

```
1  /*
2  * Proyecto: QSort
3  * Archivo:  main.cpp
4  * Autor:    J. miguel.guanira.
5  *
6  * Created on 1 de abril de 2024, 11:45
7  */
8
9  #include <iostream>
10 #include <iomanip>
11 using namespace std;
12 #include "FuncionQSort.h"
13
14 int main(int argc, char** argv) {
15     int *datos;
16     cargarDatos(datos, "datos.txt");
17     ordenarDatos(datos);
18     reporte(datos);
19     return 0;
20 }
21
22 /*
23 * Proyecto: QSort
24 * Archivo:  FuncionQsort.h
25 * Autor:    J. miguel.guanira.
26 *
27 * Created on 1 de abril de 2024, 11:46
28 */
29
30 #ifndef FUNCIONQSORT_H
31 #define FUNCIONQSORT_H
32
33 void cargarDatos(int *&codigo, const char*nombArch);
34 void aumentarEspacios(int *&codigo,int &numDat,int &cap);
35 void reporte(int *codigo);
36 void ordenarDatos(int *codigo);
37 void qSort(int *codigo,int izq,int der);
38 void cambiar(int &codigoI,int &codigoK);
39
40 #endif /* FUNCIONQSORT_H */
41
42 /*
43 * Proyecto: MemoriaPorIncrementos-DosArchivos
44 * Archivo:  FuncionesAuxiliares.cpp
45 * Autor:    J. Miguel Gunira E//miguel.guanira.
46 *
47 * Created on 29 de agosto de 2023, 09:47 AM
48 */
49
50 #include <iostream>
51 #include <fstream>
52 #include <iomanip>
53 using namespace std;
54 #include <cstring>
55 #include "FuncionQSort.h"
56 #define INCREMENTO 5
57
58 void cargarDatos(int *&datos, const char*nombArch){
59     int numDat=0, cap=0, cod;
60
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     datos = nullptr;
```

```
67
68     while(true){
69         arch>>cod;
70         if(arch.eof())break;
71         if(numDat == cap) aumentarEspacios(datos,numDat,cap);
72         datos[numDat-1]=cod;
73         numDat++;
74     }
75 }
76
77 void aumentarEspacios(int *&datos,int &numDat,int &cap){
78     int *auxCod;
79
80     cap += INCREMENTO;
81     if(datos == nullptr){
82         datos = new int[cap]{}; //No OLVIDAR {}
83         numDat = 1;
84     }
85     else{
86         auxCod = new int[cap]{}; //No OLVIDAR {}
87         for(int i=0; i<numDat; i++)
88             auxCod[i] = datos[i];
89
90         delete datos;
91         datos = auxCod;
92     }
93 }
94
95
96 void reporte(int *datos){
97     for(int i=0; datos[i]; i++)
98         cout<<setw(10)<<datos[i]<<endl;
99 }
100
101 void ordenarDatos(int *datos){
102     int numDat=0;
103     while(datos[numDat]) numDat++;
104     qSort(datos,0,numDat-1);
105 }
106
107 void qSort(int *datos,int izq,int der){
108     int limite, pMedio;
109     if(izq>= der) return;
110     pMedio = (izq+der)/2;
111     cambiar(datos[izq],datos[pMedio]);
112     limite = izq;
113     for (int i = izq+1; i <= der; i++) {
114         if(datos[i]<datos[izq]){
115             cambiar(datos[++limite],datos[i]);
116         }
117     }
118     cambiar(datos[limite],datos[izq]);
119     qSort(datos,izq,limite-1);
120     qSort(datos,limite+1,der);
121 }
122
123 void cambiar(int &datosI,int &datosK){
124     int aux;
125     aux = datosI;
126     datosI = datosK;
127     datosK = aux;
128 }
```