

```

1  /*
2  * File:   main.cpp
3  * Author: ANA RONCAL
4  * Created on 4 de octubre de 2023, 09:42 AM
5  */
6
7  #include <iostream>
8  #include <cstdlib>
9  #include "Cliente.h"
10 using namespace std;
11 #include "funciones.h"
12
13 #define MAXDATOS 100
14
15 /*
16 * EJEMPLO DE USO DE TABLA HASH
17 */
18 int main(int argc, char** argv) {
19
20     struct Cliente cliente[MAXDATOS]{};
21     int indice, clave;
22     leerDatos("Clientes.txt", cliente);
23
24     modificarDatos("Modificaciones.txt", cliente);
25
26     clave = 19970234;
27     indice = funcionHash(clave);
28     cliente[indice].celular = 999888777;
29     mostrarCliente(cliente[indice]);
30
31     mostrarTodos(cliente);
32
33     return 0;
34 }
35
36 /*
37 * File:   Cliente.h
38 * Author: ANA RONCAL
39 * Created on 4 de octubre de 2023, 10:15 AM
40 */
41
42 #ifndef CLIENTE_H
43 #define CLIENTE_H
44
45 struct Cliente{
46     int clave;
47     char nombre[50];
48     char sexo;
49     int celular;
50
51 };
52
53 #endif /* CLIENTE_H */
54
55 *
56 * File:   funciones.h
57 * Author: ANA RONCAL
58 * Created on 4 de octubre de 2023, 09:42 AM
59 */
60
61 #ifndef FUNCIONES_H
62 #define FUNCIONES_H
63
64 void leerDatos(const char * nombreArch, struct Cliente * cliente);
65 int funcionHash(int clave);
66
67 void modificarDatos(const char * nombreArch, struct Cliente * cliente);
68 void mostrarCliente(struct Cliente cliente);
69 void mostrarTodos(struct Cliente * cliente);

```

```

70 void mostrarCabecera();
71
72 #endif /* FUNCIONES_H */
73
74 /*
75  * File:    funciones.cpp
76  * Author:  ANA RONCAL
77  * Created on 4 de octubre de 2023, 09:43 AM
78  */
79
80 #include <iostream>
81 #include <iomanip>
82 #include <fstream>
83 #include <cstring>
84
85 #include "Cliente.h"
86 using namespace std;
87 #include "funciones.h"
88 #define MAXDATOS 100
89
90 void leerDatos(const char * nombreArch, struct Cliente * cliente){
91
92     ifstream arch(nombreArch, ios::in);
93     if(not arch.is_open()){
94         cout<<"Error al abrir el archivo de Clientes"<<endl;
95     }
96     struct Cliente clienteAux;
97
98     int clave;
99     char sexo;
100     char nombre[50];
101     int indice;
102     while(not arch.eof()){
103         arch>>clave;
104         if(arch.eof()) break;
105         arch>>nombre;
106         arch>>sexo;
107         clienteAux.clave = clave;
108         strcpy(clienteAux.nombre, nombre);
109         clienteAux.sexo = sexo;
110         indice = funcionHash(clave);
111         cliente[indice] = clienteAux;
112     }
113 }
114
115 }
116
117 int funcionHash(int clave){
118
119     int dig, indice, suma = 0;
120
121     while(clave){
122         dig = clave % 10;
123         suma = suma + (dig + '0');
124         clave = clave / 10;
125     }
126
127     indice = suma % MAXDATOS;
128     return indice;
129 }
130
131 void modificarDatos(const char * nombreArch, struct Cliente * cliente){
132
133     ifstream arch(nombreArch, ios::in);
134     if(not arch.is_open()){
135         cout<<"Error al abrir el archivo de Clientes"<<endl;
136     }
137
138     mostrarCabecera();

```

```

139     struct Cliente clienteAux;
140     int clave;
141     int indice;
142     int celular;
143     while(not arch.eof()){
144         arch>>clave;
145         if(arch.eof()) break;
146         arch>>celular;
147         indice = funcionHash(clave);
148         cliente[indice].celular = celular;
149         //mostrarCliente(cliente[indice]);
150     }
151 }
152
153 void mostrarCliente(struct Cliente cliente){
154
155     cout<<setw(10)<<cliente.clave
156         <<setw(20)<<cliente.nombre
157         <<setw(10)<<cliente.sexo
158         <<setw(10)<<cliente.celular<<endl;
159
160 }
161
162 void mostrarTodos(struct Cliente * cliente){
163     for(int i = 0; i< MAXDATOS; i++){
164         if(cliente[i].clave != 0){
165             mostrarCliente(cliente[i]);
166         }
167     }
168 }
169
170 void mostrarCabecera(){
171     cout<<left<<setw(10)<<"CLAVE"
172         <<setw(18)<<"CLIENTE"
173         <<setw(10)<<"SEXO"
174         <<setw(10)<<"CELULAR"<<endl;
175 }

```