

# Prueba de Caja Negra

---

*“Sistema automatizado de control de Inventario”*

**Versión 5.0**

**Integrantes:**

Caizapanta Muela Tammy Amarilis  
Guaiguacundo Aguirre Valeria Naomi  
Pincha Llanos Estefany Anahí  
Robalino Zaldumbide Alejandro Benjamín

**Fecha**

2025-01-21

# ÍNDICE

<b>ÍNDICE</b>	<b>2</b>
<b>1. REQ 001 - Agregar Productos</b>	<b>4</b>
1.1. Historia de usuario	4
1.2. Partición de clases equivalentes	4
1.3. Código	5
1.3.1. Lenguaje C en Code::Blocks	5
1.4. Ejecución	7
1.5. Pass (OK)	7
1.6. No Pass (No OK)	7
<b>2. REQ 002 - Buscar Productos</b>	<b>8</b>
2.1. Historia de usuario	8
2.2. Partición de clases equivalentes	8
2.3. Código	8
2.3.1. Lenguaje C en Code::Blocks	8
2.4. Ejecución	8
2.4.1. Pass (OK)	8
2.4.2. No Pass (No OK)	8
<b>3. REQ 003 - Entrada de productos</b>	<b>8</b>
3.1. Historia de usuario	8
3.2. Partición de clases equivalentes	9
3.3. Código	9
3.3.1. Lenguaje C en Code::Blocks	9
3.4. Ejecución	9
3.4.1. Pass (OK)	9
3.4.2. No Pass (No OK)	9
<b>4. REQ 004 - Salida de productos</b>	<b>9</b>
4.1. Historia de usuario	9
4.2. Partición de clases equivalentes	10
4.3. Código	10
4.3.1. Lenguaje C en Code::Blocks	10
4.4. Ejecución	10
4.4.1. Pass (OK)	10
4.4.2. No Pass (No OK)	10
<b>5. REQ 005</b>	<b>10</b>
5.1. Historia de usuario	10
5.2. Partición de clases equivalentes	11
5.3. Código	11
5.3.1. Lenguaje C en Code::Blocks	11
5.4. Ejecución	12
5.4.1. Pass (OK)	12
5.4.2. No Pass (No OK)	12
<b>6. REQ 006</b>	<b>12</b>
6.1. Historia de usuario	12
6.2. Partición de clases equivalentes	13
6.3. Código	13

6.3.1. Lenguaje C en Code::Blocks	13
6.4. Ejecución	13
6.4.1. Pass (OK)	13
6.4.2. No Pass (No OK)	13

## 1. REQ 001 - Agregar Productos

### 1.1. Historia de usuario

Historia de usuario	
Número: REQ 001	Usuario: Administrador
Nombre de la Historia: Agregar productos	
Prioridad: Alta	
Programador responsable: Tammy Caizapanta	
Descripción: Se selecciona la opción para digitar el código único de cada producto y su información: nombre, proveedor, cantidad, precio.	
Validación: Se registra el producto agregado con un mensaje de confirmación de "Producto agregado exitosamente".	

### 1.2. Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
nuevo_producto[50]	EC1:nuevo_producto[50]==nuevo_producto[50]	VÁLIDO	"leche entera vita"
	EC2:nuevo_producto[50]!=nuevoproducto[50]	NO VÁLIDO	"@sk12339"
confirmar	EC3:confirmar==confirmar	VÁLIDO	"V"
	EC4:confirmar==confirmar	VÁLIDO	"F"
	EC5:confirmar!=confirmar	NO VÁLIDO	"HOLA"

### 1.3. Código

#### 1.3.1. Lenguaje C en Code::Blocks

```
38 // Función para agregar productos
39 void agregar_producto() {
40     if (prod < 100) {
41         char nuevo_producto[50];
42         char confirmar;
43
44         printf("Ingrese el nombre del nuevo producto: ");
45         getchar();
46         fgets(nuevo_producto, 50, stdin);
47
48         nuevo_producto[strcspn(nuevo_producto, "\n")] = '\0';
49
50
51         for (int i = 0; nuevo_producto[i] != '\0'; i++) {
52             if (isdigit(nuevo_producto[i])) {
53                 printf("Error: El nombre del producto no puede contener numeros.\n");
54                 return;
55             }
56         }
57
58         printf("Esta seguro de que desea agregar el producto '%s' (V/F): ", nuevo_producto);
59         scanf(" %c", &confirmar);
60         confirmar = toupper(confirmar);
61
62         if (confirmar == 'V') {
63             sprintf(producto[prod].product, "%03d.%s", prod + 1, nuevo_producto);
64             prod++;
65             printf("Producto agregado exitosamente.\n");
66         } else if (confirmar == 'F') {
67             printf("No se agrega el producto.\n");
68         } else {
69             printf("Accion no valida. \n");
70         }
71     } else {
72         printf("No se pueden agregar mas productos. Capacidad maxima alcanzada.\n");
73     }
74 }
```

### 1.4. Ejecución

#### 1.5. Pass (OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 2
Ingrese el nombre del nuevo producto: Pimiento
Esta seguro de que desea agregar el producto 'Pimiento' (V/F): V
Producto agregado exitosamente.
```

#### 1.6. No Pass (No OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 2
Ingrese el nombre del nuevo producto: 1234
Error: El nombre del producto no puede contener numeros.
```

## 2. REQ 002 - Buscar Productos

### 2.1. Historia de usuario

Historia de usuario	
Número: REQ 002	Usuario: Administrador
Nombre de la Historia: Búsqueda de productos	
Prioridad: Alta	
Programador responsable: Valeria Guaiguacundo	
Descripción: El sistema muestra la información de los productos, su código, nombre y existencias.	
Validación: La búsqueda de productos por códigos es exitosa y muestra los resultados satisfactoriamente.	

### 2.2. Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
termino[50]	EC1:termino[50]==termino[50]	VÁLIDO	"leche entera"
	EC2:termino[50]!=termino[50]	NO VÁLIDO	"@9%&\$%{"

### 2.3. Código

#### 2.3.1. Lenguaje C en Code::Blocks

```
// Función para buscar productos
void buscar_producto() {
    char termino[50];
    int encontrado = 0;

    printf("Ingrese el nombre o fragmento del producto a buscar: ");
    getchar();
    fgets(termino, 50, stdin);

    termino[strcspn(termino, "\n")] = '\0';

    printf("\nResultados de búsqueda:\n");
    for (int i = 0; i < prod; i++) {
        if (strstr(producto[i].product, termino) != NULL) {
            printf(" %d: [Producto: %s] (Cantidad: %d)\n", i + 1, producto[i].product, producto[i].cantidad);
            encontrado = 1;
        }
    }

    if (!encontrado) {
        printf("No se encontraron productos que coincidan con '%s'.\n", termino);
    }
}
```

## 2.4. Ejecución

### 2.4.1. Pass (OK)

```
8. Salir
Ingrese su opcion: cebolla
Ingrese el nombre o fragmento del producto a buscar:
Resultados de busqueda:
2: [Producto: 002.Cebolla] (Cantidad: 30)

***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
```

### 2.4.2. No Pass (No OK)

```
Ingrese su opcion: 3
Ingrese el nombre o fragmento del producto a buscar: 239id

Resultados de busqueda:
No se encontraron productos que coincidan con '239id'.

***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
```

## 3. REQ 003 - Entrada de productos

### 3.1. Historia de usuario

Historia de usuario	
Número: REQ 003	Usuario: Administrador
Nombre de la Historia: Entrada de productos	
Prioridad: Alta	
Programador responsable: Estefany Pincha	
Descripción: Botón que ingresara la cantidad de un producto que se encuentra agregado inicialmente en la lista de productos disponible, caso contrario indicará el programa que el producto no se encontró.	
Validación: El ingreso del nombre del producto, cantidad y confirmación de agregación de producto	

### 3.2. Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
termino[50]	EC1:termino[50]==termino[50]	VÁLIDO	"pimientos rojos"
	EC2:termino[50]!=termino[50]	NO VÁLIDO	"\$%/"/)=!=?!"/"
cantidad	EC3:cantidad==cantidad	VÁLIDO	7
	EC4:cantidad!=cantidad	NO VÁLIDO	-6
encontrado	EC5:encontrado==encontrado	VÁLIDO	0
	EC6:encontrado!=encontrado	NO VÁLIDO	120
confirmar	EC7:confirmar==confirmar	VÁLIDO	"V"
	EC8:confirmar==confirmar	VÁLIDO	"F"
	EC9:confirmar!=confirmar	NO VÁLIDO	"A"

### 3.3. Código

#### 3.3.1. Lenguaje C en Code::Blocks

```

101 // Función para entrada de productos
102 void entrada_producto() {
103     char termino[50];
104     int cantidad;
105     int encontrado = 0;
106     char confirmar;
107
108     printf("Ingrese el código o nombre del producto para aumentar su cantidad: ");
109     getchar();
110     fgets(termino, 50, stdin);
111     termino[strcspn(termino, "\n")] = '\0';
112
113     if (strlen(termino) == 0) {
114         printf("Error: No ingreso ningun valor. Intente nuevamente.\n");
115         return;
116     }
117
118     for (int i = 0; i < prod; i++) {
119         if (strcmp(producto[i].product, termino) != NULL) {
120             printf("\nProducto encontrado: %s (Cantidad actual: %d)\n", producto[i].product, producto[i].cantidad);
121             printf("Ingrese la cantidad a agregar: ");
122             if (scanf("%d", &cantidad) != 1 || cantidad <= 0) {
123                 printf("Error: Cantidad invalida. Solo se aceptan numeros positivos.\n");
124                 while (getchar() != '\n');
125                 return;
126             }
127
128             printf("Esta seguro de que desea agregar %d a '%s' (V/F): ", cantidad, producto[i].product);
129             scanf(" %c", &confirmar);
130             confirmar = toupper(confirmar);
131
132             if (confirmar == 'V') {
133                 producto[i].cantidad += cantidad;
134                 printf("\nCantidad actualizada. Nueva cantidad: %d\n", producto[i].cantidad);
135             } else if (confirmar == 'F') {
136                 printf("\nNo se realizo ningun cambio.\n");
137             } else {
138                 printf("\nOpcion no valida.\n");
139             }
140
141             encontrado = 1;
142             break;
143         }
144     }
145     if (!encontrado) {
146         printf("\nNo se encontro el producto con el código o nombre '%s'.\n", termino);
147     }
148 }
149

```



### 3.4. Ejecución

#### 3.4.1. Pass (OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: 001
Producto encontrado: 001.Pimiento (Cantidad actual: 50)
Ingrese la cantidad a agregar: 23
Esta seguro de que desea agregar 23 a '001.Pimiento' (V/F): V
Cantidad actualizada. Nueva cantidad: 73
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: Pimiento
Producto encontrado: 001.Pimiento (Cantidad actual: 73)
Ingrese la cantidad a agregar: 45
Esta seguro de que desea agregar 45 a '001.Pimiento' (V/F): V
Cantidad actualizada. Nueva cantidad: 118
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: 001
Producto encontrado: 001.Pimiento (Cantidad actual: 118)
Ingrese la cantidad a agregar: 7
Esta seguro de que desea agregar 7 a '001.Pimiento' (V/F): V
Cantidad actualizada. Nueva cantidad: 125
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: 005
Producto encontrado: 005.Fideos Don Vitorio (Cantidad actual: 100)
Ingrese la cantidad a agregar: 23
Esta seguro de que desea agregar 23 a '005.Fideos Don Vitorio' (V/F): F
No se realizo ningun cambio.
```

### 3.4.2. No Pass (No OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: 002
Producto encontrado: 002.Cebolla (Cantidad actual: 30)
Ingrese la cantidad a agregar: -4
Error: Cantidad invalida. Solo se aceptan numeros positivos.
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: ***
No se encontro el producto con el codigo o nombre '***'.
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: Pimientos
No se encontro el producto con el codigo o nombre 'Pimientos'.
```

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 4
Ingrese el codigo o nombre del producto para aumentar su cantidad: 3
Producto encontrado: 003.Ajo (Cantidad actual: 20)
Ingrese la cantidad a agregar: 7
Esta seguro de que desea agregar 7 a '003.Ajo' (V/F): A
Opcion no valida..
```

## 4. REQ 004 - Salida de productos

### 4.1. Historia de usuario

Historia de usuario	
Número: REQ 001	Usuario: Administrador
Nombre de la Historia: Salida de productos	
Prioridad: Alta	

<b>Programador responsable:</b> Benjamin Robalino
<b>Descripción:</b> Vincular la lista ID de los productos disponibles. Crear Funcion de salida de productos. Actualizar el listado de disponibilidad.
<b>Validación:</b> Al registrar la salida de un producto este debe verse reflejado en el stock. Se tiene como límite la cantidad existente. No se puede hacer la salida de mayor cantidad.

#### 4.2. Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
termino[50]	EC1:termino[50]==termino[50]	VÁLIDO	"001.Pimiento"
	EC2:termino[50]!=termino[50]	NO VÁLIDO	"#\$\$&"%!/&"()#"
cantidad	EC3:cantidad==cantidad	VÁLIDO	5 (asumiendo stock $\geq$ 5)
	EC4:cantidad!=cantidad	NO VÁLIDO	5 (asumiendo stock $<$ 5)
encontrado	EC5:encontrado==encontrado	VÁLIDO	0
	EC6:encontrado!=encontrado	NO VÁLIDO	5
confirmar	EC7:confirmar==confirmar	VÁLIDO	"V"
	EC8:confirmar==confirmar	NO VÁLIDO	"F"
	EC9:confirmar!=confirmar	NO VÁLIDO	"A"

## 4.3. Código

### 4.3.1. Lenguaje C en Code::Blocks

```
void salida_producto() {
    char termino[50];
    int cantidad;
    int encontrado = 0;
    char confirmar;

    printf("Ingrese el código o nombre del producto para disminuir su cantidad: ");
    getchar();
    fgets(termino, 50, stdin);
    termino[strcspn(termino, "\n")] = '\0';

    for (int i = 0; i < prod; i++) {
        if (strstr(producto[i].product, termino) != NULL) {
            printf("\nProducto encontrado: %s (Cantidad actual: %d)\n", producto[i].product, producto[i].cantidad);
            printf("Ingrese la cantidad a retirar: ");
            if (scanf("%d", &cantidad) != 1 || cantidad <= 0) {
                printf("Error: Cantidad invalida.\n");
                while (getchar() != '\n');
                return;
            }

            printf("Esta seguro de que desea retirar %d de '%s' (V/F): ", cantidad, producto[i].product);
            scanf(" %c", &confirmar);
            confirmar = toupper(confirmar);

            if (confirmar == 'V') {
                if (cantidad <= producto[i].cantidad) {
                    producto[i].cantidad -= cantidad;
                    printf("\nCantidad actualizada. Nueva cantidad: %d\n", producto[i].cantidad);
                } else {
                    printf("\nCantidad insuficiente en inventario.\n");
                }
            } else if (confirmar == 'F') {
                printf("\nNo se realizo ningun cambio.\n");
            } else {
                printf("\nOpcion no valida.\n");
            }

            encontrado = 1;
            break;
        }
    }

    if (!encontrado) {
        printf("\nNo se encontro el producto con el código o nombre '%s'.\n", termino);
    }
}
```

## 4.4. Ejecución

### 4.4.1. Pass (OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 5
Ingrese el código o nombre del producto para disminuir su cantidad: 001

Producto encontrado: 001.Pimiento (Cantidad actual: 50)
Ingrese la cantidad a retirar: 4
Esta seguro de que desea retirar 4 de '001.Pimiento' (V/F): v

Cantidad actualizada. Nueva cantidad: 46

***** Seleccione una opcion: *****
```

#### 4.4.2. No Pass (No OK)

```
Ingrese su opcion: 5
Ingrese el codigo o nombre del producto para disminuir su cantidad: 999

No se encontro el producto con el codigo o nombre '999'.
```

```
Ingrese la cantidad a retirar: 50
Esta seguro de que desea retirar 50 de '001.Pimiento' (V/F): x

Opcion no valida.
```

```
Producto encontrado: 001.Pimiento (Cantidad actual: 46)
Ingrese la cantidad a retirar: 50
Esta seguro de que desea retirar 50 de '001.Pimiento' (V/F): V

Cantidad insuficiente en inventario.
```

## 5. REQ 005 - Eliminar productos

### 5.1. Historia de usuario

Historia de usuario	
Número: REQ 005	Usuario: Administrador
Nombre de la Historia: Eliminar productos	
Prioridad: Alta	
Programador responsable: Tammy Caizapanta	
Descripción: Se selecciona la opción de eliminar producto y se digita el código único de cada producto para su posterior eliminación con doble confirmación.	
Validación: Se registra la eliminación del producto con una doble confirmación de "¿Está seguro que desea eliminar este producto?" y si es afirmativo aparece el mensaje "Producto eliminado exitosamente".	

### 5.2. Partición de clases equivalentes

### 5.3.

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
termino[50]	EC1:termino[50]==termino[50]	VÁLIDO	"001.Pimiento"
	EC2:termino[50]!=termino[50]	NO VÁLIDO	"/(!&())&!)=&#(=!"
encontrado	EC5:encontrado==encontrado	VÁLIDO	0
	EC6:encontrado!=encontrado	NO VÁLIDO	5
confirmar	EC7:confirmar==confirmar	VÁLIDO	"V"
	EC8:confirmar==confirmar	NO VÁLIDO	"F"
	EC9:confirmar!=confirmar	NO VÁLIDO	"A"

### 5.4. Código

#### 5.4.1. Lenguaje C en Code::Blocks

```

199 // Función para eliminar productos
200 void eliminar_producto() {
201     char termino[50];
202     int encontrado = -1;
203     char confirmar;
204
205     printf("Ingrese el nombre o código del producto a eliminar: ");
206     getchar();
207     fgets(termino, 50, stdin);
208     termino[strcspn(termino, "\n")] = 0;
209
210     for (int i = 0; i < prod; i++) {
211         if (strstr(producto[i].product, termino) != NULL) {
212             encontrado = i;
213             break;
214         }
215     }
216
217     if (encontrado != -1) {
218         // Encuentra el índice de eliminar el producto
219         printf("Producto encontrado: %s. Esta seguro de que desea eliminarlo (V/F): ", producto[encontrado].product);
220         scanf(" %c", &confirmar);
221         confirmar = toupper(confirmar);
222
223         if (confirmar == 'V') {
224             printf("Producto eliminado: %s\n", producto[encontrado].product);
225             for (int i = encontrado; i < prod - 1; i++) {
226                 strcpy(producto[i].product, producto[i + 1].product);
227                 producto[i].cantidad = producto[i + 1].cantidad;
228             }
229             prod--;
230         } else if (confirmar == 'F') {
231             printf("Acción cancelada. No se eliminó ningún producto.\n");
232         } else {
233             printf("Acción no válida.\n");
234         }
235     } else {
236         printf("No se encontró un producto que coincida con '%s'.\n", termino);
237     }
238 }

```

## 5.5. Ejecución

### 5.5.1. Pass (OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 6
Ingrese el nombre o codigo del producto a eliminar: 001
Producto encontrado: 001.Pimiento. Esta seguro de que desea eliminarlo (V/F): V
Producto eliminado: 001.Pimiento
```

### 5.5.2. No Pass (No OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Busqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 6
Ingrese el nombre o codigo del producto a eliminar: ***
No se encontr% un producto que coincida con '***'.
```

## 6. REQ 006 - Generar reporte de la disponibilidad de productos

### 6.1. Historia de usuario

Historia de usuario	
Número: REQ 006	Usuario: Administrador
Nombre de la Historia: Generar reporte de la disponibilidad de productos	
Prioridad: Media	
Programador responsable: Estefany Pincha	

**Descripción** La opción del menú permitirá descargar la lista de productos disponibles en un archivo dentro del escritorio del computador

**Validación:** Guardar un documento en el escritorio del computador con un nombre en específico

## 6.2. Partición de clases equivalentes

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
Confirmar	EC1:Confirmar==Confirmar	VÁLIDO	V
	EC1:Confirmar==Confirmar	VÁLIDO	F
	EC2:Confirmar!=Confirmar	INVÁLIDO	A

## 6.3. Código

### 6.3.1. Lenguaje C en Code::Blocks

```

239 //Funcion para guardar el archivo
240
241 void guardar() {
242     // Abre el archivo en modo escritura
243     FILE *archivo = fopen("productos.txt", "w");
244     if (archivo == NULL) {
245         printf("Error al abrir el archivo.\n");
246         return;
247     }
248     for (int i = 0; i < prod; i++) {
249         fprintf(archivo, "%d: [Producto: %s]\n", i + 1, producto[i].product);
250     }
251     fclose(archivo);
252     printf("Lista de productos guardada exitosamente en 'productos.txt'.\n");
253 }
254
255
256

```

## 6.4. Ejecución

### 6.4.1. Pass (OK)

```

2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 7
Lista de productos guardada exitosamente en 'productos.txt'.

Desea volver al menu principal (V/F): V

**** Seleccione una opcion: ****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 7
Lista de productos guardada exitosamente en 'productos.txt'.

Desea volver al menu principal (V/F): F
Saliendo del programa. ¡Hasta luego!

-----
Process exited after 543.3 seconds with return value 0
Presione una tecla para continuar . . .

```



#### 6.4.2. No Pass (No OK)

```
***** Seleccione una opcion: *****
1. Visualizar la lista de productos disponibles
2. Agregar productos
3. Búsqueda de productos
4. Entrada de productos
5. Salida de productos
6. Eliminar productos
7. Generar reporte
8. Salir
Ingrese su opcion: 7
Lista de productos guardada exitosamente en 'productos.txt'.

Desea volver al menu principal (V/F): A
Opcion no valida. Intente nuevamente.
```