

```
1  /*
2  * Proyecto: MemoriaEXacta-DosArchivos
3  * Archivo:  main.cpp
4  * Autor:    J. Miguel Guanira E.//miguel.guanira.
5  *
6  * Created on 3 de septiembre de 2024, 08:56 AM
7  */
8
9  #include <iostream>
10 #include <iomanip>
11 using namespace std;
12 #include "FuncionesAuxiliares.h"
13
14 int main(int argc, char** argv) {
15     int *codigos;
16     char **nombres, ***cursos;
17     cargarAlumnos(codigos,nombres,"Alumnos.csv");
18     cargarCURsos(codigos,cursos,"Cursos.csv");
19     imprimirAlumnos(codigos,nombres,cursos,"ReporteDeAlumnos.txt");
20     return 0;
21 }
22
23 /*
24 * Proyecto: MemoriaPorIncrementos-DosArchivos
25 * Archivo:  FuncionesAuxiliares.h
26 * Autor:    J. Miguel Guanira E. //miguel.guanira.
27 *
28 * Created on 3 de septiembre de 2024, 08:57 AM
29 */
30
31 #ifndef FUNCIONESAUXILIARES_H
32 #define FUNCIONESAUXILIARES_H
33
34 void cargarAlumnos(int *&codigos,char**&nombres,const char *nombArch);
35 char*leeCadena(istream &arch, char='\n');
36 void imprimirAlumnos(int *codigos,char**nombres,
37                     char ***cursos,const char *nombArch);
38 void reporteDeCursos(ofstream &arch,char **cursos);
39 void cargarCURsos(int *codigos, char ***&cursos,const char *nombArch);
40 int buscarAlumnos(int codAl,int *codigos, int numDat);
41 void colocarCurso(istream &arch, char**cursos,int &nd);
42 void espaciosExactos(char**&cursos,int nd);
43
44 #endif /* FUNCIONESAUXILIARES_H */
45
46 /*
47 * Proyecto: MemoriaExacta-DosArchivos
48 * Archivo:  FuncionesAuxiliares.cpp
49 * Autor:    J. Miguel Gunira E//miguel.guanira.
50 *
51 * Created on 3 de septiembre de 2024, 08:57 AM
52 */
53
54 #include <iostream>
55 #include <fstream>
56 #include <iomanip>
57 using namespace std;
58 #include "FuncionesAuxiliares.h"
59 #include <cstring>
60
61 void cargarAlumnos(int *&codigos,char**&nombres,const char *nombArch){
62     ifstream arch(nombArch,ios::in);
63     if(not arch.is_open()){
64         cout<<"ERROR, No se pudo abrir el archivo "<<nombArch<<endl;
65         exit(1);
66     }
```

```
67     int buffCod[50] {}, numDat=0, cod;
68     char *buffNomb[50] {};
69     while(true){
70         arch>>cod;
71         if(arch.eof())break;
72         arch.get(); // sacamos la coma
73         buffCod[numDat] = cod;
74         buffNomb[numDat] = leeCadena(arch); // Aquí estaba el error
75         numDat++;
76     }
77     codigos = new int[numDat+1] {};
78     nombres = new char*[numDat+1] {};
79     for (int i = 0; i < numDat; i++) {
80         codigos[i] = buffCod[i];
81         nombres[i] = buffNomb[i];
82     }
83 }
84
85 char*leeCadena(istream &arch, char delimitador){
86     char buffer[60], *cad;
87     arch.getline(buffer,60,delimitador);
88     cad = new char[strlen(buffer)+1];
89     strcpy(cad, buffer);
90     return cad;
91 }
92
93 void cargarCursos(int *codigos,char ***&cursos,const char *nombArch){
94     ifstream arch(nombArch,ios::in);
95     if(not arch.is_open()){
96         cout<<"ERROR, No se pudo abrir el archivo "<<nombArch<<endl;
97         exit(1);
98     }
99     int numDat=0, codAl, pos ;
100     while(codigos[numDat])numDat++;
101     int nd[numDat] {};
102
103     cursos = new char**[numDat+1] {};
104     for (int i = 0; i<numDat; i++)
105         cursos[i] = new char*[50] {};
106
107     while(true){
108         arch>>codAl;
109         if(arch.eof())break;
110         arch.get(); // quitamos la coma
111         pos = buscarAlumnos(codAl,codigos,numDat);
112         if(pos!=-1)
113             colocarCurso(arch,cursos[pos],nd[pos]);
114     }
115
116     for (int i = 0; i < numDat; i++)
117         espaciosExactos(cursos[i],nd[i]);
118 }
119
120 int buscarAlumnos(int codAl,int *codigos, int numDat){
121     for (int i = 0; i < numDat; i++) {
122         if(codAl == codigos[i]) return i;
123     }
124     return -1;
125 }
126
127 void colocarCurso(istream &arch, char**cursos,int &nd){
128     char *codCur = leeCadena(arch);
129     cursos[nd] = codCur;
130     nd++;
131 }
132
```

```
133 void espaciosExactos(char**&cursos,int nd){
134     char **espaciosExactos;
135     if(nd !=0 ){
136         espaciosExactos = new char*[nd+1]{};
137         for (int i = 0; i < nd; i++)
138             espaciosExactos[i] = cursos[i];
139         delete cursos;
140         cursos = espaciosExactos;
141     }
142     else{
143         delete cursos;
144         cursos=nullptr;
145     }
146
147 }
148
149 void imprimirAlumnos(int *codigos,char**nombres,
150                     char ***cursos,const char *nombArch){
151     ofstream arch(nombArch,ios::out);
152     if(not arch.is_open()){
153         cout<<"ERROR, No se pudo abrir el archivo "<<nombArch<<endl;
154         exit(1);
155     }
156     for (int i = 0; codigos[i]; i++) {
157         arch<<left<<setw(10)<<codigos[i]<<setw(45)<<nombres[i]<<endl;
158         reporteDeCursos(arch,cursos[i]);
159     }
160 }
161
162 void reporteDeCursos(ofstream &arch,char **cursos){
163     for (int i = 0; cursos[i]; i++) {
164         arch<<right<<setw(15)<<cursos[i]<<endl;
165     }
166 }
167 }
```