO 5

void cargarAlumnos(int *&codigo,char **&nombre,const char *nombArch){

    int numDat=0, cap=0, cod;
    char*nomb;
```

```
67
68          ifstream arch(nombArch,ios::in);
69          if(not arch.is_open()){
70              cout<<"El archivo "<<nombArch<<"no se abrio"<<endl;
71              exit(1);
72          }
73         codigo = nullptr;
74         nombre = nullptr;
75         while(true){
76             arch>>cod;
77             if(arch.eof())break;
78             arch.get();         //coma
79             nomb = leeCadena(arch);
80             if(numDat==cap)
81                 aumentarEspacios(codigo,nombre,numDat,cap);
82             codigo[numDat-1]=cod;
83             nombre[numDat-1]=nomb;
84             numDat++;
85         }
86     }
87
88     void aumentarEspacios(int *&codigo, char**&nombre,int &nd, int &cap){
89         int *auxCod;
90         char **auxNomb;
91
92         cap += INCREMENTO;
93         if (codigo == nullptr){
94             codigo = new int[cap]{}; // No olvidar las {}
95             nombre = new char*[cap]{}; // No olvidar las {}
96             nd = 1;
97         }
98         else{
99             auxCod = new int [cap]{};
100            auxNomb = new char*[cap]{};
101            for (int i=0; i<nd; i++){
102                auxCod[i] = codigo[i];
103                auxNomb[i] = nombre[i];
104            }
105            delete codigo;
106            delete nombre;
107            codigo = auxCod;
108            nombre = auxNomb;
109        }
110    }
111
112    char *leeCadena(ifstream &arch){
113        char cadena[60], *cad;
114        arch.getline(cadena,60);
115        cad = new char [strlen(cadena)+1];
116        strcpy(cad,cadena);
117        return cad;
118    }
119
120    void cargarCursos(int *codigo,char ***&cursos,const char *nombArch){
121        ifstream arch(nombArch,ios::in);
122        if(not arch.is_open()){
123            cout<<"El archivo "<<nombArch<<"no se abrio"<<endl;
124            exit(1);
125        }
126        int numDat=0, nd[50]{},cap[50]{},cod, pos;
127
128        //Inicializar cursos;
129        while(codigo[numDat])numDat++;
130        cursos = new char**[numDat+1]{};
131        while(true){
132            arch>>cod;
```

```
133             if(arch.eof())break;
134             arch.get();
135             pos = buscarAlumno(cod,codigo);
136             if(pos!=-1){
137                 colocarCurso(arch,cursos[pos],nd[pos],cap[pos]);
138             }
139             else
140                 while(arch.get()!='\n');
141         }
142     }
143
144     void colocarCurso(ifstream &arch, char**&cursos,int &nd,int &cap){
145         char *cur;
146         cur = leeCadena(arch);
147         if(nd == cap)
148             aumentarEspacios(cursos,nd,cap);
149         cursos[nd-1]=cur;
150         nd++;
151     }
152
153     void aumentarEspacios(char **&cursos,int &nd,int &cap){
154         char **auxCur;
155
156         cap += INCREMENTO;
157         if (cursos == nullptr){
158             cursos = new char*[cap]{}; // No olvidar las {}
159             nd = 1;
160         }
161         else{
162             auxCur = new char*[cap]{};
163             for (int i=0; i<nd; i++){
164                 auxCur[i] = cursos[i];
165             }
166             delete cursos;
167             cursos = auxCur;
168         }
169     }
170
171     int buscarAlumno(int cod,int *codigo){
172         for(int i=0; codigo[i]; i++){
173             if(cod == codigo[i])return i;
174         }
175         return -1;
176     }
177
178     void reporte(int *codigo,char **nombre,char ***cursos){
179        for(int i=0; codigo[i]; i++){
180             cout<<left<<setw(10)<<codigo[i]<<nombre[i]<<endl;
181             if(cursos[i])
182                 reporteCursos(cursos[i]);
183             else
184                 cout<<"NO TIENE CURSOS"<<endl;
185         }
186     }
187
188     void reporteCursos(char**cursos){
189         for(int i=0; cursos[i]; i++){
190             cout<<right<<setw(15)<<cursos[i]<<endl;
191         }
192     }
```