

Prueba de Caja Negra

**“Implementación de un sistema de inventario para el local
ChaskiBots”**

Integrantes:

Cobeña Bravo Davis Ariel

German Luna Jhon Alexis

Ramos Vilca Paulo Alejandro

Grupo N° 4

Fecha: 2025 – 02 – 12

Índice

Contenido

Historial de versionamiento.....	3
1. REQ 001 Inicio de sesión	4
1.1. Historia de usuario.....	4
1.2. Partición de clases equivalentes.....	4
1.3. Código.....	4
1.4. Ejecución.....	5
<input type="checkbox"/> Pass (OK)	5
<input type="checkbox"/> Fail (no OK)	5
2. REQ002 Añadir un producto	5
2.1 Historia de usuario	5
2.2 Partición de clases equivalentes.....	6
2.3 Código.....	6
2.4 Ejecución.....	6
<input type="checkbox"/> Pass (OK)	7
<input type="checkbox"/> Fail (no OK)	7
3. REQ003 Modificar los productos.....	8
3.1 Historia de usuario.....	8
3.2 Partición de clases equivalentes.....	9
3.3 Codigo.....	9
3.4 Ejecucion.....	10
<input type="checkbox"/> Pass (OK)	10
<input type="checkbox"/> Fail (no OK)	10
4. REQ 004 Eliminar productos	12
4.1 Historia de usuario.....	12
4.2 Partición de clases equivalentes.....	12
4.3 Codigo.....	12
4.4 Ejecucion.....	13
<input type="checkbox"/> Pass (OK)	13
<input type="checkbox"/> Fail (no OK)	13
5. REQ 005 Búsqueda de productos	14
5.1 Historia de usuario.....	14
5.2 Partición de clases equivalentes.....	14
5.3 Codigo.....	14
5.4 Ejecución.....	15
<input type="checkbox"/> Pass (OK)	15
<input type="checkbox"/> Fail (no OK)	15

Historial de versionamiento

Fecha	Versión	Descripción	Autores
09 – 01 – 2025	1	Versión inicial REQ001.	Davis Cobeña Jhon German Paulo Ramos
16 – 01 – 2025	2	Partición de clases equivalentes hasta REQ004.	Davis Cobeña Jhon German Paulo Ramos
22 – 01 – 2025	3	Partición de clases equivalentes hasta REQ005.	Davis Cobeña Jhon German Paulo Ramos
11 – 02 – 2025	4	Corrección de partición de clases equivalentes	Davis Cobeña Jhon German Paulo Ramos
12 – 02 – 2025	5	Última versión.	Davis Cobeña Jhon German Paulo Ramos

1. REQ 001 Inicio de sesión

1.1. Historia de usuario

Historia de usuario	
REQ001	Nombre de historia: Inicio de sesión
Prioridad: Alta	Programador responsable: Paulo Ramos
Descripción: El sistema permitirá el inicio de sesión de un administrador utilizando un usuario y una contraseña predefinidos.	
Validación: Si el usuario y contraseña son correctas se permitirá el acceso al sistema de inventario.	

1.2. Partición de clases equivalentes

Variable	Clase de Equivalencia	Estado	Representante
Usuario	EC1: UsuarioU=usuario	Valido	ChaskiBots1
	EC2: UsuarioU!=usuario	No valido	*****
	EC3: UsuarioU!=usuario	No valido	Radimony7
	EC4: UsuarioU!=usuario	No valido	Tessy68fv
Contraseña	EC1: Contraseña=contraseña	Valido	ChaskiBots1
	EC2: Contraseña!=contraseña	No valido	Chaskibbt3
	EC3: Contraseña!=contraseña	No valido	Prueba315
	EC4: Contraseña!=contraseña	No valido	///*7*3*a/

1.3. Código

```
82 int main() {
83     // Usuario y contraseña predefinidos
84     char usuarioAdmin[] = "ChaskiBots1";
85     char contrasenaAdmin[] = "ChaskiBots1";
86
87     // Variables para ingresar datos
88     char usuario[20];
89     char contrasena[20];
90
91     printf("Inicio de sesión\n");
92     printf("Usuario: ");
93     scanf("%s", usuario);
94     printf("Contraseña: ");
95     scanf("%s", contrasena);
96
97     // Verificar credenciales
98     if (strcmp(usuario, usuarioAdmin) != 0 || strcmp(contrasena, contrasenaAdmin) != 0) {
99         printf("Usuario o contraseña incorrectos.\n");
100         return 1; // Salir del programa si las credenciales son incorrectas
101     }
102
103     printf("Inicio de sesión exitoso.\n");
104 }
```

1.4. Ejecución

- Pass (OK)

```
Inicio de sesi n
Usuario: ChaskiBots1
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n:
```

- Fail (no OK)

```
Inicio de sesi n
Usuario: Radimony7
Contrase a: Prueba315
Usuario o contrase a incorrectos.
-----
```

2. REQ002 A adir un producto

2.1 Historia de usuario

Historia de usuario	
REQ002	Nombre de historia: A�adir un producto
Prioridad: Alta	Programador responsable: Jhon German
Descripci�n: El sistema permitir� al administrador a�adir un nuevo producto en el inventario.	
Validaci�n: Al a�adir un producto se ingresa ID, nombre y cantidad de producto, si son los datos correctos se agrega el producto.	

2.2 Partición de clases equivalentes

Variable	Clase de Equivalencia	Estado	Representante
ID	EC1: ID = ID	Valido	1515
	EC2: ID! = ID	No valido	Placa
	EC3: ID! = ID	No valido	*****
Producto	EC1: Producto=producto	Valido	Placa
	EC2: Producto!=producto	No valido	15151
	EC3: Producto!=producto	No valido	*****
Cantidad	EC1: Cantidad=cantidad	Valido	5
	EC2: Cantidad!=cantidad	No valido	Placa
	EC3: Cantidad!=cantidad	No valido	*****

2.3 Código

```

121 // Validar ID (solo números enteros)
122 printf("Ingrese el ID del producto: ");
123 int id;
124 if (scanf("%d", &id) != 1) {
125     printf("Error en el ID. Debe ser un número entero.\n");
126     while (getchar() != '\n'); // Limpiar el buffer de entrada
127     break;
128 }
129 productos[n].id = id;
130
131 // Validar Nombre (solo caracteres alfabéticos)
132 printf("Ingrese el nombre del producto: ");
133 char nombre[50];
134 scanf("%[^\n]s", nombre); // Leer el nombre con espacios
135 if (!validarNombre(nombre)) {
136     printf("Error en el Nombre. Solo se permiten caracteres alfabéticos.\n");
137     break;
138 }
139 strcpy(productos[n].nombre, nombre);
140
141 // Validar Cantidad (solo números enteros)
142 printf("Ingrese la cantidad del producto: ");
143 int cantidad;
144 if (scanf("%d", &cantidad) != 1) {
145     printf("Error: Introduzca un número entero.\n");
146     while (getchar() != '\n'); // Limpiar el buffer de entrada
147     break;
148 }
149 productos[n].cantidad = cantidad;
150
151 printf("\nProducto agregado:\n");
152 printf("ID: %d\n", productos[n].id);
153 printf("Nombre: %s\n", productos[n].nombre);
154 printf("Cantidad: %d\n", productos[n].cantidad);
155
156 n++; // Incrementar el contador de productos
157 guardar_productos(productos, n); // Guardar Los productos en el archivo
158 break;
159

```

2.4 Ejecución

- **Pass (OK)**

```

Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 1
Ingrese el ID del producto: 1515
Ingrese el nombre del producto: Arduino
Ingrese la cantidad del producto: 5

Producto agregado:
ID: 1515
Nombre: Arduino
Cantidad: 5

```

- **Fail (no OK)**

```

Inicio de sesi n
Usuario: ChaskiBots1
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 1
Ingrese el ID del producto: placa
Error en el ID. Debe ser un n mero entero.

```

```

Inicio de sesi n
Usuario: ChaskiBots1
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 1
Ingrese el ID del producto: 1515
Ingrese el nombre del producto: 45
Error en el Nombre. Solo se permiten caracteres alfab ticos.

```

```

Inicio de sesi n
Usuario: ChaskiBots1
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 1
Ingrese el ID del producto: 1515
Ingrese el nombre del producto: Arduino
Ingrese la cantidad del producto: lspdka454
Error: Introduzca un n mero entero.

```

3. REQ003 Modificar los productos

3.1 Historia de usuario

Historia de usuario	
REQ003	Nombre de historia: Modificar los productos
Prioridad: Alta	Programador responsable: Davis Cobeña
Descripción: El sistema permitirá al administrador sumar o restar la cantidad de unidades de cada producto.	
Validación: Al aumentar o disminuir la cantidad de un producto nos confirma con la cantidad de producto disponible.	

3.2 Partición de clases equivalentes

Variable	Clase de Equivalencia	Estado	Representante
Añadir la cantidad de productos	EC1: Añadir=añadir	Valido	5
	EC2: Añadir!=añadir	No valido	-4
	EC3: Añadir!=añadir	No valido	placa
Disminuir la cantidad de productos	EC1: Disminuir=disminuir	Valido	5
	EC2: Disminuir!=disminuir	No valido	-4
	EC3: Disminuir!=disminuir	No valido	placa

3.3 Codigo

```

2 // Modificar cantidad de producto
3 int id, cantidad, opcionCantidad;
4 printf("Ingrese el ID del producto a modificar: ");
5 scanf("%d", &id);
6
7 int encontrado = 0;
8 for (int i = 0; i < n; i++) {
9     if (productos[i].id == id) {
10         encontrado = 1;
11         do {
12             printf("\n¿Desea agregar o disminuir la cantidad?\n");
13             printf("1. Agregar cantidad\n");
14             printf("2. Disminuir cantidad\n");
15             printf("3. Salir\n");
16             printf("Seleccione una opción: ");
17             scanf("%d", &opcionCantidad);
18
19             if (opcionCantidad == 1) {
20                 printf("Ingrese la cantidad a agregar: ");
21                 if (scanf("%d", &cantidad) != 1 || cantidad < 0) {
22                     printf("Error: Introduzca un número entero no negativo.\n");
23                     while (getchar() != '\n'); // Limpiar el buffer de entrada
24                     continue;
25                 }
26                 productos[i].cantidad += cantidad;
27                 printf("Nueva cantidad: %d\n", productos[i].cantidad);
28             } else if (opcionCantidad == 2) {
29                 printf("Ingrese la cantidad a disminuir: ");
30                 if (scanf("%d", &cantidad) != 1 || cantidad < 0) {
31                     printf("Error: Introduzca un número entero no negativo.\n");
32                     while (getchar() != '\n'); // Limpiar el buffer de entrada
33                     continue;
34                 }
35                 if (cantidad <= productos[i].cantidad) {
36                     productos[i].cantidad -= cantidad;
37                     printf("Nueva cantidad: %d\n", productos[i].cantidad);
38                 } else {
39                     printf("No puedes reducir la cantidad a un valor negativo.\n");
40                 }
41             } else if (opcionCantidad == 3) {
42                 printf("Saliendo...\n");
43             } else {
44                 printf("Opción inválida. Intenta nuevamente.\n");
45             }
46         } while (opcionCantidad != 3);
47         guardar_productos(productos, n); // Guardar los productos en el archivo
48         break;
49     }
50 }
51 if (!encontrado) {
52     printf("Producto con ID %d no encontrado.\n", id);
53 }
54 break;
55 }

```

3.4 Ejecucion

- Pass (OK)

```
Contraseña: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 2
Ingrese el ID del producto a modificar: 634

  Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opci n: 1
Ingrese la cantidad a agregar: 5
Nueva cantidad: 10
```

```
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 2
Ingrese el ID del producto a modificar: 634

  Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opci n: 2
Ingrese la cantidad a disminuir: 3
Nueva cantidad: 2
```

- Fail (no OK)

```
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 2
Ingrese el ID del producto a modificar: 634

  Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opci n: 1
Ingrese la cantidad a agregar: -4
Error: Introduzca un n mero entero no negativo.
```

```
Contrase a: ChaskiBots1
Inicio de sesi n exitoso.

Men  Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opci n: 2
Ingrese el ID del producto a modificar: 634

  Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opci n: 1
Ingrese la cantidad a agregar: pls
Error: Introduzca un n mero entero no negativo.
```

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 2
Ingrese el ID del producto a modificar: 634

¿Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opción: 2
Ingrese la cantidad a disminuir: -4
Error: Introduzca un número entero no negativo.
```

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 2
Ingrese el ID del producto a modificar: 634

¿Desea agregar o disminuir la cantidad?
1. Agregar cantidad
2. Disminuir cantidad
3. Salir
Seleccione una opción: 2
Ingrese la cantidad a disminuir: plca
Error: Introduzca un número entero no negativo.
```

4. REQ 004 Eliminar productos

4.1 Historia de usuario

Historia de usuario	
REQ004	Nombre de historia: Eliminar productos
Prioridad: Alta	Programador responsable: Paulo Ramos
Descripción: El sistema permite al administrador la eliminación de un producto con el ID o nombre.	
Validación: El administrador busca el producto y confirma la eliminación del mismo.	

4.2 Partición de clases equivalentes

Variable	Clase de Equivalencia	Estado	Representante
ID	EC1: ID = ID	Valido	1515
	EC2: ID! = ID	No valido	Placa
	EC3: ID! = ID	No valido	*****
Producto	EC1: Producto=producto	Valido	Placa
	EC2: Producto!=producto	No valido	15151
	EC3: Producto!=producto	No valido	*****

4.3 Codigo

```
1 case 3: {
2     // Eliminar producto
3     int opcionEliminar;
4     printf("¿Desea buscar el producto por ID o por Nombre?\n");
5     printf("1. ID\n");
6     printf("2. Nombre\n");
7     printf("Seleccione una opción: ");
8     scanf("%d", &opcionEliminar);
9
10    if (opcionEliminar == 1 || opcionEliminar == 2) {
11        int id;
12        char nombre[50];
13        int encontrado = 0;
14
15        if (opcionEliminar == 1) {
16            // Buscar por ID
17            printf("Ingrese el ID del producto a eliminar: ");
18            if (scanf("%d", &id) != 1) {
19                printf("Error: Introduzca un número entero.\n");
20                while (getchar() != '\n'); // Limpiar el buffer de entrada
21                break;
22            }
23        } else {
24            // Buscar por nombre
25            printf("Ingrese el nombre del producto a eliminar: ");
26            scanf("%[^\n]s", nombre);
27        }
28
29        // Buscar el producto
30        for (int i = 0; i < n; i++) {
31            if ((opcionEliminar == 1 && productos[i].id == id) ||
32                (opcionEliminar == 2 && strstr(productos[i].nombre, nombre) != NULL)) {
33                encontrado = 1;
34                printf("Producto encontrado: %s (ID: %d, Cantidad: %d)\n",
35                    productos[i].nombre, productos[i].id, productos[i].cantidad);
36
37                // Confirmar eliminación
38                printf("¿Desea eliminar el producto?\n");
39                printf("1: Eliminar producto\n");
```

```

40 printf("2: No eliminar\n");
41 int confirmar;
42 printf("Seleccione una opción: ");
43 scanf("%d", &confirmar);
44
45 if (confirmar == 1) {
46     // Eliminar el producto (desplazar los elementos restantes)
47     for (int j = i; j < n - 1; j++) {
48         productos[j] = productos[j + 1];
49     }
50     n--; // Decrementar el contador de productos
51     printf("Producto eliminado.\n");
52     guardar_productos(productos, n); // Guardar los productos en el archivo
53 } else {
54     printf("El producto no fue eliminado.\n");
55 }
56 break;
57 }
58 }
59
60 if (!encontrado) {
61     if (opcionEliminar == 1) {
62         printf("Producto con ID %d no encontrado.\n", id);
63     } else {
64         printf("No se encontraron productos con el nombre \"%s\".\n", nombre);
65     }
66 }
67 } else {
68     printf("Opción no válida.\n");
69 }
70 break;
71 }

```

4.4 Ejecucion

- Pass (OK)

```

1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 3
¿Desea buscar el producto por ID o por Nombre?
1. ID
2. Nombre
Seleccione una opción: 1
Ingrese el ID del producto a eliminar: 1515
Producto encontrado: Arduino (ID: 1515, Cantidad: 3)
¿Desea eliminar el producto?
1: Eliminar producto
2: No eliminar
Seleccione una opción: 1
Producto eliminado.

```

```

1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 3
¿Desea buscar el producto por ID o por Nombre?
1. ID
2. Nombre
Seleccione una opción: 2
Ingrese el nombre del producto a eliminar: Arduino
Producto encontrado: Arduino (ID: 1515, Cantidad: 5)
¿Desea eliminar el producto?
1: Eliminar producto
2: No eliminar
Seleccione una opción: 1
Producto eliminado.

```

- Fail (no OK)

```

Usuario: ChaskiBots1
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Men. Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 3
¿Desea buscar el producto por ID o por Nombre?
1. ID
2. Nombre
Seleccione una opción: 1
Ingrese el ID del producto a eliminar: Arduino
Error: Introduzca un número entero.

```

```

Usuario: ChaskiBots1
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Men. Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 3
¿Desea buscar el producto por ID o por Nombre?
1. ID
2. Nombre
Seleccione una opción: 2
Ingrese el nombre del producto a eliminar: 1515
No se encontraron productos con el nombre "1515".

```

5. REQ 005 Búsqueda de productos

5.1 Historia de usuario

Historia de usuario	
REQ005	Nombre de historia: Búsqueda de productos
Prioridad: Alta	Programador responsable: Davis Cobeña
Descripción: El sistema permite al administrador la búsqueda de los productos.	
Validación: El administrador busca el producto por ID o el nombre del producto.	

5.2 Partición de clases equivalentes

Variable	Clase de Equivalencia	Estado	Representante
ID	EC1: ID = ID	Valido	1515
	EC2: ID! = ID	No valido	Placa
	EC3: ID! = ID	No valido	*****
Producto	EC1: Producto=producto	Valido	Placa
	EC2: Producto!=producto	No valido	15151
	EC3: Producto!=producto	No valido	*****

5.3 Codigo

```
286 // Buscar producto
287 int opcionBusqueda, id;
288 char nombre[50];
289 printf("\nSeleccione una opción de búsqueda:\n");
290 printf("1. Buscar por ID\n");
291 printf("2. Buscar por Nombre\n");
292 printf("Ingrese opción: ");
293 scanf("%d", &opcionBusqueda);
294
295 if (opcionBusqueda == 1) {
296     printf("Ingrese el ID del producto: ");
297     if (scanf("%d", &id) != 1) {
298         printf("Ingrese el ID correcto (número entero).\n");
299         while (getchar() != '\n'); // Limpiar el buffer de entrada
300         break;
301     }
302     buscar_por_id(productos, n, id);
303 } else if (opcionBusqueda == 2) {
304     printf("Ingrese el nombre del producto: ");
305     scanf("%[^\n]s", nombre); // Para Leer el nombre con espacios
306     if (!validarNombre(nombre)) {
307         printf("Error: Solo se permiten caracteres alfabéticos.\n");
308         break;
309     }
310     buscar_por_nombre(productos, n, nombre);
311 } else {
312     printf("Opción no válida.\n");
313 }
314 break;
315 }
```

5.4 Ejecución

- Pass (OK)

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 4

Seleccione una opción de búsqueda:
1. Buscar por ID
2. Buscar por Nombre
Ingrese opción: 1
Ingrese el ID del producto: 4589
Producto encontrado: Resistencia (ID: 4589, Cantidad: 5)
```

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 4

Seleccione una opción de búsqueda:
1. Buscar por ID
2. Buscar por Nombre
Ingrese opción: 2
Ingrese el nombre del producto: Resistencia
Producto encontrado: Resistencia (ID: 4589, Cantidad: 5)
```

- Fail (no OK)

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 4

Seleccione una opción de búsqueda:
1. Buscar por ID
2. Buscar por Nombre
Ingrese opción: 1
Ingrese el ID del producto: Resistencia
Ingrese el ID correcto (número entero).
```

```
Contraseña: ChaskiBots1
Inicio de sesión exitoso.

Menú Principal:
1. Agregar producto
2. Modificar cantidad de producto
3. Eliminar producto
4. Buscar producto
5. Salir
Seleccione una opción: 4

Seleccione una opción de búsqueda:
1. Buscar por ID
2. Buscar por Nombre
Ingrese opción: 2
Ingrese el nombre del producto: 4589
Error: Solo se permiten caracteres alfabéticos.
```