

```
1  /*
2  * Proyecto: Pregunta01_Examen02_2022-2
3  * Archivo:  main.cpp
4  * Autor:    J. Miguel Guanira E.
5  *
6  * Created on 23 de noviembre de 2022, 07:24 PM
7  */
8
9  #include <iostream>
10 #include <iomanip>
11 using namespace std;
12 #include "ServicioDeSalud.h"
13 int main(int argc, char** argv) {
14     class ServicioDeSalud servicioDeSalud;
15     servicioDeSalud.leerMedicinas("Medicinas-Preg01.csv");
16     servicioDeSalud.imprimirMedicinas("PruebaMedicinas.txt");
17     servicioDeSalud.leerConsultas("Consultas-Preg01.csv");
18     servicioDeSalud.imprimePacientes("PruebaConsultas.txt");
19     servicioDeSalud.totalizar();
20     servicioDeSalud.imprimePacientes("PruebaConsultasTotal.txt");
21     return 0;
22 }
23
24 /*
25 * Proyecto: Pregunta01_Examen02_2022-2
26 * Archivo:  Medicina.h
27 * Autor:    J. Miguel Guanira E.
28 *
29 * Created on 23 de noviembre de 2022, 07:25 PM
30 */
31
32
33 #ifndef MEDICINA_H
34 #define MEDICINA_H
35
36 class Medicina {
37 private:
38     int codigo;
39     char *descripcion;
40     double precio;
41 public:
42     Medicina();
43     Medicina(const Medicina& orig);
44     virtual ~Medicina();
45     void SetPrecio(double precio);
46     double GetPrecio() const;
47     void SetDescripcion(const char* descripcion);
48     void GetDescripcion(char*) const;
49     void SetCodigo(int codigo);
50     int GetCodigo() const;
51     void operator = (const Medicina& orig);
52 };
53
54 void operator >>(ifstream &, class Medicina&);
55 void operator <<(ofstream &,const class Medicina&);
56 #endif /* MEDICINA_H */
57
58 /*
59 * Proyecto: Pregunta01_Examen02_2022-2
60 * Archivo:  Paciente.h
61 * Autor:    J. Miguel Guanira E.
62 *
63 * Created on 23 de noviembre de 2022, 07:38 PM
64 */
65
66
```

```
67 #ifndef PACIENTE_H
68 #define PACIENTE_H
69
70 class Paciente {
71 private:
72     int dni;
73     char *nombre;
74     int fechaUltimaConsulta;
75     char *codigoDelMedicoDeUltimaConsulta;
76 public:
77     Paciente();
78     Paciente(const Paciente& orig);
79     virtual ~Paciente();
80     void SetFechaUltimaConsulta(int fechaUltimaConsulta);
81     int GetFechaUltimaConsulta() const;
82     void SetNombre(const char* nombre);
83     void GetNombre(char*) const;
84     void SetDni(int dni);
85     int GetDni() const;
86     void SetCodigoDelMedicoDeUltimaConsulta(const char*);
87     void GetCodigoDelMedicoDeUltimaConsulta(char*) const;
88     void operator = (const Paciente& orig);
89     void leerDatos(istream &);
90     void imprimirDatos(ofstream &arch);
91 };
92
93 #endif /* PACIENTE_H */
94
95 /*
96  * Proyecto: Pregunta01_Examen02_2022-2
97  * Archivo:  ServicioDeSalud.h
98  * Autor:    J. Miguel Guanira E.
99  *
100  * Created on 23 de noviembre de 2022, 08:22 PM
101  */
102
103
104 #ifndef SERVICIODESALUD_H
105 #define SERVICIODESALUD_H
106 #include <list>
107 #include <map>
108 using namespace std;
109 #include "Medicina.h"
110 #include "Paciente_Medicina.h"
111
112 class ServicioDeSalud {
113 private:
114     map<int, Medicina> medicina;
115     list<Paciente_Medicina> paciente_medicina;
116 public:
117     leerMedicinas(const char*);
118     imprimirMedicinas(const char*);
119     leerConsultas(const char*);
120     imprimePacientes(const char*);
121     list<Paciente_Medicina>::iterator buscarPaciente(int);
122     void totalizar();
123 };
124
125 #endif /* SERVICIODESALUD_H */
126
127 /*
128  * Proyecto: Pregunta01_Examen02_2022-2
129  * Archivo:  ServicioDeSalud.cpp
130  * Autor:    J. Miguel Guanira E.
131  *
132  * Created on 23 de noviembre de 2022, 08:22 PM
```

```

133     */
134
135     #include <iostream>
136     #include <fstream>
137     #include <iomanip>
138     #include <map>
139     #include <list>
140     #include <iterator>
141     using namespace std;
142     #include "Medicina.h"
143     #include "ServicioDeSalud.h"
144
145     int ServicioDeSalud::leerMedicinas(const char*nomArch) {
146         ifstream arch(nomArch, ios::in);
147         if(not arch.is_open()){
148             cout<<"ERROR: no se pudo abrir el archivo "<<nomArch<<endl;
149             exit(1);
150         }
151         class Medicina med;
152         while(true){
153             arch>>med;
154             if(arch.eof())break;
155             medicina[med.GetCodigo()] = med;
156         }
157     }
158
159     int ServicioDeSalud::imprimirMedicinas(const char*nomArch) {
160         ofstream arch(nomArch, ios::out);
161         if(not arch.is_open()){
162             cout<<"ERROR: no se pudo abrir el archivo "<<nomArch<<endl;
163             exit(1);
164         }
165         for(map<int,Medicina>::iterator it=medicina.begin();
166             it!=medicina.end(); it++)
167             arch<<it->second;
168     }
169
170     int ServicioDeSalud::leerConsultas(const char*nomArch) {
171         ifstream arch(nomArch, ios::in);
172         if(not arch.is_open()){
173             cout<<"ERROR: no se pudo abrir el archivo "<<nomArch<<endl;
174             exit(1);
175         }
176         class Paciente pac;
177         list<class Paciente_Medicina>::iterator itPacMed;
178         while(true){
179             class Paciente_Medicina pacMed;
180             pac.leerDatos(arch);
181             if(arch.eof())break;
182             itPacMed = buscarPaciente(pac.GetDni());
183             if(itPacMed == paciente_medicina.end()){
184                 pacMed = pac;
185                 pacMed.leeMedicinas(arch);
186                 paciente_medicina.push_back(pacMed);
187             }
188             else{
189                 if (*itPacMed>pac)
190                     pacMed = pac; // para simplificar copiamos todos los datos
191                                     // incluyendo la nueva fecha y nuevo codigo del médico
192                 itPacMed->leeMedicinas(arch);
193             }
194         }
195     }
196
197     list<Paciente_Medicina>::iterator ServicioDeSalud::buscarPaciente(int dni) {
198         for(list<Paciente_Medicina>::iterator it=paciente_medicina.begin();

```

```

199         it!=paciente_medicina.end(); it++)
200         if(it->GetDni()==dni) return it;
201     return paciente_medicina.end();
202 }
203
204 int ServicioDeSalud::imprimePacientes(const char*nomArch) {
205     ofstream arch(nomArch, ios::out);
206     if(not arch.is_open()){
207         cout<<"ERROR: no se pudo abrir el archivo "<<nomArch<<endl;
208         exit(1);
209     }
210     for(list<Paciente_Medicina>::iterator it=paciente_medicina.begin();
211         it!=paciente_medicina.end(); it++){
212         it->imprimePacienteMedicina(arch);
213     }
214 }
215
216 void ServicioDeSalud::totalizar() {
217     for(list<Paciente_Medicina>::iterator it=paciente_medicina.begin();
218         it!=paciente_medicina.end(); it++){
219         it->totalizar(medicina);
220     }
221 }
222
223 /*
224  * Proyecto: Pregunta01_Examen02_2022-2
225  * Archivo: Paciente_Medicina.h
226  * Autor: J. Miguel Guanira E.
227  *
228  * Created on 23 de noviembre de 2022, 08:13 PM
229  */
230
231
232 #ifndef PACIENTE_MEDICINA_H
233 #define PACIENTE_MEDICINA_H
234 #include <list>
235 #include <map>
236 using namespace std;
237 #include "Paciente.h"
238 #include "Medicina.h"
239 #include "Medicina_Cantidad.h"
240
241 class Paciente_Medicina {
242 private:
243     class Paciente paciente;
244     list<class Medicina_Cantidad>medicina_cantidad;
245     double totalDeGastos;
246 public:
247     Paciente_Medicina();
248     void SetTotalDeGastos(double totalDeGastos);
249     double GetTotalDeGastos() const;
250     bool operator>(const class Paciente &pac);
251     int GetDni();
252     void operator=(const class Paciente &pac);
253     void leeMedicinas(ifstream &arch);
254     list<class Medicina_Cantidad>::iterator buscarMedicina(int codMed);
255     void imprimePacienteMedicina(ofstream &arch);
256     void totalizar(map<int,Medicina>med);
257 };
258
259 #endif /* PACIENTE_MEDICINA_H */
260
261 /*
262  * Proyecto: Pregunta01_Examen02_2022-2
263  * Archivo: Paciente_Medicina.cpp
264  * Autor: J. Miguel Guanira E.

```

```
265      *
266      * Created on 23 de noviembre de 2022, 08:13 PM
267      */
268
269      #include <iostream>
270      #include <fstream>
271      #include <iomanip>
272      #include <list>
273      #include <iterator>
274      using namespace std;
275
276      #include "Paciente_Medicina.h"
277
278      Paciente_Medicina::Paciente_Medicina() {
279          totalDeGastos = 0;
280      }
281
282      void Paciente_Medicina::SetTotalDeGastos(double totalDeGastos) {
283          this->totalDeGastos = totalDeGastos;
284      }
285
286      double Paciente_Medicina::GetTotalDeGastos() const {
287          return totalDeGastos;
288      }
289
290      bool Paciente_Medicina::operator>(const class Paciente& pac) {
291          return paciente.GetFechaUltimaConsulta()>pac.GetFechaUltimaConsulta();
292      }
293
294      int Paciente_Medicina::GetDni() {
295          return paciente.GetDni();
296      }
297
298      void Paciente_Medicina::operator=(const class Paciente &pac) {
299          paciente = pac;
300      }
301
302      void Paciente_Medicina::leeMedicinas(ifstream& arch) {
303          class Medicina_Cantidad med_Cant;
304          char fin;
305          int cant;
306          list<class Medicina_Cantidad>::iterator itMedCant;
307          while(true){
308              fin = arch>>med_Cant;
309              itMedCant = buscarMedicina(med_Cant.GetCodigo());
310              if(itMedCant == medicina_cantidad.end())
311                  medicina_cantidad.push_back(med_Cant);
312              else{
313                  cant = med_Cant.GetCantidad();
314                  cant += itMedCant->GetCantidad();
315                  itMedCant->SetCantidad(cant);
316              }
317              if(fin=='\n')break;
318          }
319      }
320
321      list<Medicina_Cantidad>::iterator Paciente_Medicina::buscarMedicina(int codMed){
322          for(list<Medicina_Cantidad>::iterator it=medicina_cantidad.begin();
323              it!=medicina_cantidad.end(); it++)
324              if(it->GetCodigo()==codMed) return it;
325          return medicina_cantidad.end();
326      }
327
328      void Paciente_Medicina::imprimePacienteMedicina(ofstream& arch) {
329          paciente.ImprimirDatos(arch);
330          for(list<Medicina_Cantidad>::iterator it=medicina_cantidad.begin();
```

```
331         it!=medicina_cantidad.end(); it++){
332             it->imprimeMedicina(arch);
333         }
334         arch.precision(2);
335         arch<<fixed;
336         arch<<totalDeGastos<<endl;
337     }
338
339     void Paciente_Medicina::totalizar(map<int, Medicina> med) {
340         int cod, cant;
341         for(list<Medicina_Cantidad>::iterator it=medicina_cantidad.begin();
342             it!=medicina_cantidad.end(); it++){
343             cod = it->GetCodigo();
344             cant = it->GetCantidad();
345
346             totalDeGastos += cant * med[cod].GetPrecio();
347         }
348     }
349
350     /*
351     * Proyecto: Pregunta01_Examen02_2022-2
352     * Archivo: Medicina_Cantidad.h
353     * Autor: J. Miguel Guanira E.
354     *
355     * Created on 23 de noviembre de 2022, 08:07 PM
356     */
357
358
359     #ifndef MEDICINA_CANTIDAD_H
360     #define MEDICINA_CANTIDAD_H
361
362     class Medicina_Cantidad {
363     private:
364         int codigo;
365         int cantidad;
366     public:
367         void SetCantidad(int cant);
368         int GetCantidad() const;
369         void SetCodigo(int codig);
370         int GetCodigo() const;
371         void imprimeMedicina(ofstream &arch);
372     };
373
374     char operator>>(ifstream &, class Medicina_Cantidad &);
375
376     #endif /* MEDICINA_CANTIDAD_H */
377
378     /*
379     * Proyecto: Pregunta01_Examen02_2022-2
380     * Archivo: Medicina_Cantidad.cpp
381     * Autor: J. Miguel Guanira E.
382     *
383     * Created on 23 de noviembre de 2022, 08:07 PM
384     */
385
386     #include <iostream>
387     #include <fstream>
388     #include <iomanip>
389     using namespace std;
390
391     #include "Medicina_Cantidad.h"
392
393     void Medicina_Cantidad::SetCantidad(int cantidad) {
394         this->cantidad = cantidad;
395     }
396
```

```
397     int Medicina_Cantidad::GetCantidad() const {
398         return cantidad;
399     }
400
401     void Medicina_Cantidad::SetCodigo(int codigo) {
402         this->codigo = codigo;
403     }
404
405     int Medicina_Cantidad::GetCodigo() const {
406         return codigo;
407     }
408
409     char operator>>(ifstream &arch, class Medicina_Cantidad &med_Cant){
410         int codMed, cant;
411         char c;
412         arch>>codMed>>c>>cant;
413         med_Cant.SetCodigo(codMed);
414         med_Cant.SetCantidad(cant);
415         c = arch.get();
416         return c;
417     }
418
419     void Medicina_Cantidad::imprimeMedicina(ofstream& arch) {
420         arch<<right<<"      "<<setw(10)<<codigo<<setw(10)<<cantidad<<endl;
421     }
422
```