

```

1  /*
2  * Proyecto: Preg01_Ex02_2022-1
3  * Archivo:  Cliente.h
4  * Autor:    J. Miguel Guanira E.
5  *
6  * Created on 5 de julio de 2022, 05:42 PM
7  */
8
9
10 #ifndef CLIENTE_H
11 #define CLIENTE_H
12 #include <fstream>
13 using namespace std;
14
15 class Cliente {
16 private:
17     int dni;
18     char *nombre;
19     char tipo;
20     int telefono;
21     double descuento;
22 public:
23     Cliente(); // Además de los punteros, pone en cero el descuento
24     Cliente(const Cliente& orig);
25     virtual ~Cliente();
26     void operator = (const class Cliente &orig);
27     void SetTelefono(int telefono);
28     int GetTelefono() const;
29     void SetTipo(char tipo);
30     char GetTipo() const;
31     void SetNombre(const char* nombre);
32     void GetNombre(char*) const;
33     void SetDni(int dni);
34     int GetDni() const;
35     void SetDescuento(double descuento);
36     double GetDescuento() const;
37     // Lee una línea de la forma:
38     // 71984468,IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO B,935441620
39     // NO coloca valores al atributo "descuento"
40     void leerDatos(istream &);
41     // Imprime los atributos: dni, nombre, tipo y telefono
42     void imprimirDatos(ofstream &);
43 };
44
45 #endif /* CLIENTE_H */
46
47 /*
48 * Proyecto: Preg01_Ex02_2022-1
49 * Archivo:  Cliente.cpp
50 * Autor:    J. Miguel Guanira E.
51 *
52 * Created on 5 de julio de 2022, 05:42 PM
53 */
54
55 #include <iostream>
56 #include <iomanip>
57 using namespace std;
58 #include <cstring>
59 #include "Cliente.h"
60
61 Cliente::Cliente() {
62     nombre = nullptr;
63     tipo = '-'; // Por si no tiene tipo
64     descuento = 0.0;

```

```
65     }
66
67     Cliente::Cliente(const Cliente& orig) {
68         nombre = nullptr;
69         tipo = '-'; // Por si no tiene tipo
70         descuento = 0.0;
71         *this = orig;
72     }
73
74     Cliente::~~Cliente() {
75         if(nombre != nullptr) delete nombre;
76     }
77
78     void Cliente::operator=(const Cliente &orig) {
79         char cad[200];
80         dni = orig.dni;
81         orig.GetNombre(cad);
82         SetNombre(cad);
83         tipo = orig.tipo;
84         telefono = orig.telefono;
85         descuento = orig.descuento;
86     }
87
88     void Cliente::SetTelefono(int telefono) {
89         this->telefono = telefono;
90     }
91
92     int Cliente::GetTelefono() const {
93         return telefono;
94     }
95
96     void Cliente::SetTipo(char tipo) {
97         this->tipo = tipo;
98     }
99
100    char Cliente::GetTipo() const {
101        return tipo;
102    }
103
104    void Cliente::SetNombre(const char* cad) {
105        if(nombre!=nullptr) delete nombre;
106        nombre = new char[strlen(cad)+1];
107        strcpy(nombre,cad);
108    }
109
110    void Cliente::GetNombre(char*cad) const {
111        if(nombre == nullptr) cad[0] = 0;
112        else strcpy(cad,nombre);
113    }
114
115    void Cliente::SetDni(int dni) {
116        this->dni = dni;
117    }
118
119    int Cliente::GetDni() const {
120        return dni;
121    }
122
123    void Cliente::SetDescuento(double descuento) {
124        this->descuento = descuento;
125    }
126
127    double Cliente::GetDescuento() const {
128        return descuento;
```

```

129     }
130
131     void Cliente::leerDatos(ifstream &arch) {
132         //71984468,IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO B,935441620
133         char cad[60];
134         int n;
135         arch >> dni;
136         if(arch.eof())return;
137         arch.get();
138         arch.getline(cad,60,',');
139         arch >> telefono;
140         n = strlen(cad);
141         if(cad[n-2]==' ') {
142             tipo = cad[n-1];
143             cad[n-2] = 0;
144         }
145         SetNombre(cad);
146     }
147
148     void Cliente::imprimirDatos(ofstream &arch) {
149         arch.precision(2);
150         arch<<fixed;
151         arch<<left<<setw(10)<<dni<<setw(45)<<nombre<<setw(3)<<tipo
152             <<right<<setw(10)<<descuento<<setw(12)<<telefono<<endl;
153     }
154
155     /*
156     * Proyecto: Preg01_Ex02_2022-1
157     * Archivo:  Producto.h
158     * Autor:    J. Miguel Guanira E.
159     *
160     * Created on 6 de julio de 2022, 07:58 AM
161     */
162
163
164     #ifndef PRODUCTO_H
165     #define PRODUCTO_H
166     #include <fstream>
167     using namespace std;
168
169     class Producto {
170     private:
171         int codigo;
172         char *descripcion;
173         double descuento;
174         double precioUnitario;
175     public:
176         Producto();
177         Producto(const Producto& orig);
178         virtual ~Producto();
179         void operator = (const class Producto &orig);
180         void SetPrecioUnitario(double precioUnitario);
181         double GetPrecioUnitario() const;
182         void SetDescripcion(const char* descripcion);
183         void GetDescripcion(char* descripcion) const;
184         void SetDescuento(double descuento);
185         double GetDescuento() const;
186         void SetCodigo(int codigo);
187         int GetCodigo() const;
188         // Lee una línea de la forma:
189         // 306824,FLAN LALA SABOR VAINILLA 100GR,5.88,
190         // 186026,YOGHURT NATURAL LIGHT LALA 1KG,%4.7,25.86,
191         void leerDatos(ifstream &);
192         // Imprime los atributos: código, descripción y precio unitario

```

```
193     void imprimirDatos(ofstream &);
194
195 };
196
197 #endif /* PRODUCTO_H */
198
199 /*
200  * Proyecto: Preg01_Ex02_2022-1
201  * Archivo:  Producto.cpp
202  * Autor:    J. Miguel Guanira E.
203  *
204  * Created on 6 de julio de 2022, 07:58 AM
205  */
206
207 #include <iostream>
208 #include <iomanip>
209 #include <ratio>
210 using namespace std;
211 #include <cstring>
212 #include "Producto.h"
213
214 Producto::Producto() {
215     descripcion = nullptr;
216     descuento = 0.0;
217 }
218
219 Producto::Producto(const Producto& orig) {
220     descripcion = nullptr;
221     descuento = 0.0;
222     *this = orig;
223 }
224
225 Producto::~~Producto() {
226     if(descripcion != nullptr) delete descripcion;
227 }
228
229 void Producto::operator=(const class Producto &orig) {
230     char cad[60];
231     codigo = orig.codigo;
232     orig.GetDescripcion(cad);
233     SetDescripcion(cad);
234     precioUnitario = orig.precioUnitario;
235     descuento = orig.GetDescuento();
236 }
237
238 void Producto::SetPrecioUnitario(double precioUnitario) {
239     this->precioUnitario = precioUnitario;
240 }
241
242 double Producto::GetPrecioUnitario() const {
243     return precioUnitario;
244 }
245
246 void Producto::SetDescripcion(const char* cad) {
247     if(descripcion!=nullptr) delete descripcion;
248     descripcion = new char[strlen(cad)+1];
249     strcpy(descripcion,cad);
250 }
251
252 void Producto::GetDescripcion(char* cad) const {
253     if(descripcion == nullptr) cad[0] = 0;
254     else strcpy(cad,descripcion);
255 }
256
```

```

257
258 void Producto::SetDescuento(double descuento) {
259     this->descuento = descuento;
260 }
261
262 double Producto::GetDescuento() const {
263     return descuento;
264 }
265
266 void Producto::SetCodigo(int codigo) {
267     this->codigo = codigo;
268 }
269
270 int Producto::GetCodigo() const {
271     return codigo;
272 }
273
274 void Producto::leerDatos(istream &arch) {
275     // 306824,FLAN LALA SABOR VAINILLA 100GR,5.88,
276     // 186026,YOGHURT NATURAL LIGHT LALA 1KG,%4.7,25.86,
277     char cad[60];
278     arch >> codigo;
279     if(arch.eof())return;
280     arch.get();
281     arch.getline(cad,60,',');
282     SetDescripcion(cad);
283     arch >> precioUnitario;
284     if(arch.fail()){
285         arch.clear();
286         arch.get();
287         arch >> descuento;
288         arch.get();
289         arch >> precioUnitario;
290     }
291     else descuento = 0.0;
292     arch.get();
293 }
294
295 void Producto::imprimirDatos(ofstream &arch) {
296     arch.precision(2);
297     arch<<fixed;
298     arch<<left<<setw(10)<<codigo<<setw(45)<<descripcion
299         <<right<<setw(10)<<descuento<<setw(10)<<precioUnitario<<endl;
300 }
301
302 /*
303  * Proyecto: Preg01_Ex02_2022-1
304  * Archivo: Pedido.h
305  * Autor: J. Miguel Guanira E.
306  *
307  * Created on 6 de julio de 2022, 08:16 AM
308  */
309
310
311 #ifndef PEDIDO_H
312 #define PEDIDO_H
313 #include <fstream>
314 using namespace std;
315
316 class Pedido {
317 private:
318     int codigo; // Codigo del producto
319     double cantidad; // Cantidad del producto solicitado
320     int fecha; // Fecha del pedido;

```

```

321     double subTotal;        // Precio unitario X cantidad X descuentos
322 public:
323     void SetSubTotal(double subTotal);
324     double GetSubTotal() const;
325     void SetFecha(int fecha);
326     int GetFecha() const;
327     void SetCantidad(double cantidad);
328     double GetCantidad() const;
329     void SetDescuento(double descuento);
330     double GetDescuento() const;
331     void SetCodigo(int codigo);
332     int GetCodigo() const;
333     void dateTOddmmaa(int &, int &, int &);
334     void operator=(const class Producto &);
335     bool operator<(const class Pedido &);
336
337     // Lee una línea de la forma:
338     // 8.13,71984468,29/03/2022 (cantidad solicitada, dni del cliente, fecha
339     int leerDatos(ifstream &);
340     // Imprime los atributos: codigo del producto, descuento del producto,
341     //                                descuento del cliente, cantidad, fecha y subtotal
342     void imprimirDatos(ofstream &);
343 };
344
345 #endif /* PEDIDO_H */
346
347 /*
348  * Proyecto: Preg01_Ex02_2022-1
349  * Archivo:  Pedido.cpp
350  * Autor:    J. Miguel Guanira E.
351  *
352  * Created on 6 de julio de 2022, 08:16 AM
353  */
354
355 #include <iostream>
356 #include <fstream>
357 #include <iomanip>
358 using namespace std;
359 #include <cstring>
360 #include "Pedido.h"
361 #include "Producto.h"
362
363 void Pedido::SetSubTotal(double subTotal) {
364     this->subTotal = subTotal;
365 }
366
367 double Pedido::GetSubTotal() const {
368     return subTotal;
369 }
370
371 void Pedido::SetFecha(int fecha) {
372     this->fecha = fecha;
373 }
374
375 int Pedido::GetFecha() const {
376     return fecha;
377 }
378
379 void Pedido::SetCantidad(double cantidad) {
380     this->cantidad = cantidad;
381 }
382
383 double Pedido::GetCantidad() const {
384     return cantidad;

```

```

385     }
386
387     void Pedido::SetCodigo(int codigo) {
388         this->codigo = codigo;
389     }
390
391     int Pedido::GetCodigo() const {
392         return codigo;
393     }
394
395     void Pedido::operator=(const class Producto &prod) {
396         codigo = prod.GetCodigo();
397         subTotal = cantidad * prod.GetPrecioUnitario()*prod.GetDescuento();
398     }
399
400     bool Pedido::operator<(const class Pedido&ped) {
401         return fecha < ped.fecha;
402     }
403
404
405     int Pedido::leerDatos(istream &arch) {
406         // 8.13,71984468,29/03/2022
407         int dni, dd,mm,aa;
408         char c;
409         arch >> cantidad;
410         arch.get();
411         arch >> dni;
412         arch.get();
413         arch >>dd>>c>>mm>>c>>aa;
414         fecha = dd + mm*100 + aa*10000;
415         arch.get();
416         return dni;
417     }
418
419     void Pedido::imprimirDatos(ofstream &arch) {
420         arch.precision(2);
421         arch<<fixed;
422         int dd,mm,aa;
423         dateTOddmmaa(dd,mm,aa);
424         arch<<right<<setfill('0')<<setw(2)<<dd<< '/'<<setw(2)<<mm<< '/'<<setfill(' ')
425             <<setw(4)<<aa<<setw(10)<<codigo<<setw(10)<<cantidad
426             <<setw(10)<<subTotal<<endl;
427     }
428
429     void Pedido::dateTOddmmaa(int&dd, int&mm, int&aa) {
430         int f = fecha;
431         aa = f/10000;
432         f %= 10000;
433         mm = f/100;
434         dd = f%100;
435     }
436
437     /*
438     * Proyecto: Preg01_Ex02_2022-1
439     * Archivo:  RegCliente.h
440     * Autor:    J. Miguel Guanira E.
441     *
442     * Created on 6 de julio de 2022, 12:06 PM
443     */
444
445
446     #ifndef REGCLIENTE_H
447     #define REGCLIENTE_H
448     #include <fstream>

```

```
449 #include <list>
450 using namespace std;
451 #include "Cliente.h"
452 #include "Pedido.h"
453
454 class RegCliente {
455 private:
456     class Cliente cliente;
457     list <class Pedido> pedidos;
458 public:
459     void leerDatosCliente(ifstream &);
460     void imprimeDatos(ofstream &);
461     double GetDescuento();
462     int GetDni();
463     void agregarPedido(class Pedido &);
464     void ordenaPedidos();
465 };
466
467 #endif /* REGCLIENTE_H */
468
469 /*
470  * Proyecto: Preg01_Ex02_2022-1
471  * Archivo: RegCliente.cpp
472  * Autor: J. Miguel Guanira E.
473  *
474  * Created on 6 de julio de 2022, 12:06 PM
475  */
476
477 #include <iostream>
478 #include <iomanip>
479 using namespace std;
480
481 #include "RegCliente.h"
482
483 void RegCliente::leerDatosCliente(ifstream &arch) {
484     cliente.leerDatos(arch);
485     switch(cliente.GetTipo()){
486         case 'A':
487             cliente.SetDescuento(23.5);
488             break;
489         case 'B':
490             cliente.SetDescuento(16.8);
491             break;
492         case 'C':
493             cliente.SetDescuento(8.3);
494             break;
495     }
496 }
497
498 void RegCliente::imprimeDatos(ofstream &arch) {
499     cliente.imprimirDatos(arch);
500     for(list<class Pedido>::iterator it = pedidos.begin();
501         it != pedidos.end(); it++)
502         (*it).imprimirDatos(arch);
503 }
504
505 double RegCliente::GetDescuento() {
506     return cliente.GetDescuento();
507 }
508
509 int RegCliente::GetDni() {
510     return cliente.GetDni();
511 }
512
```



```

513 void RegCliente::agregarPedido(Pedido&ped) {
514     pedidos.push_back(ped);
515 }
516
517 void RegCliente::ordenaPedidos() {
518     pedidos.sort();
519 }
520
521 /*
522  * Proyecto: Preg01_Ex02_2022-1
523  * Archivo: Empresa.h
524  * Autor: J. Miguel Guanira E.
525  *
526  * Created on 6 de julio de 2022, 10:33 AM
527  */
528
529
530 #ifndef EMPRESA_H
531 #define EMPRESA_H
532
533 #include <fstream>
534 #include <list>
535 using namespace std;
536 #include "Producto.h"
537 #include "RegCliente.h"
538 class Empresa {
539 private:
540     list<class Producto> lstProductos;
541     list<class RegCliente> lstClientes;
542 public:
543     void leerClientes(const char*);
544     void imprimirClientes(const char*);
545     void leerPedidos(const char*);
546     void imprimirProductos(const char*);
547     void ordenarPedidos();
548     list<class Producto>::iterator buscarProducto(int);
549     list<class RegCliente>::iterator buscarCliente(int);
550 };
551
552 #endif /* EMPRESA_H */
553
554 /*
555  * Proyecto: Preg01_Ex02_2022-1
556  * Archivo: Empresa.cpp
557  * Autor: J. Miguel Guanira E.
558  *
559  * Created on 6 de julio de 2022, 10:33 AM
560  */
561
562 #include <iostream>
563 #include <fstream>
564 #include <iomanip>
565 #include <iterator>
566 using namespace std;
567
568 #include "Empresa.h"
569 #include "RegCliente.h"
570 #include "Producto.h"
571 #include "Pedido.h"
572
573 void Empresa::leerClientes(const char*nombArch) {
574     ifstream arch(nombArch,ios::in);
575     if(not arch.is_open()){
576         cout << "ERROR: no se pudo abrir el archivo " << nombArch << endl;

```

```
577         exit(1);
578     }
579     class RegCliente regCliente;
580     while(true){
581         regCliente.leerDatosCliente(arch);
582         if(arch.eof()) break;
583         lstClientes.push_back(regCliente);
584     }
585 }
586
587 void Empresa::imprimirClientes(const char*nombArch) {
588     ofstream arch(nombArch,ios::out);
589     if(not arch.is_open()){
590         cout << "ERROR: no se pudo abrir el archivo " << nombArch << endl;
591         exit(1);
592     }
593
594     for(list<class RegCliente>::iterator it = lstClientes.begin();
595         it != lstClientes.end(); it++){
596         (*it).imprimeDatos(arch);
597     }
598 }
599
600 void Empresa::leerPedidos(const char* nombArch) {
601     ifstream arch(nombArch,ios::in);
602     if(not arch.is_open()){
603         cout << "ERROR: no se pudo abrir el archivo " << nombArch << endl;
604         exit(1);
605     }
606     class Producto producto;
607     class Pedido pedido;
608     int dni;
609     list<class Producto>::iterator itProducto;
610     list<class RegCliente>::iterator itCliente;
611     while(true){
612         producto.leerDatos(arch);
613         if(arch.eof()) break;
614         itProducto = buscarProducto(producto.GetCodigo());
615         if(itProducto == lstProductos.end())
616             lstProductos.push_back(producto);
617         dni = pedido.leerDatos(arch);
618         itCliente = buscarCliente(dni);
619         if(itCliente != lstClientes.end()){
620             pedido.SetCodigo(producto.GetCodigo());
621             pedido.SetSubTotal(producto.GetPrecioUnitario()*
622                 pedido.GetCantidad()*
623                 (1-producto.GetDescuento()/100)*
624                 (1-itCliente->GetDescuento()/100));
625         }
626         itCliente->agregarPedido(pedido);
627     }
628 }
629
630 list<Producto>::iterator Empresa::buscarProducto(int cod) {
631     for(list<class Producto>::iterator it = lstProductos.begin();
632         it != lstProductos.end(); it++){
633         if (cod == (*it).GetCodigo()) return it;
634     }
635     return lstProductos.end();
636 }
637
638 list<RegCliente>::iterator Empresa::buscarCliente(int dni) {
639     int Dni;
640     for(list<class RegCliente>::iterator it = lstClientes.begin();
641         it != lstClientes.end(); it++){
```

```
641         Dni = (*it).GetDni();
642         if (dni == Dni) return it;
643     }
644     return lstClientes.end();
645 }
646
647 void Empresa::imprimirProductos(const char*nombArch) {
648     ofstream arch(nombArch,ios::out);
649     if(not arch.is_open()){
650         cout << "ERROR: no se pudo abrir el archivo " << nombArch << endl;
651         exit(1);
652     }
653
654     for(list<class Producto>::iterator it = lstProductos.begin();
655         it != lstProductos.end(); it++){
656         (*it).imprimirDatos(arch);
657     }
658 }
659
660 void Empresa::ordenarPedidos() {
661     for(list<class RegCliente>::iterator it = lstClientes.begin();
662         it != lstClientes.end(); it++){
663         (*it).ordenaPedidos();
664     }
665
666     /*
667     * Proyecto: Preg01_Ex02_2022-1
668     * Archivo:  main.cpp
669     * Autor:    J. Miguel Guanira E.
670     *
671     * Created on 5 de julio de 2022, 05:42 PM
672     */
673
674     #include <iostream>
675     #include <iomanip>
676     using namespace std;
677     #include "Empresa.h"
678     int main(int argc, char** argv) {
679         class Empresa empresa;
680         empresa.leerClientes("clientes.csv");
681         empresa.imprimirClientes("PruebaClientes.txt");
682         empresa.leerPedidos("Pedidos.csv");
683         empresa.ordenarPedidos();
684         empresa.imprimirProductos("ReporteDeProductos.txt");
685         empresa.imprimirClientes("ReporteDeClientes.txt");
686
687
688         return 0;
689     }
690 }
```