

```

1  /*
2   * Pablo_Villa 874773
3   * 01/12/23
4   */
5
6  #include <iostream>
7  using namespace std;
8
9  const int MAX_PRODUCTOS = 100;
10 const int PRODUCTO_NO_ENCONTRADO = -1;
11 const int ALTA = 1;
12 const int BAJA = 2;
13 const int LISTADO = 3;
14 const int ENTRADA = 4;
15 const int SALIDA = 5;
16 const int BUSCAR = 6;
17 const int FIN = 7;
18
19 struct producto_almacen {
20     int codigo = 0;
21     string descripcion = "";
22     int existencias = 0;
23 };
24
25 void anadir_producto(producto_almacen productos[], int &numero_productos);
26 void leer_producto(producto_almacen &producto);
27 void listar_productos(producto_almacen productos[], int numero_productos);
28 void anadir(producto_almacen productos[], int &numero_productos);
29 void eliminar(producto_almacen productos[], int &numero_productos);
30 void borrar_producto(producto_almacen productos[], int &numero_productos);
31 void buscar(producto_almacen productos[], int &numero_productos);
32 int busqueda(const producto_almacen productos[], int codigo, int numero_productos);
33
34 int main(){
35     producto_almacen productos[MAX_PRODUCTOS];
36     int opcion = 0;
37     int numero_productos = 0;
38
39     while(opcion != FIN){
40         cout << ALTA << "->Alta " << BAJA << "->Baja " << LISTADO << "->Listado " << ENTRADA
41             << "->Entrada " << SALIDA << "->Salida " << BUSCAR << "-> Buscar " << FIN << "->Acabar: ";
42
43         cin >> opcion;
44
45         switch(opcion){
46             case ALTA:
47                 anadir_producto(productos, numero_productos);
48                 break;
49
50             case BAJA:
51                 borrar_producto(productos, numero_productos);
52                 break;
53
54             case LISTADO:
55                 listar_productos(productos, numero_productos);
56                 break;
57
58             case ENTRADA:
59                 anadir(productos, numero_productos);
60                 break;
61
62             case SALIDA:
63                 eliminar(productos, numero_productos);
64                 break;
65
66             case BUSCAR:

```

```

67         buscar(productos, numero_productos);
68         break;
69
70     case FIN:
71         cout << "Fin" << endl;
72         break;
73     }
74 }
75 }
76
77 void leer_producto(producto_almacen &producto){
78     cout << "Código : ";
79     cin >> producto.codigo;
80     cout << "Descripción : ";
81     cin.ignore();
82     getline(cin, producto.descripcion);
83     cout << "Existencias : ";
84     cin >> producto.existencias;
85 }
86
87 void anadir_producto(producto_almacen productos[], int &numero_productos){
88     if (numero_productos > MAX_PRODUCTOS){
89         cout << "Número máximo de productos" << endl;
90     } else {
91         leer_producto(productos[numero_productos]);
92         numero_productos++;
93     }
94 }
95
96 void borrar_producto(producto_almacen productos[], int &numero_productos){
97     int codigo = 0;
98     int posicion = 0;
99     cout << "Introduce un código de un producto que desees eliminar ";
100    cin >> codigo;
101
102
103    posicion = busqueda(productos, codigo, numero_productos);
104    if (posicion != PRODUCTO_NO_ENCONTRADO) {
105        productos[posicion] = productos[numero_productos - 1];
106        numero_productos--;
107        cout << "Producto eliminado" << endl;
108    }
109    else {
110        cout << "Producto no encontrado" << endl;
111    }
112 }
113
114 void mostrar_producto(const producto_almacen &producto){
115     cout << "Codigo: ";
116     cout << producto.codigo << endl;
117     cout << "Descripcion: ";
118     cout << producto.descripcion << endl;
119     cout << "Existencias: ";
120     cout << producto.existencias << endl;
121 }
122
123 void listar_productos(producto_almacen productos[], int numero_productos){
124     for (int i = 0; i < numero_productos; i++){
125         mostrar_producto(productos[i]);
126     }
127 }
128
129 void anadir(producto_almacen productos[], int &numero_productos){
130     int codigo = 0;
131     int existencias = 0;
132     bool encontrado = false;

```

```

133     int posicion = 0;
134
135     cout << "codigo: ";
136     cin >> codigo;
137     cout << "Añadir existencias: ";
138     cin >> existencias;
139
140     posicion = busqueda(productos, codigo, numero_productos);
141     if (posicion == PRODUCTO_NO_ENCONTRADO) {
142         cout << "El código introducido no corresponde con ningún producto." << endl;
143     } else {
144         productos[posicion].existencias = productos[posicion].existencias + existencias;
145         cout << "Se han almacenado (" << existencias << ") de "
146             << productos[posicion].descripcion << " con código " << productos[posicion].codigo << endl;
147     }
148 }
149
150 void eliminar(producto_almacen productos[], int &numero_productos){
151     int codigo = 0;
152     int existencias = 0;
153     bool encontrado = false;
154     int posicion = 0;
155
156     cout << "Codigo: ";
157     cin >> codigo;
158     cout << "Eliminar existencias: ";
159     cin >> existencias;
160
161     posicion = busqueda(productos, codigo, numero_productos);
162     if (posicion == PRODUCTO_NO_ENCONTRADO) {
163         cout << "El código introducido no corresponde con ningún producto." << endl;
164     } else {
165         productos[posicion].existencias -= existencias;
166         cout << "Se han retirado (" << existencias << ") de "
167             << productos[posicion].descripcion << " con código " << productos[posicion].codigo << endl;
168     }
169 }
170
171 void buscar(producto_almacen productos[], int &numero_productos){
172     int codigo = 0;
173     int posicion = 0;
174     cout << "Introduce un código que desees buscar ";
175     cin >> codigo;
176
177     posicion = busqueda(productos, codigo, numero_productos);
178     if(posicion == PRODUCTO_NO_ENCONTRADO){
179         cout << "El código introducido no corresponde a ningún producto" << endl;
180     } else {
181         mostrar_producto(productos[posicion]);
182     }
183 }
184
185 int busqueda(const producto_almacen productos[], int codigo, int numero_productos){
186     for (int i = 0; i < numero_productos; i++){
187         if(codigo == productos[i].codigo){
188             return i;
189         }
190     }
191     return PRODUCTO_NO_ENCONTRADO;
192 }

```