

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

```

67     double Medicina::GetPrecio() const {
68         return precio;
69     }
70
71     void Medicina::SetDescripcion(const char* cad) {
72         if(descripcion!=nullptr) delete descripcion;
73         descripcion = new char[strlen(cad)+1];
74         strcpy(descripcion,cad);
75     }
76
77     void Medicina::GetDescripcion(char *cad) const {
78         if(descripcion == nullptr) cad[0] = 0;
79         else strcpy(cad,descripcion);
80     }
81
82     void Medicina::SetCodigo(int codigo) {
83         this->codigo = codigo;
84     }
85
86     int Medicina::GetCodigo() const {
87         return codigo;
88     }
89
90     void Medicina::operator=(const Medicina& orig) {
91         char cad[60];
92         codigo = orig.codigo;
93         orig.GetDescripcion(cad);
94         SetDescripcion(cad);
95         precio = orig.precio;
96     }
97
98     void operator>>(ifstream &arch, class Medicina&med) {
99         int cod;
100         char desc[60];
101         double prec;
102         arch>>cod;
103         if(arch.eof()) return;
104         arch.get();
105         arch.getline(desc,60,',' );
106         arch>>prec;
107         med.SetCodigo(cod);
108         med.SetDescripcion(desc);
109         med.SetPrecio(prec);
110     }
111
112     void operator <<(ofstream &arch,const class Medicina&med) {
113         char desc[60];
114         arch.precision(2);
115         arch<<fixed;
116         med.GetDescripcion(desc);
117         arch<<left<<setw(19)<<med.GetCodigo()<<setw(45)<<desc
118             <<right<<setw(10)<<med.GetPrecio()<<endl;
119     }
120
121     /*
122     * Proyecto: Pregunta01_Examen02_2022-2
123     * Archivo:  Paciente.h
124     * Autor:    J. Miguel Guanira E.
125     *
126     * Created on 23 de noviembre de 2022, 07:38 PM
127     */
128
129
130     #ifndef PACIENTE_H
131     #define PACIENTE_H
132

```

```

133     class Paciente {
134     private:
135         int dni;
136         char *nombre;
137         int fechaUltimaConsulta;
138         char *codigoDelMedicoDeUltimaConsulta;
139     public:
140         Paciente();
141         Paciente(const Paciente& orig);
142         virtual ~Paciente();
143         void SetFechaUltimaConsulta(int fechaUltimaConsulta);
144         int GetFechaUltimaConsulta() const;
145         void SetNombre(const char* nombre);
146         void GetNombre(char*) const;
147         void SetDni(int dni);
148         int GetDni() const;
149         void SetCodigoDelMedicoDeUltimaConsulta(const char*);
150         void GetCodigoDelMedicoDeUltimaConsulta(char*) const;
151         void operator = (const Paciente& orig);
152         void leerDatos(ifstream &);
153         void imprimirDatos(ofstream &arch);
154     };
155
156     #endif /* PACIENTE_H */
157
158     /*
159     * Proyecto: Pregunta01_Examen02_2022-2
160     * Archivo: Paciente.cpp
161     * Autor: J. Miguel Guanira E.
162     *
163     * Created on 23 de noviembre de 2022, 07:38 PM
164     */
165
166     #include <iostream>
167     #include <fstream>
168     #include <iomanip>
169     using namespace std;
170     #include <cstring>
171     #include "Paciente.h"
172
173     Paciente::Paciente() {
174         nombre = nullptr;
175         codigoDelMedicoDeUltimaConsulta = nullptr;
176         fechaUltimaConsulta = 30000000;
177         fechaUltimaConsulta = 0;
178     }
179
180     Paciente::Paciente(const Paciente& orig) {
181         nombre = nullptr;
182         codigoDelMedicoDeUltimaConsulta = nullptr;
183         fechaUltimaConsulta = 30000000;
184         *this = orig;
185     }
186
187     Paciente::~~Paciente() {
188         if (nombre != nullptr) delete nombre;
189     }
190
191     void Paciente::SetFechaUltimaConsulta(int fechaUltimaConsulta) {
192         this->fechaUltimaConsulta = fechaUltimaConsulta;
193     }
194
195     int Paciente::GetFechaUltimaConsulta() const {
196         return fechaUltimaConsulta;
197     }
198

```

```

199 void Paciente::SetNombre(const char* cad) {
200     if(nombre!=nullptr) delete nombre;
201     nombre = new char[strlen(cad)+1];
202     strcpy(nombre,cad);
203 }
204
205 void Paciente::GetNombre(char *cad) const {
206     if(nombre == nullptr) cad[0] = 0;
207     else strcpy(cad,nombre);
208 }
209
210 void Paciente::SetDni(int dni) {
211     this->dni = dni;
212 }
213
214 int Paciente::GetDni() const {
215     return dni;
216 }
217
218 void Paciente::SetCodigoDelMedicoDeUltimaConsulta(const char *cad) {
219     if(codigoDelMedicoDeUltimaConsulta!=nullptr)
220         delete codigoDelMedicoDeUltimaConsulta;
221     codigoDelMedicoDeUltimaConsulta = new char[strlen(cad)+1];
222     strcpy(codigoDelMedicoDeUltimaConsulta,cad);
223 }
224
225 void Paciente::GetCodigoDelMedicoDeUltimaConsulta(char *cad) const {
226     if(codigoDelMedicoDeUltimaConsulta == nullptr) cad[0] = 0;
227     else strcpy(cad,codigoDelMedicoDeUltimaConsulta);
228 }
229
230 void Paciente::operator=(const Paciente& orig) {
231     char cad[60];
232     dni = orig.dni;
233     orig.GetNombre(cad);
234     SetNombre(cad);
235     fechaUltimaConsulta = orig.fechaUltimaConsulta;
236     orig.GetCodigoDelMedicoDeUltimaConsulta(cad);
237     SetCodigoDelMedicoDeUltimaConsulta(cad);
238 }
239
240 void Paciente::leerDatos(istream&arch) {
241     int dd,mm,aa,fecha, dni;
242     char c, nomb[60],codMed[8];
243
244     arch>>dd;
245     if(arch.eof())return;
246     arch>>c>>mm>>c>>aa>>c>>dni>>c;
247     arch.getline(nomb,60,',');
248     arch.getline(codMed,60,',');
249     fecha = aa*10000+100*mm+dd;
250     SetDni(dni);
251     SetNombre(nomb);
252     SetFechaUltimaConsulta(fecha);
253     SetCodigoDelMedicoDeUltimaConsulta(codMed);
254 }
255
256 void Paciente::imprimirDatos(ofstream& arch) {
257     int fecha, dd,mm,aa;
258     fecha = fechaUltimaConsulta;
259     aa = fecha / 10000;
260     fecha %= 10000;
261     mm = fecha/100;
262     dd = fecha%100;
263     arch<<left<<setw(10)<<dni<<"    "<<left<<setw(40)<<nombre
264         <<setw(10)<<codigoDelMedicoDeUltimaConsulta

```

```

265         <<right<< " " <<setfill('0')<<setw(2)<<dd<<'/'<<setw(2)<<mm<<'/'
266         <<setw(4)<<aa<<setfill(' ')<<endl;
267     }
268
269     /*
270     * Proyecto: Pregunta01_Examen02_2022-2
271     * Archivo:  main.cpp
272     * Autor:    J. Miguel Guanira E.
273     *
274     * Created on 23 de noviembre de 2022, 07:24 PM
275     */
276
277     #include <iostream>
278     #include <iomanip>
279     using namespace std;
280     #include "ServicioDeSalud.h"
281     int main(int argc, char** argv) {
282         class ServicioDeSalud servicioDeSalud;
283         servicioDeSalud.leerMedicinas("Medicinas-Preg01.csv");
284         servicioDeSalud.imprimirMedicinas("PruebaMedicinas.txt");
285         servicioDeSalud.leerConsultas("Consultas-Preg01.csv");
286         servicioDeSalud.imprimePacientes("PruebaConsultas.txt");
287         servicioDeSalud.totalizar();
288         servicioDeSalud.imprimePacientes("PruebaConsultasTotal.txt");
289         return 0;
290     }
291
292     /*
293     * Proyecto: Pregunta01_Examen02_2022-2
294     * Archivo:  ServicioDeSalud.h
295     * Autor:    J. Miguel Guanira E. //miguel.guanira.
296     *
297     * Created on 8 de noviembre de 2023, 08:54 AM
298     */
299
300
301     #ifndef SERVICIODESALUD_H
302     #define SERVICIODESALUD_H
303     #include <list>
304     #include <map>
305     using namespace std;
306     #include "Medicina.h"
307     #include "Paciente_Medicina.h"
308
309     class ServicioDeSalud {
310     private:
311         map<int, Medicina> medicina;
312         list<Paciente_Medicina> paciente_medicina;
313     public:
314         void leerMedicinas(const char*);
315         void imprimirMedicinas(const char*);
316         void leerConsultas(const char*);
317         void imprimePacientes(const char*);
318         void totalizar();
319         list<class Paciente_Medicina>::iterator buscarPaciente(int);
320
321     };
322
323     #endif /* SERVICIODESALUD_H */
324
325     /*
326     * Proyecto: Pregunta01_Examen02_2022-2
327     * Archivo:  ServicioDeSalud.cpp
328     * Autor:    J. Miguel Guanira E. //miguel.guanira.
329     *
330     * Created on 8 de noviembre de 2023, 08:54 AM

```

```

331     */
332
333     #include <iostream>
334     #include <fstream>
335     #include <iomanip>
336     #include <list>
337     #include <iterator>
338     using namespace std;
339     #include "Medicina.h"
340     #include "ServicioDeSalud.h"
341
342     void ServicioDeSalud::leerMedicinas(const char*nombArch) {
343         ifstream arch(nombArch, ios::in);
344         if(not arch.is_open()){
345             cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
346             exit(1);
347         }
348         class Medicina med;
349         while(true){
350             arch>>med;
351             if(arch.eof())break;
352             medicina[med.GetCodigo()] = med;
353         }
354     }
355
356     void ServicioDeSalud::imprimirMedicinas(const char*nombArch) {
357         ofstream arch(nombArch, ios::out);
358         if(not arch.is_open()){
359             cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
360             exit(1);
361         }
362         arch<<setw(47)<<"LISTADO DE MEDICINAS"<<endl<<endl;
363         arch<<left<<setw(19)<<"CODIGO"<<setw(45)<<"DESCRIPCION"
364             <<right<<setw(11)<<"PRECIO"<<endl;
365         for(auto m: medicina)
366             arch<<m.second;
367     }
368
369     void ServicioDeSalud::leerConsultas(const char* nombArch) {
370         ifstream arch(nombArch, ios::in);
371         if(not arch.is_open()){
372             cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
373             exit(1);
374         }
375         class Paciente pac;
376         list<class Paciente_Medicina>::iterator itPacMed;
377
378         while(true){
379             class Paciente_Medicina pacMed;
380             pac.leerDatos(arch);
381             if(arch.eof())break;
382             itPacMed = buscarPaciente(pac.GetDni()); // Hay que implementar
383             if(itPacMed == paciente_medicina.end()){ // Si no lo encontramos
384                 pacMed = pac; // Hay que implementar
385                 pacMed.leerMedicinas(arch); // Hay que implementar
386                 paciente_medicina.push_back(pacMed);
387             }
388             else { // Si lo encontramos
389                 if(*itPacMed < pac) // Hay que implementar
390                     *itPacMed = pac; // Para simplificar // Hay que implementar
391                 itPacMed->leerMedicinas(arch);
392             }
393         }
394     }
395 }
396

```

```

397     list<Paciente_Medicina>::iterator ServicioDeSalud::buscarPaciente(int dni) {
398         for(list<Paciente_Medicina>::iterator iter=paciente_medicina.begin();
399             iter!=paciente_medicina.end();iter++)
400             if(iter->GetDni() == dni) return iter;
401         return paciente_medicina.end();
402     }
403
404     void ServicioDeSalud::imprimePacientes(const char*nombArch) {
405         ofstream arch(nombArch,ios::out);
406         if(not arch.is_open()){
407             cout<<"ERROR: No se pudo abrir el archivo "<<nombArch<<endl;
408             exit(1);
409         }
410         arch<<right<<setw(45)<<"LISTADO DE PACIENTES"<<endl<<endl;
411
412         for(Paciente_Medicina pM: paciente_medicina)
413             pM.imprimePacienteMedicina(arch); // Hay que implementar
414     }
415
416     void ServicioDeSalud::totalizar() {
417         // ESTO NO SE DEBE HACER PORQUE SE ESTÁ TRABAJANDO CON UN ACOPIA DEL DATO (pM)
418         // ESTO QUIERE DECIR QUE SE ACTUALIZA pM PERO NO EL DATO DE LA LISTA
419         //     for(Paciente_Medicina pM: paciente_medicina)
420         //         pM.totalizar(medicina);
421
422         for(list<Paciente_Medicina>::iterator it=paciente_medicina.begin();
423             it!=paciente_medicina.end(); it++){
424             it->totalizar(medicina);
425         }
426     }
427
428     /*
429     * Proyecto: Pregunta01_Examen02_2022-2
430     * Archivo:   Paciente_Medicina.h
431     * Autor:    J. Miguel Guanira E. //miguel.guanira.
432     *
433     * Created on 8 de noviembre de 2023, 08:54 AM
434     */
435
436
437     #ifndef PACIENTE_MEDICINA_H
438     #define PACIENTE_MEDICINA_H
439     #include <list>
440     #include <fstream>
441     using namespace std;
442     #include "Paciente.h"
443     #include "Medicina_Cantidad.h"
444     #include "Medicina.h"
445     #include <map>
446
447     class Paciente_Medicina {
448     private:
449         class Paciente paciente;
450         list<class Medicina_Cantidad>medicina_cantidad;
451         double totalDeGastos;
452     public:
453         Paciente_Medicina();
454         void SetTotalDeGastos(double totalDeGastos);
455         double GetTotalDeGastos() const;
456         int GetDni();
457         bool operator < (const class Paciente &pac);
458         void operator = (const class Paciente &pac);
459         void leerMedicinas(ifstream&);
460         list<Medicina_Cantidad>::iterator buscarMedicina(int);
461         void imprimePacienteMedicina(ofstream &arch);

```

```

463     void totalizar(map<int,Medicina>med);
464 };
465
466 #endif /* PACIENTE_MEDICINA_H */
467
468 /*
469  * Proyecto: Pregunta01_Examen02_2022-2
470  * Archivo:  Paciente_Medicina.cpp
471  * Autor:    J. Miguel Guanira E. //miguel.guanira.
472  *
473  * Created on 8 de noviembre de 2023, 08:54 AM
474  */
475
476 #include <iostream>
477 #include <iomanip>
478 using namespace std;
479
480 #include "Paciente_Medicina.h"
481 #include "Medicina.h"
482
483 Paciente_Medicina::Paciente_Medicina() {
484     totalDeGastos = 0;
485 }
486
487 void Paciente_Medicina::SetTotalDeGastos(double totalDeGastos) {
488     this->totalDeGastos = totalDeGastos;
489 }
490
491 double Paciente_Medicina::GetTotalDeGastos() const {
492     return totalDeGastos;
493 }
494
495 int Paciente_Medicina::GetDni() {
496     return paciente.GetDni();
497 }
498
499 bool Paciente_Medicina::operator<(const class Paciente &pac) {
500     return paciente.GetFechaUltimaConsulta() < pac.GetFechaUltimaConsulta();
501 }
502
503 void Paciente_Medicina::operator=(const class Paciente &pac) {
504     paciente = pac;
505 }
506
507 void Paciente_Medicina::leerMedicinas(ifstream&arch) {
508     class Medicina_Cantidad med_cant;
509     list<class Medicina_Cantidad>::iterator itMedCant;
510     int cant;
511
512     while(true){
513         arch>>med_cant; // Hay que implementar
514         itMedCant = buscarMedicina(med_cant.GetCodigo()); // Hay que implementar
515         if(itMedCant == medicina_cantidad.end()) // Si no lo encontré
516             medicina_cantidad.push_back(med_cant);
517         else { // Si lo encontré
518             cant = med_cant.GetCantidad(); // Hay que implementar
519             cant += itMedCant->GetCantidad();
520             itMedCant->SetCantidad(cant); // Hay que implementar
521         }
522         if(arch.get()=='\n')break;
523     }
524 }
525
526 list<Medicina_Cantidad>::iterator Paciente_Medicina::buscarMedicina(int codMed){
527     for(list<class Medicina_Cantidad>::iterator iter=medicina_cantidad.begin();
528         iter != medicina_cantidad.end(); iter++)

```



```

529         if(iter->GetCodigo()== codMed) return iter;
530         return medicina_cantidad.end();
531     }
532
533     void Paciente_Medicina::imprimePacienteMedicina(ofstream& arch) {
534         paciente.imprimirDatos(arch);
535         arch<<right<<"MEDICINAS: ";<<setw(10)<<"CODIGO"<<setw(13)<<"CANTIDAD"<<endl;
536         for(Medicina_Cantidad m: medicina_cantidad)
537             m.imprimeMedicina(arch); // Hay que implementar
538         arch.precision(2);
539         arch<<fixed;
540         arch<<"Total de gastos: "<<setw(10)<<totalDeGastos<<endl<<endl;
541     }
542
543     void Paciente_Medicina::totalizar(map<int, Medicina> med) {
544         int cod, cant;
545         map<int,Medicina>::iterator iterM;
546         // ESTO SI ES CORRECTO PORQUE SE ESTÁ ACTUALIZANDO UN ATRIBUTO DE LA
547         // CLASE Paciente_Medicina NO DE Medicina
548         for(auto m: medicina_cantidad){
549             cod = m.GetCodigo();
550             cant = m.GetCantidad();
551             iterM = med.find(cod);
552             if(iterM != med.end())
553                 totalDeGastos += cant * iterM->second.GetPrecio();
554         }
555     }
556
557     /*
558     * Proyecto: Pregunta01_Examen02_2022-2
559     * Archivo: Medicina_Cantidad.h
560     * Autor: J. Miguel Guanira E. //miguel.guanira.
561     *
562     * Created on 8 de noviembre de 2023, 08:54 AM
563     */
564
565     #ifndef MEDICINA_CANTIDAD_H
566     #define MEDICINA_CANTIDAD_H
567     #include <fstream>
568     class Medicina_Cantidad {
569     private:
570         int codigo;
571         int cantidad;
572     public:
573         void SetCantidad(int cantidad);
574         int GetCantidad() const;
575         void SetCodigo(int codigo);
576         int GetCodigo() const;
577         void imprimeMedicina(ofstream &arch);
578     };
579
580     void operator >>(ifstream &, class Medicina_Cantidad &);
581
582     #endif /* MEDICINA_CANTIDAD_H */
583
584     /*
585     * Proyecto: Pregunta01_Examen02_2022-2
586     * Archivo: Medicina_Cantidad.cpp
587     * Autor: J. Miguel Guanira E. //miguel.guanira.
588     *
589     * Created on 8 de noviembre de 2023, 08:54 AM
590     */
591
592     #include <iostream>
593     #include <iomanip>

```

```
595     using namespace std;
596
597     #include "Medicina_Cantidad.h"
598
599     void Medicina_Cantidad::SetCantidad(int cantidad) {
600         this->cantidad = cantidad;
601     }
602
603     int Medicina_Cantidad::GetCantidad() const {
604         return cantidad;
605     }
606
607     void Medicina_Cantidad::SetCodigo(int codigo) {
608         this->codigo = codigo;
609     }
610
611     int Medicina_Cantidad::GetCodigo() const {
612         return codigo;
613     }
614
615     void operator >>(ifstream &arch, class Medicina_Cantidad &med_cant){
616         int codMed, cant;
617         arch>>codMed;
618         arch.get();
619         arch>>cant;
620         med_cant.SetCodigo(codMed);
621         med_cant.SetCantidad(cant);
622     }
623
624     void Medicina_Cantidad::imprimeMedicina(ofstream& arch) {
625         arch<<right<<setw(14)<<' '<<setw(10)<<codigo<<setw(10)<<cantidad<<endl;
626     }
627
```