

```
1  /*
2  * Proyecto: MemoriaPorIncrementos
3  * Archivo:  main.cpp
4  * Autor:    J. Miguel Guanira E.//miguel.guanira.
5  *
6  * Created on 3 de abril de 2024, 08:16 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 *datos;
16
17     leerDatosPorIncrementos(datos,"datos.txt");
18     ordenar(datos);
19     imprimirDatos(datos);
20
21     return 0;
22 }
23
24 /*
25 * Proyecto: MemoriaPorIncrementos
26 * Archivo:  FuncionesAuxiliares.h
27 * Autor:    J. Miguel Guanira E. //miguel.guanira.
28 *
29 * Created on 3 de abril de 2024, 08:25 AM
30 */
31
32 #ifndef FUNCIONES_AUXILIARES_H
33 #define FUNCIONES_AUXILIARES_H
34
35 void leerDatosPorIncrementos(int *&datos,const char *nombArch);
36 void aumentarEspacios(int *&datos,int &nd, int &cap);
37 void imprimirDatos(int *datos);
38 void ordenar(int *datos);
39 void qSort(int *datos, int izq, int der);
40 void cambiar(int &datoI,int &datoK);
41
42 #endif /* FUNCIONES_AUXILIARES_H */
43
44 /*
45 * Proyecto: MemoriaPorIncrementos
46 * Archivo:  FuncionesAuxiliares.cpp
47 * Autor:    J. Miguel Gunira E//miguel.guanira.
48 *
49 * Created on 3 de abril de 2024, 08:25 AM
50 */
51
52 #include <iostream>
53 #include <fstream>
54 #include <iomanip>
55 using namespace std;
56 #include "FuncionesAuxiliares.h"
57
58 #define INCREMENTO 5
59
60 void leerDatosPorIncrementos(int *&datos,const char *nombArch){
61     ifstream arch(nombArch,ios::in);
62     if(not arch.is_open()){
63         cout<<"El archivo "<<nombArch<<"no se abrio"<<endl;
64         exit(1);
65     }
66     int nd=0,cap=0, dato;
```

```
67     datos=NULLPTR;
68     while(true){
69         arch>>dato;
70         if(arch.eof())break;
71         if(nd==cap)
72             aumentarEspacios(datos,nd,cap);
73         datos[nd-1] = dato;
74         nd++;
75     }
76 }
77
78 void aumentarEspacios(int *&datos,int &nd, int &cap){
79     int *aux;
80
81     cap += INCREMENTO;
82     if (datos == nullptr){
83         datos = new int[cap]{}; // No olvidar las {}
84         nd = 1;
85     }
86     else{
87         aux = new int [cap]{};
88         for (int i=0; i<nd; i++)
89             aux[i] = datos[i];
90         delete datos;
91         datos = aux;
92     }
93 }
94
95 void ordenar(int *datos){
96     int numDat=0;
97     while(datos[numDat])numDat++;
98     qSort(datos,0, numDat-1);
99 }
100
101 void qSort(int *datos, int izq, int der){
102     int limite,pos;
103     if(izq >= der)return;
104     pos = (izq+der)/2;
105     cambiar(datos[izq],datos[pos]);
106     limite = izq;
107     for(int i=izq+1;i<=der; i++)
108         if(datos[i]<datos[izq]){
109             limite++;
110             cambiar(datos[limite],datos[i]);
111         }
112     cambiar(datos[limite],datos[izq]);
113     qSort(datos,izq,limite-1);
114     qSort(datos,limite+1,der);
115 }
116
117 void cambiar(int &datoI,int &datoK){
118     int aux;
119     aux=datoI;
120     datoI = datoK;
121     datoK = aux;
122 }
123
124 void imprimirDatos(int *datos){
125     for(int i=0; datos[i]; i++)
126         cout<<setw(8)<<datos[i]<<endl;
127 }
128
```