

```
1  /*
2  * Proyecto: MemoriaExacta-UnArchivo
3  * Archivo:  main.cpp
4  * Autor:    J. Miguel Guanira E.//miguel.guanira.
5  *
6  * Created on 28 de agosto de 2024, 09:06 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,cursos,"Alumnos-Cursos.csv");
18     imprimirAlumnos(codigos,nombres,cursos,"ReporteDeAlumnos.txt");
19     return 0;
20 }
21
22 /*
23 * Proyecto: MemoriaExacta-UnArchivo
24 * Archivo:  FuncionesAuxiliares.h
25 * Autor:    J. Miguel Guanira E. //miguel.guanira.
26 *
27 * Created on 28 de agosto de 2024, 09:22 AM
28 */
29
30 #ifndef FUNCIONES_AUXILIARES_H
31 #define FUNCIONES_AUXILIARES_H
32
33 void cargarAlumnos(int *&codigos,char**&nombres,
34                   char ***&cursos,const char *nombArch);
35 void imprimirAlumnos(int *codigos,char**nombres,
36                     char ***cursos,const char *nombArch);
37 char ** leeCursos(ifstream &arch);
38 char*leeCadena(ifstream &arch);
39 char * leeCadenaComa(ifstream &arch);
40 void reporteDeCursos(ofstream &arch,char **cursos);
41
42 #endif /* FUNCIONES_AUXILIARES_H */
43
44 /*
45 * Proyecto: MemoriaExacta-UnArchivo
46 * Archivo:  FuncionesAuxiliares.cpp
47 * Autor:    J. Miguel Gunira E//miguel.guanira.
48 *
49 * Created on 28 de agosto de 2024, 09:23 AM
50 */
51
52 #include <iostream>
53 #include <fstream>
54 #include <iomanip>
55 using namespace std;
56 #include "FuncionesAuxiliares.h"
57 #include <cstring>
58
59 void cargarAlumnos(int *&codigos,char**&nombres,
60                   char ***&cursos,const char *nombArch){
61     ifstream arch(nombArch,ios::in);
62     if(not arch.is_open()){
63         cout<<"ERROR, No se pudo abrir el archivo "<<nombArch<<endl;
64         exit(1);
65     }
66     int buffCod[50] {}, numDat=0, cod;
```

```
67     char *buffNomb[50] {}, **buffCur[50] {};  
68     while(true) {  
69         arch>>cod;  
70         if(arch.eof()) break;  
71         arch.get(); // sacamos la coma  
72         buffCod[numDat] = cod;  
73         buffNomb[numDat] = leeCadena(arch);  
74         buffCur[numDat] = leeCursos(arch);  
75         numDat++;  
76     }  
77     codigos = new int[numDat+1] {};  
78     nombres = new char*[numDat+1] {};  
79     cursos = new char**[numDat+1] {};  
80     for (int i = 0; i < numDat; i++) {  
81         codigos[i] = buffCod[i];  
82         nombres[i] = buffNomb[i];  
83         cursos[i] = buffCur[i];  
84     }  
85 }  
86  
87 char* leeCadena(ifstream &arch) {  
88     char buffer[60], *cad;  
89     arch.getline(buffer, 60, ',');  
90     cad = new char[strlen(buffer)+1];  
91     strcpy(cad, buffer);  
92     return cad;  
93 }  
94  
95 char ** leeCursos(ifstream &arch) {  
96     char *buffCad[20] {}, *cur, **cursos;  
97     int nc=0;  
98     while(true) {  
99         cur = leeCadenaComa(arch);  
100        buffCad[nc] = cur;  
101        nc++;  
102        if(arch.get() == '\\n') break;  
103    }  
104    cursos = new char*[nc+1] {};  
105    for (int i = 0; i < nc; i++) {  
106        cursos[i] = buffCad[i];  
107    }  
108    return cursos;  
109 }  
110  
111 char * leeCadenaComa(ifstream &arch) {  
112     char *cad;  
113     cad = new char[7];  
114     arch.get(cad, 7);  
115     return cad;  
116 }  
117  
118 void imprimirAlumnos(int *codigos, char**nombres,  
119                     char ***cursos, const char *nombArch) {  
120     ofstream arch(nombArch, ios::out);  
121     if(not arch.is_open()) {  
122         cout<<"ERROR, No se pudo abrir el archivo "<<nombArch<<endl;  
123         exit(1);  
124     }  
125     for (int i = 0; codigos[i]; i++) {  
126         arch<<left<<setw(10)<<codigos[i]<<setw(45)<<nombres[i]<<endl;  
127         reporteDeCursos(arch, cursos[i]);  
128     }  
129 }  
130  
131 void reporteDeCursos(ofstream &arch, char **cursos) {  
132     for (int i = 0; cursos[i]; i++) {
```

```
133         arch<<right<<setw(15)<<ursos[i]<<endl;  
134     }  
135  
136 }
```