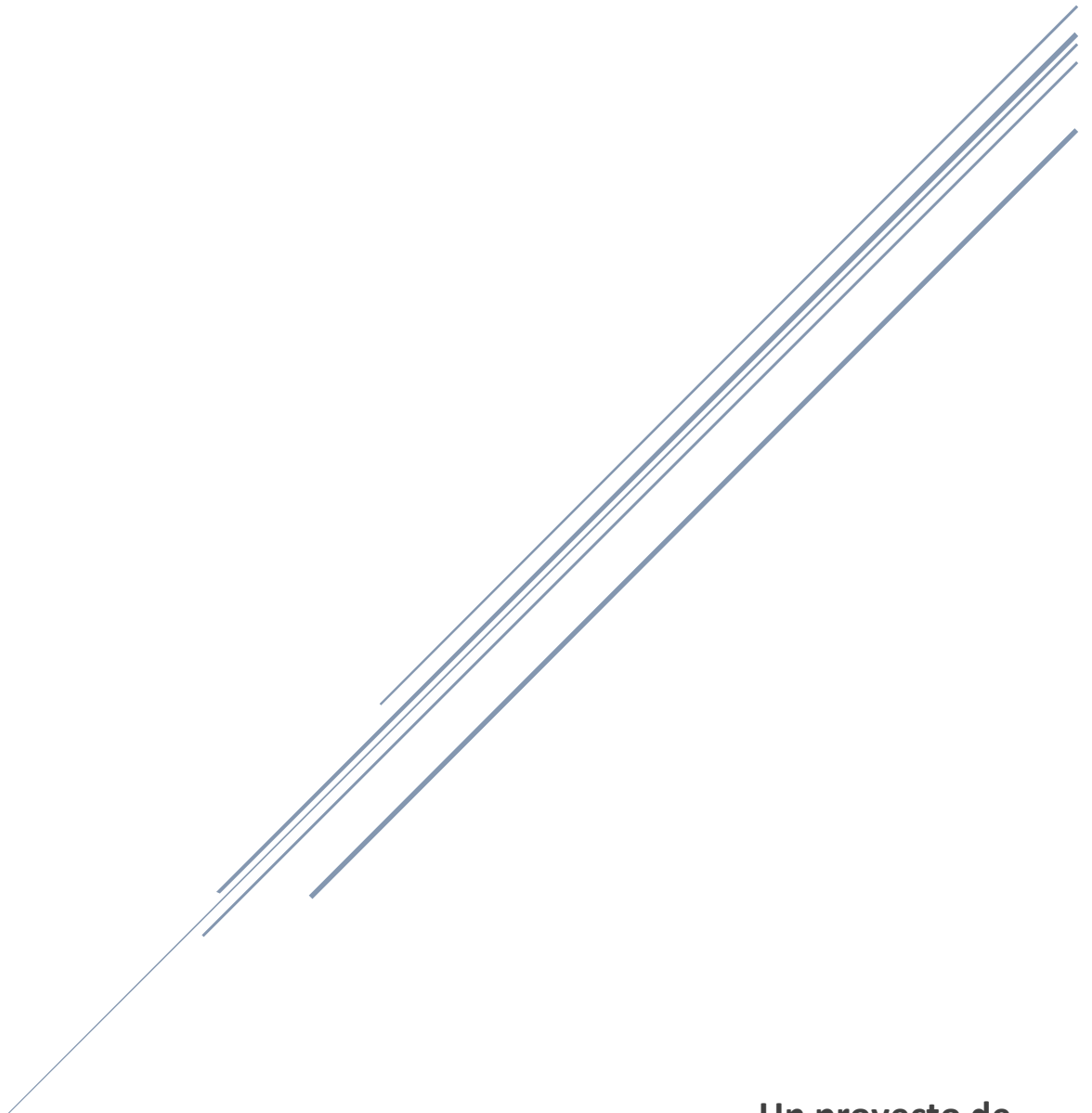


MANUAL DEL SISTEMA

PIICSAGIFT



Un proyecto de
The Piicsa Software Company



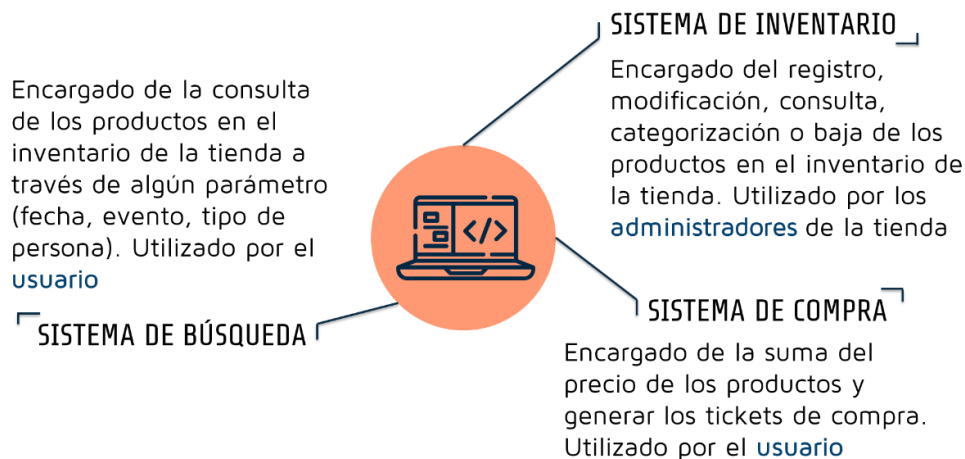
TABLA DE CONTENIDO

Estructura general del sistema	1
Sistema de Inventario.....	1
Sistema de Búsqueda	2
Sistema de Compra (carrito).....	3
Diagramas de clases	4
Capturas de pantalla del código	4
Diagrama general del programa.....	17



Estructura general del sistema

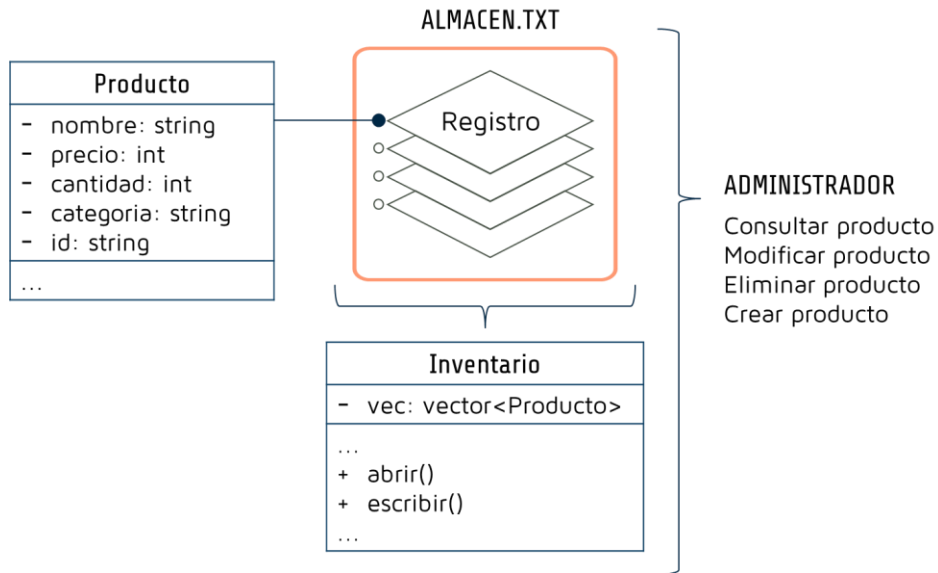
El sistema general está conformado por tres módulos básicos: Inventario, Compras y Búsqueda. El primero será empleado únicamente por los administradores designados por la tienda o establecimiento donde se encuentre implantado el software, mientras que los últimos dos serán empleados por los clientes del establecimiento, como se muestra a continuación:



Sistema de Inventario

Se conforma de una clase estática que contiene un vector del tipo Producto (un objeto personalizado donde se guardarán la información de los productos vendidos en la tienda) y de varios métodos con los que se podrá acceder al inventario de la tienda y manipular sus registros.

En la siguiente imagen se muestra la forma en que está estructurado este sistema:



Los registros del inventario se encuentran almacenados de manera permanente en un archivo de texto llamado *Almacen.txt*, al iniciar el programa el inventario ejecuta el método *abrir ()* que va a leer línea por línea, registro por registro, el contenido de dicho archivo (*Almacen.txt*) para irlo guardando en nuevos registros, pero ahora dentro del vector llamado *vec* del tipo *Producto*. De esta forma, en lugar de tener que abrir el archivo de *Almacen.txt* cada vez que se quiera realizar una consulta o modificación al inventario, solo será necesario abrirlo una única vez para vaciar su contenido al vector y desde ahí trabajar su contenido.

Así mismo, al terminar el programa - en caso de que se haya modificado el contenido del inventario - se ejecutará otra instrucción llamada *escribir ()* que sobrescribirá el contenido de *Almacen.txt* con lo que en ese momento se encuentre dentro del vector.



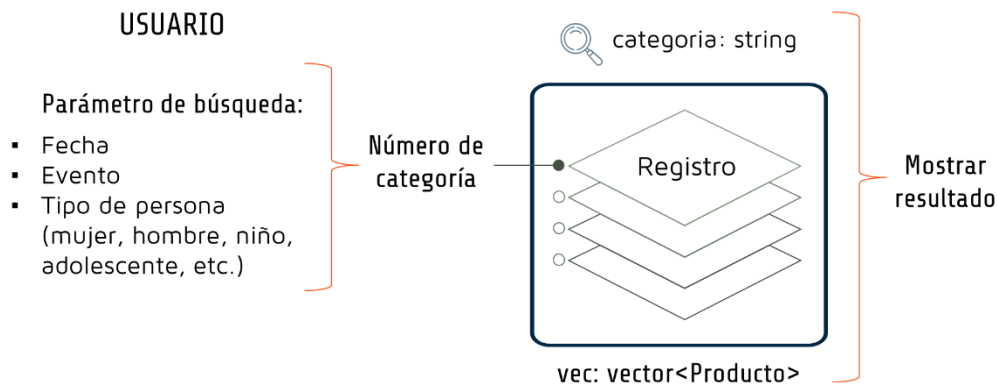
Sistema de Búsqueda

A través de un pequeño catálogo de eventos, fechas y tipos de personas representadas cada una por un número del 1 al 10, que se muestra a continuación, el sistema hace una búsqueda de aquellos productos cuya categoría coincida con el número que ingresó el usuario en su búsqueda.

Temática	Categoría
Día de la madre	1
Día del padre	2
Día del niño	3
Navidad	4
Baby Shower	5
14 de febrero	6
Día del maestro	7

Regalos comunes	8
Hombre	9
Mujer	10

Así, por ejemplo, si un usuario selecciona el número “1” en el menú de búsqueda de productos, el sistema buscará aquellos productos cuya categoría contenga el número “1” y los mostrará en pantalla.

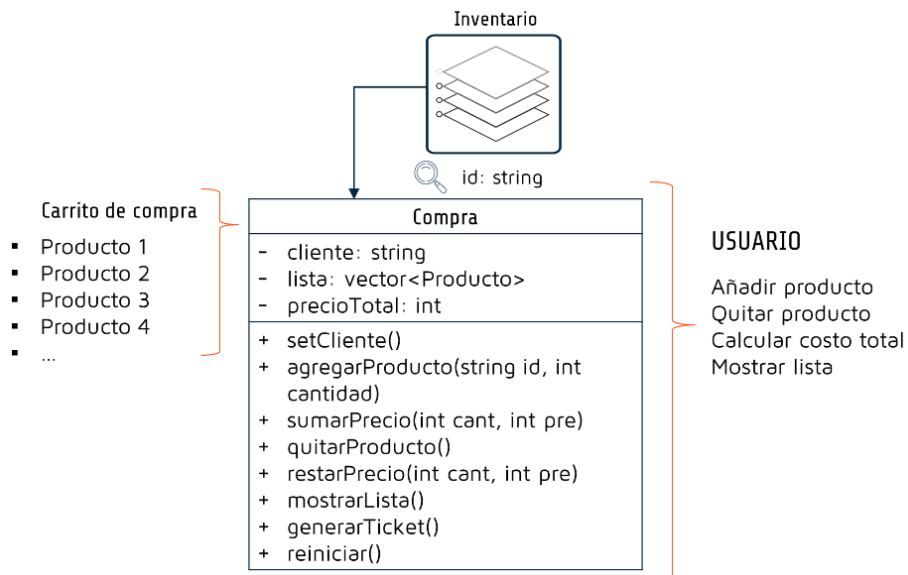


Este sistema se encuentra dentro del sistema de inventario.



Sistema de Compra (carrito)

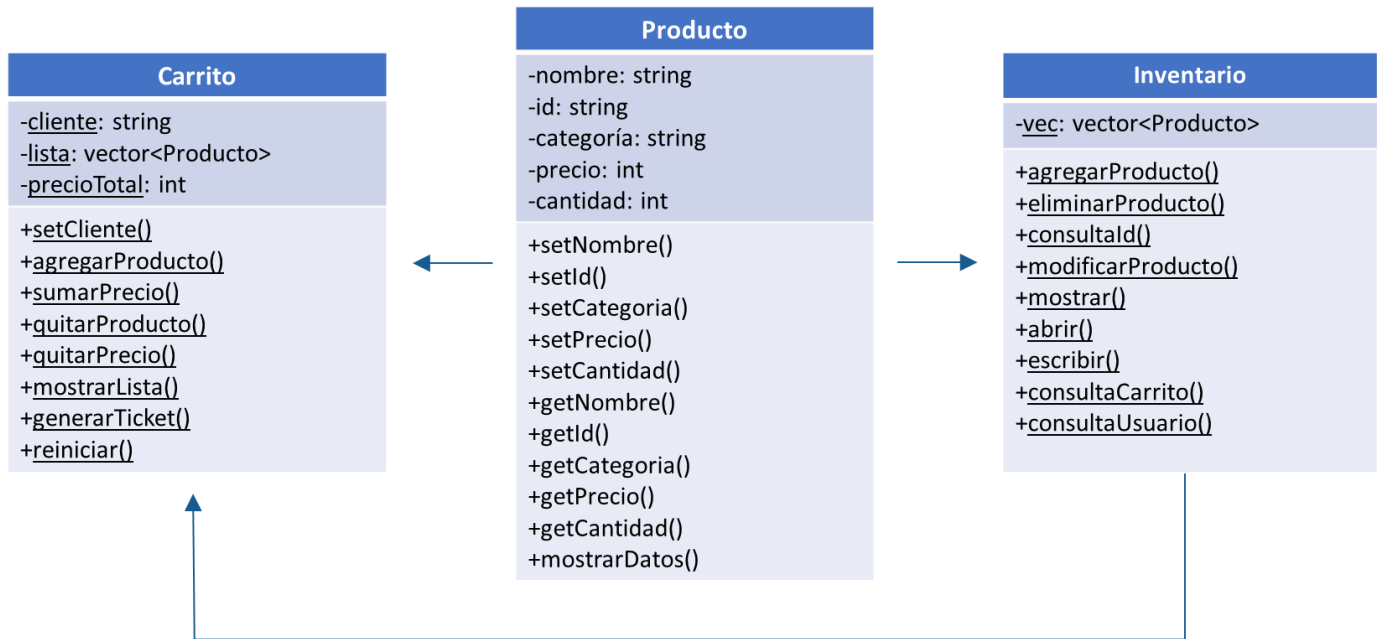
Al igual que el sistema de inventario, el sistema de compra cuenta con un vector del tipo Producto llamado *lista*, donde se irán guardando los productos que el usuario quiera ingresar al carrito de compras, pasando primero por una búsqueda en el inventario del sistema para después ser ingresados a dicho vector (*lista*). Al mismo tiempo, al agregar o quitar productos de carrito de compra, el sistema irá calculando el total a pagar y, una vez confirmada la compra, generará el ticket de compra del cliente.



El ticket de compra se generará en un archivo de texto llamado *Ticket.txt* que contendrá toda la información de la compra.



Diagramas de clases



Capturas de pantalla del código

Main2 (programa principal)

```

Carrito.cpp × Inventario.cpp × Main2.cpp × Producto.cpp ×
1  #include<iostream>
2  #include<algorithm>
3  #include<string>
4  #include"Carrito.cpp"
5  using namespace std;
6
7  void admin();
8  void usuario();
9
10 int main(){
11     Inventario::abrir();
12     int opc;
13     int clave;
14
15     do{
16         system("cls");
17         cout<<"Bienvenido a PiicsaGift\n";
18         cout<<"Es usted cliente o administrador?\n";
19         cout<<"[1] Cliente\n";
20         cout<<"[2] Administrador\n";
21         cout<<"[3] Salir\n";
22         cin>>opc;
    
```

```

23
24 □ if(opc == 1){
25     usuario();//Se ejecuta el sistema del usuario
26 }else if(opc == 2){
27     system("cls");
28     cout<<"Ingrese la clave de acceso\n";
29     cout<<"(teclea 0 para regresar)\n";
30     cin>>clave;
31
32 □     if(clave == 1984){
33         admin();//Se ejecuta el sistema del administrador
34     }else if(clave == 0){
35         opc = 5;//para que se vuelva a repetir el bucle
36     }else{
37         system("cls");
38         cout<<"Clave de acceso invalida\n";
39         cout<<"Acceso denegado :(\n";
40         system("pause");
41         opc = 5;//para que se vuelva a repetir el bucle
42     }
43 }else if(opc == 3){
44     break;
45 }else{
46     cout<<"Opcion fuera de rango X_X\n";
47     system("pause");
48 }
49 }while((opc != 1) || (opc != 2));//while opc!=3 queda mejor
50 Inventario::escribir();
51 return 0;
52 };
53
54 □ void admin(){
55     int opc;
56
57 □     do{
58         system("cls");
59         cout<<"Usted ha ingresado como administrador\n";
60         cout<<"Teclee la opcion deseada...\n";
61         cout<<"[1] Dar de alta producto\n";
62         cout<<"[2] Modificar producto\n";
63         cout<<"[3] Dar de baja producto\n";
64         cout<<"[4] Consulta individual por ID\n";
65         cout<<"[5] Consulta general\n";
66         cout<<"[6] Salir\n";

```

```

67     cin>>opc;
68
69     switch(opc){
70     case 1:
71         Inventario::agregarProducto();
72         break;
73     case 2:
74         Inventario::modificarProducto();
75         break;
76     case 3:
77         Inventario::eliminarProducto();
78         break;
79     case 4:
80         Inventario::consultaId();
81         break;
82     case 5:
83         Inventario::mostrar();
84         break;
85     case 6:
86         break;
87     default:
88         cout<<"Opcion fuera de rango X X\n";
89         system("pause");
90     }
91
92     }while(opc != 6);
93 }
94 void usuario(){
95     int opc = 0;
96     Carrito::setCliente();
97     do{
98         system("cls");
99         cout<<"Existe un regalo para todo, para buscar el suyo\n";
100        cout<<"ingrese el evento, fecha o tipo de persona correspondiente:\n";
101        cout<<"[1] Dia de la madre      [6] 14 de febrero\n";
102        cout<<"[2] Dia del padre        [7] Dia del maestro\n";
103        cout<<"[3] Dia del nin~o          [8] Regalos comunes\n";
104        cout<<"[4] Navidad                [9] Hombre\n";
105        cout<<"[5] Baby shower           [10] Mujer\n\n";
106        cout<<"[11] Ir al carrito de compra\n";
107        cin>>opc;
108
109        if(opc > 0 && opc < 11){
110            system("cls");

```

```

111     int op = 0;
112     int cant = 0;
113     cout<<"Ingrese el id del producto que\n";
114     cout<<"desea agregar al carrito\n";
115     cout<<"(teclea 0 para regresar)\n";
116     cout<<"~~~~~ Productos recomendados ~~~~~\n";
117     Inventario::consultaUsuario(to_string(opc));//se buscan los productos que
118     cin>>op;
119     if(op != 0){
120         int op2;
121         cout<<"Ingrese la cantidad de producto: ";
122         cin>>cant;
123         Carrito::agregarProducto(to_string(op), cant);//se busca el producto
124         //encuentre o no el registro no hay pido
125         cout<<"Desea seguir comprando?\n";
126         cout<<"[1] Si\n";
127         cout<<"[2] No, finalizar compra\n";
128         cin>>op2;
129         if(op2 == 1){
130             op = 0;//se vuelve a inicializar la variable
131             //no pasa nada
132         }else if(op2 == 2){//se muestra contenido del carrito
133             int op3 = 4;
134             do{
135                 Carrito::mostrarLista();
136                 cout<<"Desea proceder con el pago?\n";
137                 cout<<"[1] Si\n";
138                 cout<<"[2] No, seguir comprando\n";
139                 cout<<"[3] Quitar un producto del carrito\n";
140                 cin>>op3;
141
142                 if(op3 == 1){
143                     system("cls");
144                     Carrito::generarTicket();
145                     cout<<"Su ticket ha sido generado\n";
146                     cout<<"Gracias por su compra, hasta luego :)\n";
147                     system("pause");
148                     opc = 11;
149                     break;
150                 }else if(op3 == 2){
151                     break;
152                 }else if(op3 == 3){
153                     Carrito::quitarProducto();
154                     op3 = 4;//para que se vuelva a mostrar

```



```

155         }else{
156             op3 = 4; //para que se vuelva a mostrar
157             cout<<"Opcion fuera de rango X_X\n";
158             system("pause");
159         }
160     }while(op3 > 3);
161
162     }else{
163         cout<<"Opcion fuera de rango X_X\n";
164         system("pause");
165     }
166 }
167 }else if( opc == 11){
168     int op4 = 4;
169     do{
170         Carrito::mostrarLista();
171         cout<<"Desea proceder con el pago?\n";
172         cout<<"[1] Si\n";
173         cout<<"[2] No, seguir comprando\n";
174         cout<<"[3] Quitar un producto del carrito\n";
175         cin>>op4;
176
177         if(op4 == 1){
178             system("cls");
179             Carrito::generarTicket();
180             cout<<"Su ticket ha sido generado\n";
181             cout<<"Gracias por su compra, hasta luego :)\n";
182             system("pause");
183             break;
184         }else if(op4 == 2){
185             opc = 2; //para que se vuelva a mostrar el menu de ini
186             break;
187         }else if(op4 == 3){
188             Carrito::quitarProducto();
189             op4 = 4; //para que se vuelva a mostrar
190         }else{
191             op4 = 4; //para que se vuelva a mostrar
192             cout<<"Opcion fuera de rango X_X\n";
193             system("pause");
194         }
195     }while(op4 > 3);
196 }else{
197     cout<<"Opcion fuera de rango X_X\n";
198     system("pause");

```

```

199     }
200     }while(opc != 11);
201
202     Carrito::reiniciar();
203 }
204

```

Inventario (clase estática)

```

Inventario.cpp × Carrito.cpp × Producto.cpp ×
1  #include<iostream>
2  #include<vector>
3  #include<algorithm>
4  #include<fstream>
5  #include<string>
6  #include<sstream>
7  #include<cstdlib>
8  #include<stdlib.h>
9  #include"Producto.cpp"
10 using namespace std;
11
12 class Inventario{
13     private:
14         static vector<Producto> vec;
15     public:
16         static void agregarProducto(){
17             system("cls");
18             string nom;
19             string id;
20             string cat;
21             int pre;
22             int can;
23
24             cout<<"Nombre: ";
25             cin.ignore();
26             getline(cin, nom);//Validar longitud de la cadena
27             cout<<"ID: ";
28             cin>>id;
29             cout<<"Categoria/s: ";
30             cin>>cat;
31             cout<<"Precio: ";
32             cin>>pre;

```

```

33     cout<<"Cantidad: ";
34     cin>>can;
35     Producto ob1;
36     ob1.setNombre(nom);
37     ob1.setId(id);
38     ob1.setCategoria(cat);
39     ob1.setPrecio(pre);
40     ob1.setCantidad(can);
41
42     vec.push_back(ob1);
43
44     cout<<"Producto registrado :)\n";
45     system("pause");
46 }
47 static void eliminarProducto(){
48     bool aux = true;
49     string id;
50     system("cls");
51     cout<<"Ingrese el ID del producto a eliminar: ";
52     cin>>id;
53
54     for(int i = 0; i < vec.size(); i++){
55         if(vec[i].getId() == id){
56             aux = false;
57             cout<<"Producto '"<<vec[i].getNombre()<<"' eliminado :)\n";
58             vec.erase(vec.begin() + i);
59             break;
60         }
61     }
62     if(aux == true){
63         cout<<"Producto no encontrado X_X\n";
64     }
65     system("pause");
66 }
67 static void consultaId(){
68     bool aux = false;
69     system("cls");
70     string id;
71     cout<<"Ingrese el ID del producto a buscar: ";
72     cin>>id;
73
74     for(int i = 0; i < vec.size(); i++){
75         if(vec[i].getId() == id){
76             aux = true;
77             cout<<"-----Informacion del producto-----\n";
78             vec[i].mostrarDatos();
79             break;
80         }
81     }
82     if(aux == false){
83         cout<<"Producto no encontrado X_X\n";
84     }

```

```

85         system("pause");
86     }
87     static void modificarProducto(){
88         bool aux = true;
89         string id;
90         system("cls");
91         cout<<"Ingrese el ID del producto a modificar: ";
92         cin>>id;
93
94         for(int i = 0; i < vec.size(); i++){
95             if(vec[i].getId() == id){
96                 aux = false;
97                 int op;
98                 cout<<"Producto '"<<vec[i].getNombre()<<"' encontrado\n";
99                 cout<<"Seleccione el valor a modificar...\n";
100                 cout<<"[1] Nombre\n";
101                 cout<<"[2] ID\n";
102                 cout<<"[3] Categoria/s\n";
103                 cout<<"[4] Precio\n";
104                 cout<<"[5] Cantidad\n";
105                 cin>>op;
106
107                 switch(op){
108                     case 1:{
109
110                         string nuevoNom;
111                         cout<<"Ingrese el nuevo nombre: ";
112                         cin.ignore();
113                         getline(cin, nuevoNom);
114
115                         vec[i].setNombre(nuevoNom);
116                         cout<<"\nProducto modificado :)\n";
117                         break;
118                     }
119                     case 2:{
120
121                         string nuevoId;
122                         cout<<"Ingrese el nuevo ID: ";
123                         cin>>nuevoId;
124
125                         vec[i].setId(nuevoId);
126                         cout<<"\nProducto modificado :)\n";
127                         break;
128                     }
129                     case 3:{
130
131                         string nuevaCat;
132                         cout<<"Ingrese la nueva categoria: ";
133                         cin>>nuevaCat;
134
135                         vec[i].setCategoria(nuevaCat);
136                         cout<<"\nProducto modificado :)\n";

```

```

137         break;
138     }
139     case 4:{
140
141         int nuevoPre;
142         cout<<"Ingrese el nuevo precio: ";
143         cin>>nuevoPre;
144
145         vec[i].setPrecio(nuevoPre);
146         cout<<"\nProducto modificado :)\n";
147         break;
148     }
149     case 5:{
150
151         int nuevaCan;
152         cout<<"Ingrese el stock a sumar: ";
153         cin>>nuevaCan;
154
155         vec[i].setCantidad(vec[i].getCantidad() + nuevaCan);
156         cout<<"\nStock actualizado: "<<vec[i].getCantidad();
157         cout<<"\nProducto modificado :)\n";
158         break;
159     }
160     default:
161         cout<<"Opcion fuera de rango X_X\n";
162     }
163     break;
164 }
165 }
166 if(aux == true){
167     cout<<"Producto no encontrado X_X\n";
168 }
169 system("pause");
170 }
171 static void mostrar(){
172     system("cls");
173     cout<<"~~~~~ Objetos ~~~~~\n";
174     for(int i = 0; i < vec.size(); i++){
175         vec[i].mostrarDatos();
176     }
177     system("pause");
178 }
179 static void abrir(){
180     string archivo="Almacen.txt";
181     string linea;
182     ifstream entrada(archivo);
183
184     if(!entrada.is_open()) { //ta mal :(
185         system("cls");
186         cout<<"Error al abrir el archivo - '"<< archivo <<"' X_X\n";
187         cout<<"Inventario vacio\n";
188         system("pause");

```

```

189         }else{//ta bien :)
190             while (getline(entrada, linea)){
191                 vector<string> resultado;
192                 char delim = ' ';
193                 istream isstream(linea);
194                 string palabra;
195
196                 while(std::getline(isstream, palabra, delim)){
197                     resultado.push_back(palabra);
198                 }
199
200                 Producto ob;
201                 ob.setNombre(resultado[0]);
202                 ob.setCategoria(resultado[1]);
203                 ob.setId(resultado[2]);
204                 ob.setPrecio(atoi(resultado[3].c_str()));
205                 ob.setCantidad(atoi(resultado[4].c_str()));
206                 vec.push_back(ob);
207             }
208         }
209         entrada.close();
210     }
211     static void escribir(){
212         ofstream archivo;
213         archivo.open("Almacen.txt", ios::out);
214
215         if(archivo.fail()){
216             system("cls");
217             cout<<"Ha ocurrido un error al crear el archivo de inventario X_X\n";
218             system("pause");
219             exit(1);
220         }
221
222         for(int i = 0; i < vec.size(); i++){
223             archivo<<vec[i].getNombre()<<" "<<vec[i].getCategoria()<<" "
224             <<vec[i].getId()<<" "<<vec[i].getPrecio()<<" "<<vec[i].getCantidad();
225             archivo<<"\n";
226         }
227         archivo.close();
228     }
229     static Producto consultaCarrito(string id){
230         Producto prod;
231         for(int i = 0; i < vec.size(); i++){
232             if(vec[i].getId() == id){
233                 prod = vec[i];
234                 break;
235             }
236         }
237         return prod;//validar que no llegue nulo
238     }
239     static void consultaUsuario(string categ){
240         for(int i = 0; i < vec.size(); i++){

```

```

241         size_t found = vec[i].getCategoria().find(categ);
242         if (found != string::npos){
243             cout<<"["<<vec[i].getId()<<"] "<<vec[i].getNombre()<<" $"<<vec[i].getPrecio()<<"\n";
244         }
245     }
246     //system("pause") estatra en el main
247 }
248 };
249 //inicializa el inventario
250 vector<Producto> Inventario::vec = {};

```

Carrito (clase estática)

```

Inventario.cpp × Carrito.cpp × Producto.cpp ×
1  #include<iostream>
2  #include<vector>
3  #include<algorithm>
4  #include"Inventario.cpp"
5  using namespace std;
6
7  class Carrito{
8      private:
9          static string cliente;
10         static vector<Producto> lista;
11         static int precioTotal;
12     public:
13         static void setCliente(){
14             system("cls");
15             string nom;
16             cout<<"Ingrese su nombre de cliente: ";
17             cin>>nom;
18             cliente = nom;
19         }
20         static void agregarProducto(string id, int cantidad){//sin cls
21             Producto prod;
22             prod = Inventario::consultaCarrito(id);
23
24             if(prod.getNombre().empty()){
25                 cout<<"Producto no encontrado X_X\n";
26             }else{
27                 prod.setCantidad(cantidad);//Cantidad de producto en el CARRITO
28                 lista.push_back(prod);
29                 sumarPrecio(cantidad, prod.getPrecio());//Se suma el precio al total
30                 cout<<cantidad<<" pzs. de '"<<prod.getNombre()<<"' agregadas al carrito :)\n";
31                 cout<<"Total a pagar actual: "<<precioTotal<<"\n";
32             }
33
34             //system("pause"); estara en el main
35         }
36         static void sumarPrecio(int cant, int pre){
37             precioTotal += pre*cant;
38         }
39         static void quitarProducto(){//sin cls
40             bool aux = true;
41             string id;

```

```

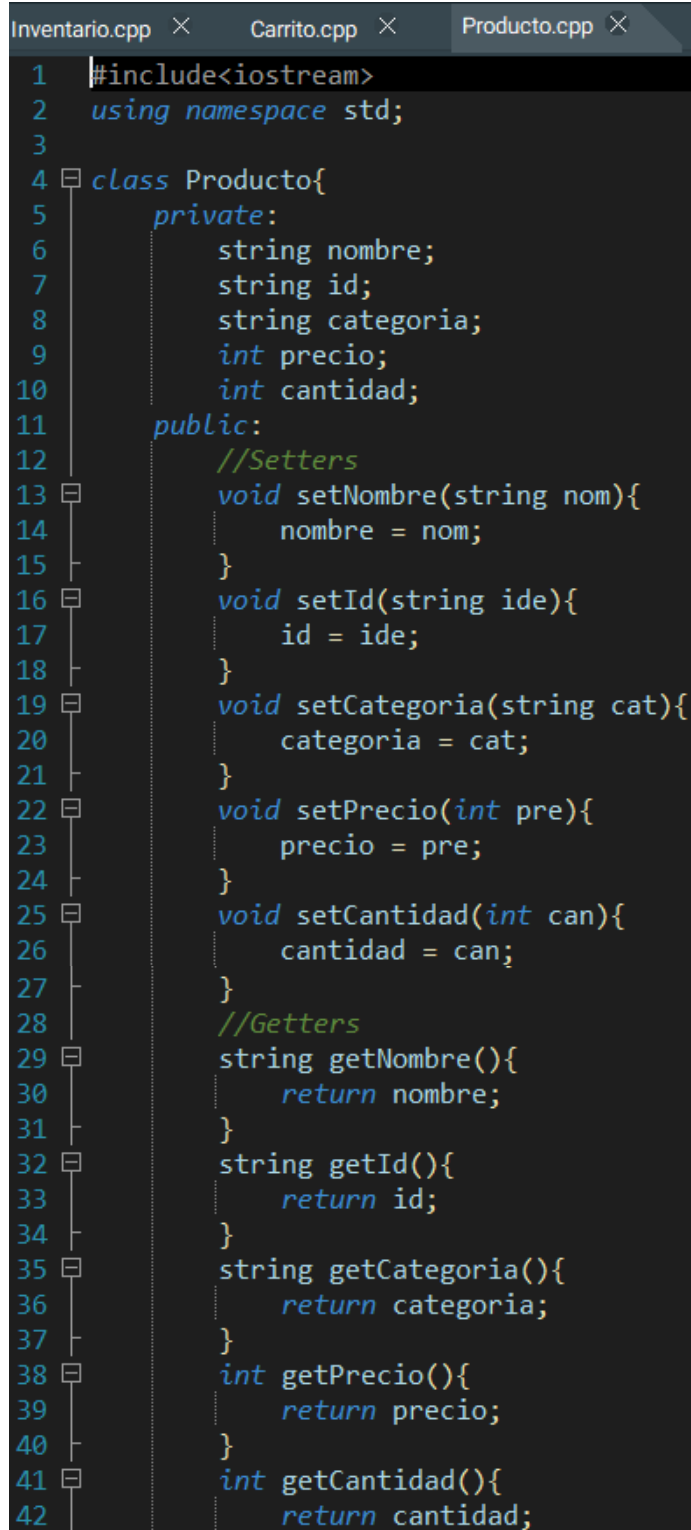
42     cout<<"Ingrese el ID del producto a eliminar: ";
43     cin>>id;
44
45     for(int i = 0; i < lista.size(); i++){
46         if(lista[i].getId() == id){
47             aux = false;
48             cout<<"Producto '"<<lista[i].getNombre()<<"' eliminado :)\n";
49             quitarPrecio(lista[i].getCantidad(), lista[i].getPrecio());//antes de eliminar se nece
50             lista.erase(lista.begin() + i);
51             cout<<"Total a pagar actual: "<<precioTotal<<"\n";
52             break;
53         }
54     }
55     if(aux == true){
56         cout<<"Producto no encontrado X_X\n";
57     }
58     system("pause");
59 }
60 static void quitarPrecio(int cant, int pre){
61     precioTotal -= pre*cant;
62 }
63 static void mostrarLista(){
64     system("cls");
65     cout<<"~~~~~ Carrito ~~~~~\n";
66     for(int i = 0; i < lista.size(); i++){
67         cout<<lista[i].getId()<<" " <<lista[i].getNombre()<<" x"<<lista[i].getCantidad()
68         <<" pzs.-----: $"<<lista[i].getPrecio()<<"c/u \n";
69     }
70     cout<<"Total a pagar: "<<precioTotal<<"\n";
71     //system("pause"); estara en el main
72 }
73 static void generarTicket(){
74     ofstream archivo;
75     archivo.open("Ticket.txt", ios::out);
76
77     if(archivo.fail()){
78         system("cls");
79         cout<<"Ha ocurrido un error al crear el archivo de ticket X_X\n";
80         system("pause");
81         exit(1);
82     }
83     archivo<<"Cliente: "<<cliente<<"\n";
84     archivo<<"~~~~~ Carrito ~~~~~\n";
85     for(int i = 0; i < lista.size(); i++){
86         archivo<<"Nombre-----: "<<lista[i].getNombre()<<"\n";
87         archivo<<"Precio-----: $"<<lista[i].getPrecio()<<"\n";
88         archivo<<"Cantidad----: "<<lista[i].getCantidad()<<" pzs.\n";
89         archivo<<"Total por producto: $"<<(lista[i].getPrecio()*lista[i].getCantidad())<<"\n\n";
90     }
91     archivo<<"Total a pagar: $"<<precioTotal;
92     archivo.close();
93 }
94 static void reiniciar(){
95     lista = {};
96     cliente = "";
97     precioTotal = 0;
98 }
99 };

```



```
100 //Se inicializan las variables de la clase
101 vector<Producto> Carrito::lista = {};
102 string Carrito::cliente = "";
103 int Carrito::precioTotal = 0;
```

Producto (clase/objeto)



```
1  #include<iostream>
2  using namespace std;
3
4  class Producto{
5      private:
6          string nombre;
7          string id;
8          string categoria;
9          int precio;
10         int cantidad;
11     public:
12         //Setters
13         void setNombre(string nom){
14             nombre = nom;
15         }
16         void setId(string ide){
17             id = ide;
18         }
19         void setCategoria(string cat){
20             categoria = cat;
21         }
22         void setPrecio(int pre){
23             precio = pre;
24         }
25         void setCantidad(int can){
26             cantidad = can;
27         }
28         //Getters
29         string getNombre(){
30             return nombre;
31         }
32         string getId(){
33             return id;
34         }
35         string getCategoria(){
36             return categoria;
37         }
38         int getPrecio(){
39             return precio;
40         }
41         int getCantidad(){
42             return cantidad;
```

```

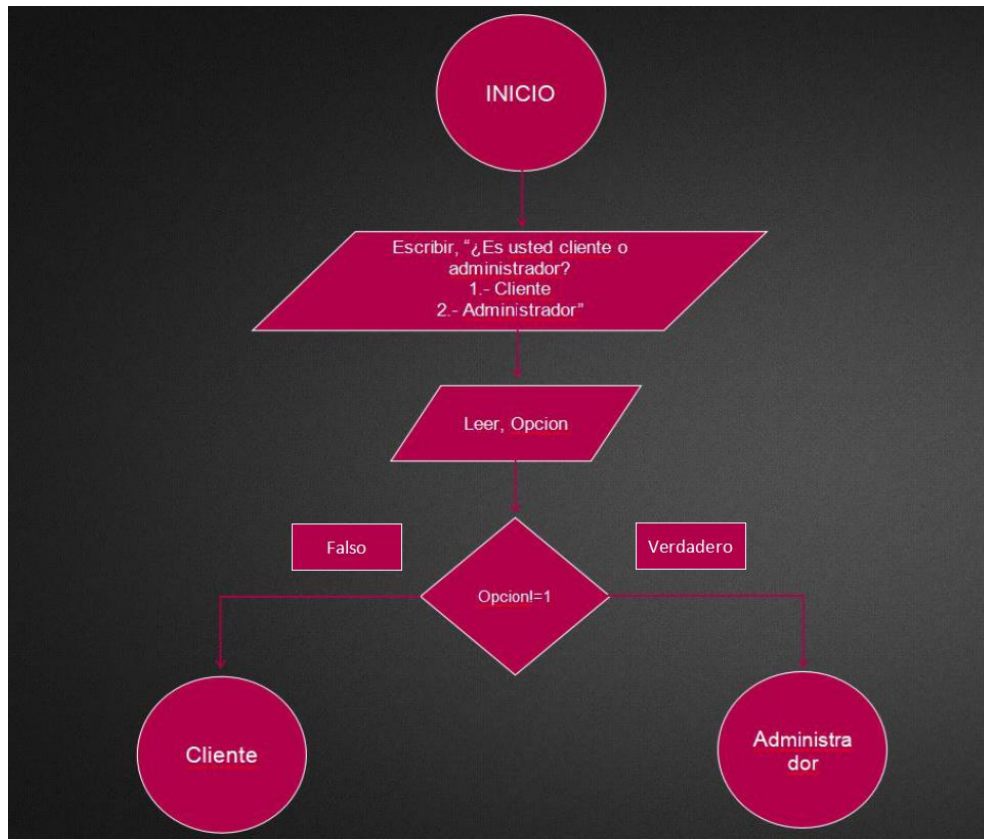
43 }
44 void mostrarDatos(){
45     cout<<"Nombre-----: "<<nombre<<"\n";
46     cout<<"Id-----: "<<id<<"\n";
47     cout<<"Categoria/s---: "<<categoria<<"\n";
48     cout<<"Precio-----: $"<<precio<<"\n";
49     cout<<"Cantidad-----: "<<cantidad<<"pzs. \n\n";
50 }
51 };

```



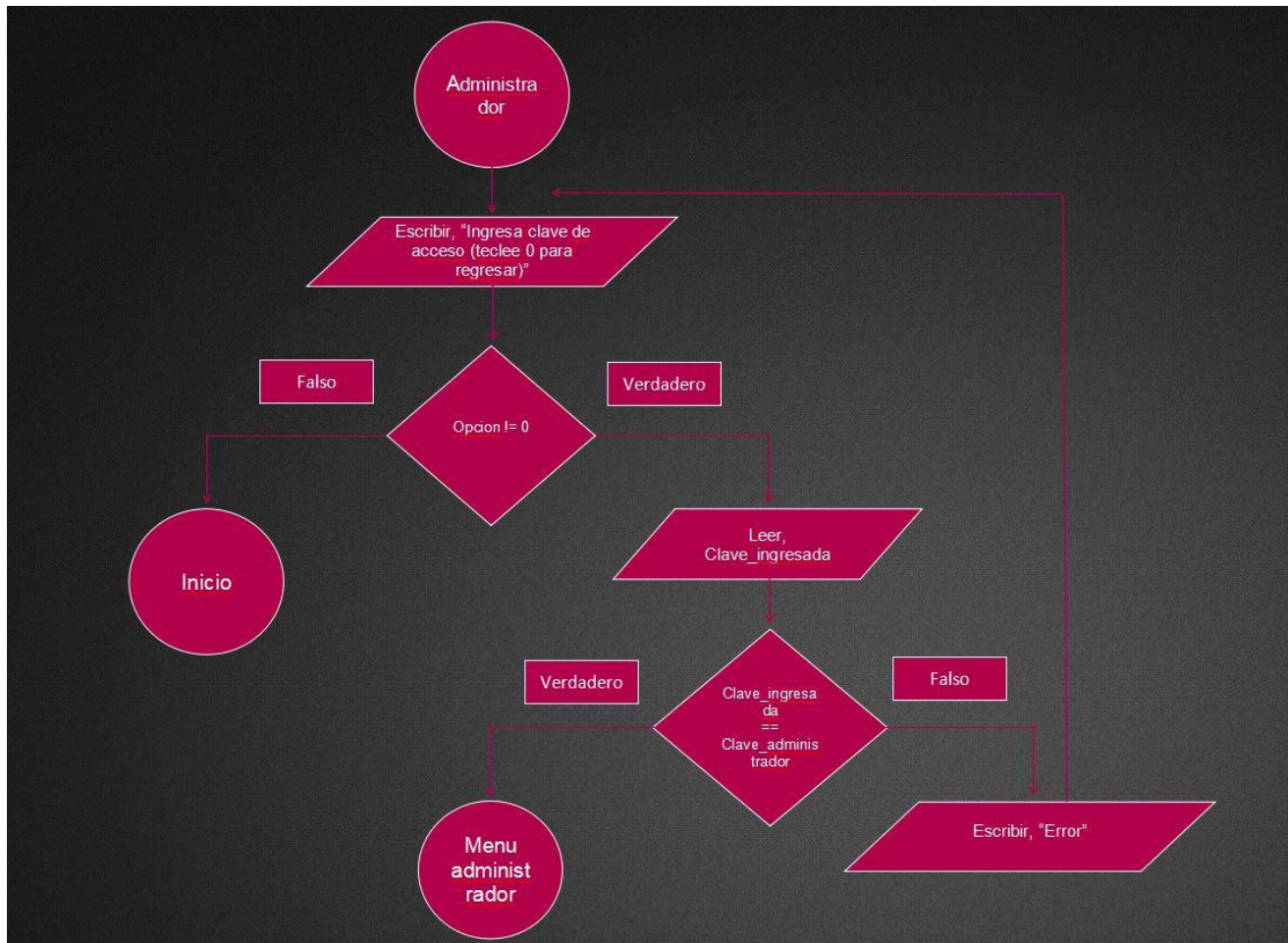
Diagrama general del programa

Menú principal del programa



Al ejecutar el programa esto es lo primero que verá el usuario. Teniendo la opción de ingresar como cliente o administrador según sea la ocasión.

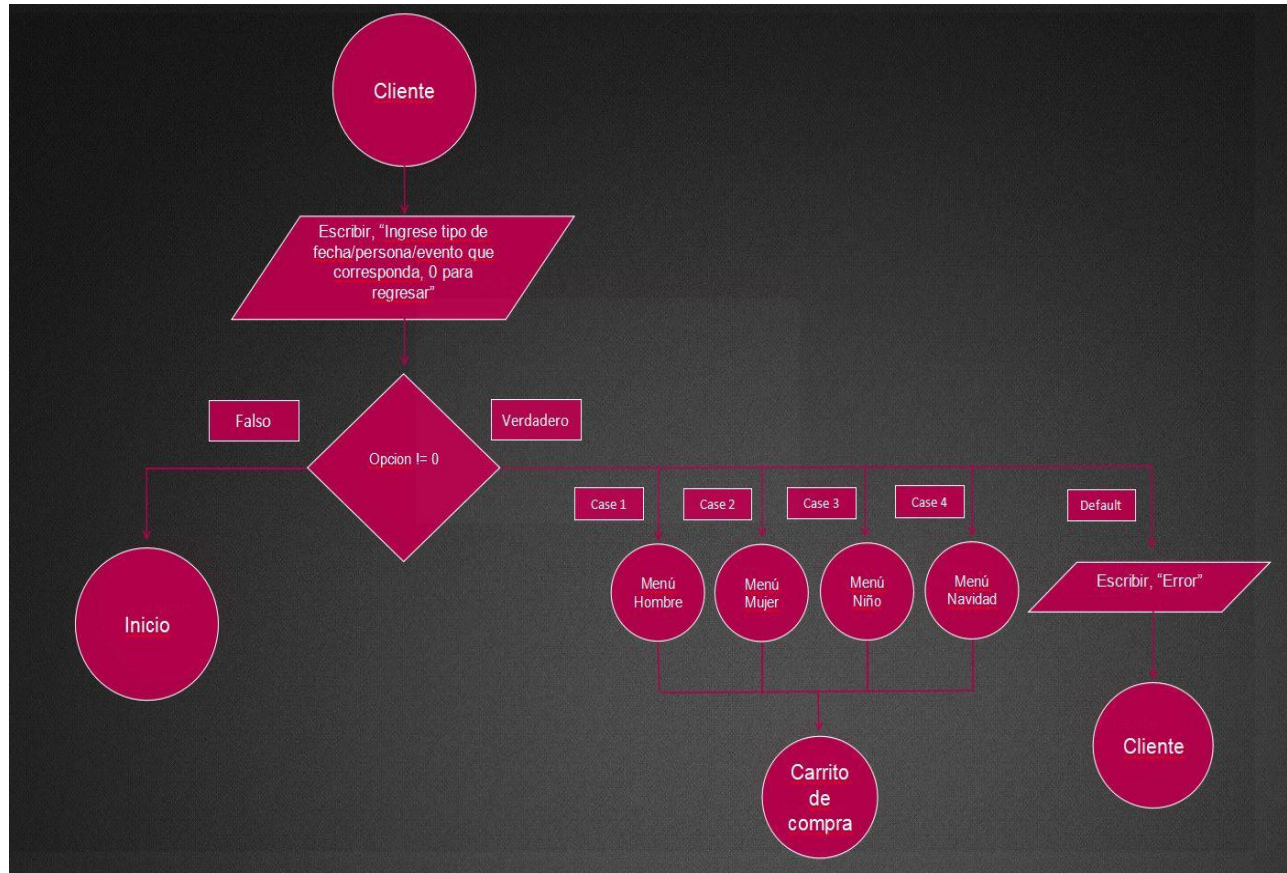
Administrador



Al ingresar por el lado del administrador deberemos ingresar la clave de acceso válida. Una vez sea ingresada la clave correcta, accederemos al menú del administrador:

- Dar de baja/alta productos
- Modificar productos
- Consultar inventario
- ...

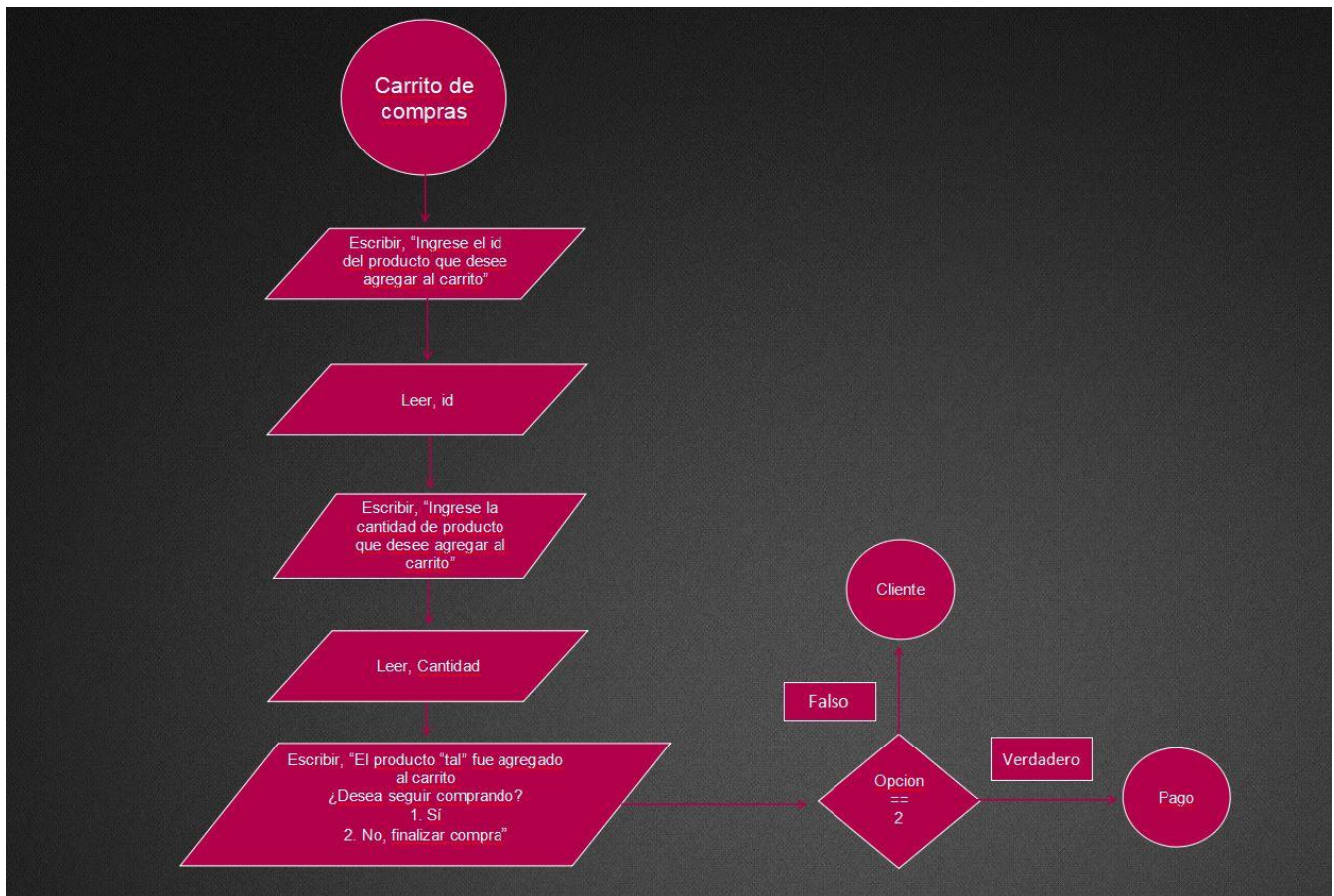
Cliente



Al ingresar por el lado del cliente podremos iniciar la búsqueda de los artículos que queremos de acuerdo con:

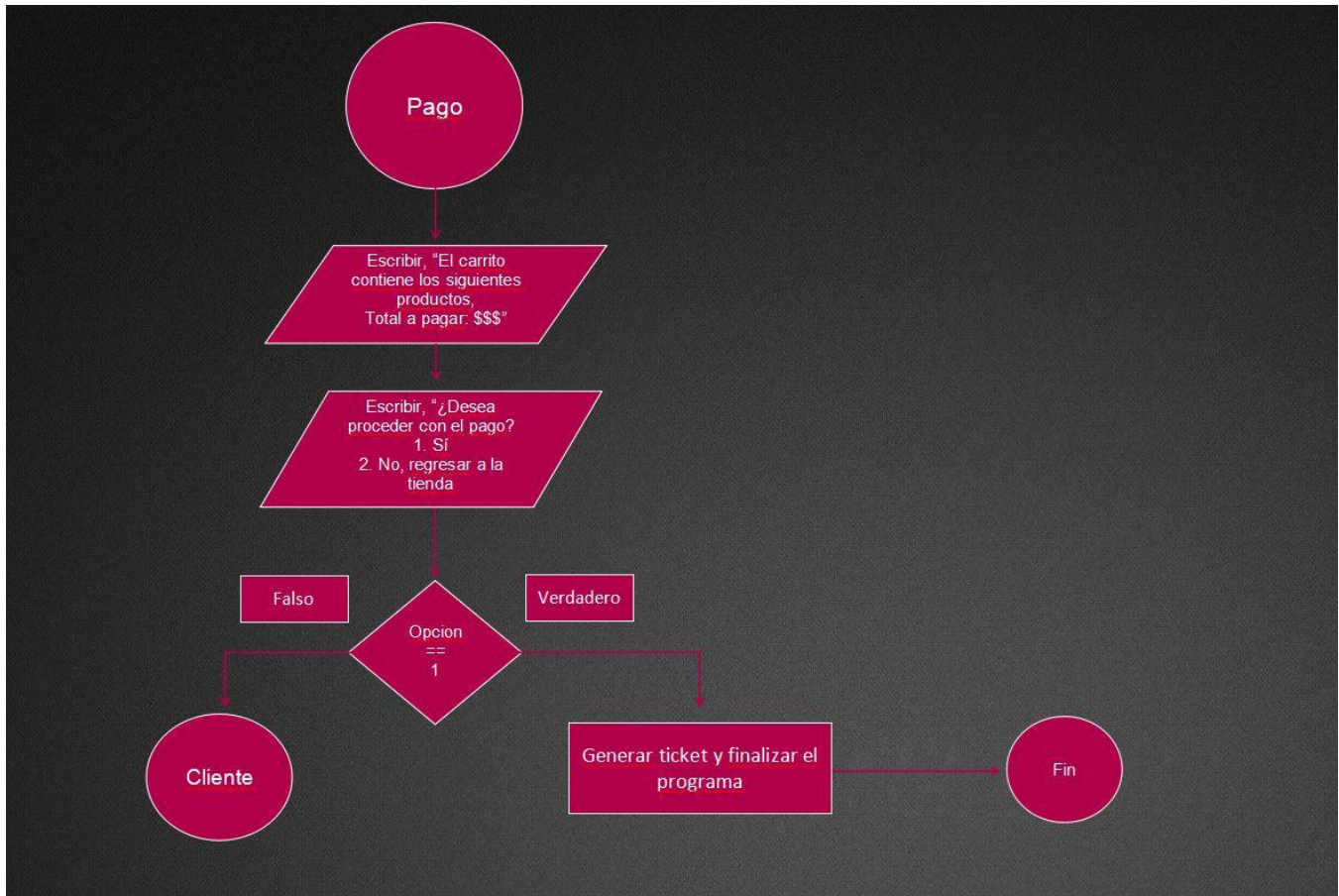
- Fecha
- Evento
- Hombre
- Mujer
- Niño
- ...

Carrito de compras



El cliente podrá añadir todos los productos que desee, mientras existan en el inventario, ingresando el id del producto y la cantidad de este.

Proceder con el pago



El cliente podrá visualizar a detalle los productos que añadió y el monto total, podrá regresar al menú o proceder a pagar y generar un ticket de compra.