

# Pruebas de Caja Negra – RF1, RF2, RF4

---

## Requisito Funcional Nº1 – Agregar producto

El programa debe permitir agregar un nuevo producto si el código no está repetido.

### Partición de clases de equivalencia

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
codigo	EC1: código no existe en el inventario	Válido	A123
	EC2: código ya existe en el inventario	No válido	B456
nombre	EC3: nombre no vacío y con texto	Válido	Manzana
	EC4: nombre vacío o solo espacios	No válido	
precio	EC5: precio > 0	Válido	1.50
	EC6: precio ≤ 0 o no numérico	No válido	-2.0, abc
cantidad	EC7: cantidad ≥ 0	Válido	10
	EC8: cantidad negativa o no numérica	No válido	-5, xyz

Captura de la pantalla de la ejecución

Descripción de la ejecución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
U Ingrese el codigo del producto: 123
A Ingrese el nombre del producto: resistencias
S Ingrese el precio del producto: 0.50
  Ingrese la cantidad disponible: 6
  Producto agregado correctamente. Presione ENTER para continuar...|
```

## Requisito Funcional N°2 – Buscar producto por código

El programa debe mostrar la información de un producto existente, si el código se encuentra.

### Partición de clases de equivalencia

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
codigo	EC1: código existe en el inventario	Válido	P001
	EC2: código no existe en el inventario	No válido	P999

Captura de la pantalla de la ejecución

Descripción de la ejecución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
Ingrese el codigo del producto a buscar: 123
```

```
Codigo: 123
```

```
Nombre: resistencias
```

```
Precio: $0.50
```

```
Cantidad disponible: 6
```

```
Presione ENTER para continuar...
```

### Requisito Funcional N°4 – Eliminar producto

El programa debe permitir eliminar un producto si el código existe y se confirma con 'S' o 's'.

#### Partición de clases de equivalencia

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
codigo	EC1: código existe en el inventario	Válido	C777
	EC2: código no existe en el inventario	No válido	C999
confirmación	EC3: confirmación = 'S' o 's'	Válido	s
	EC4: confirmación = 'N' o 'n'	Válido	N
	EC5: confirmación ≠ 'S', 's', 'N', 'n'	No válido	x

Captura de la pantalla de la ejecución

Descripción de la ejecución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
Ingrese el codigo del producto a eliminar: 123
```

```
Ingrese el codigo del producto a eliminar: 123
Seguro que desea eliminar el producto 'resistencias'? (S/N): N
Eliminacion cancelada.
Presione ENTER para continuar...
```

```
Ingrese el codigo del producto a eliminar: 123
Seguro que desea eliminar el producto 'resistencias'? (S/N): S
Producto eliminado correctamente.
Presione ENTER para continuar...
```

Partición de clases equivalentes

TABLA 1:

Requisito Funcional N°5 El programa debe permitir actualizar la información de un producto existente (nombre, precio y cantidad).

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
codigo confirmación	EC1: codigo = codigo	Válido	1234
	EC2: codigo != codigo	No Válido	6532
	EC3: confirmacion =S o S	Válido	s
	EC4: confirmacion=N o n	Válido	n
	EC5 confirmacion!=S, s, N, n	No válido	wer

## Captura de la pantalla de la ejecución

Descripción de la ejecución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
Ingrese el codigo del producto a actualizar: 1234
Seguro que desea actualizar el producto 'chips'? (S/N): |
```

```
Ingrese el codigo del producto a actualizar: 6532
Producto no encontrado.
Presione ENTER para continuar...|
```

```
--- MENU DE ACTUALIZACION ---
1. Actualizar nombre
2. Actualizar precio
3. Actualizar cantidad
4. Actualizar todo
5. Cancelar
Seleccione una opcion: |
```

```
Ingrese el codigo del producto a actualizar: 1234
Seguro que desea actualizar el producto 'chips'? (S/N): n
Actualizacion cancelada.
Presione ENTER para continuar...|
```

```
--- MENU DE INVENTARIO ---
1. Agregar producto
2. Buscar producto por codigo
3. Mostrar todos los productos
4. Eliminar producto
5. Actualizar producto
6. Realizar venta
7. Ver historial de ventas
8. Cambiar contrasena
9. Salir
Opcion: Opcion invalida. Ingrese un numero entre 1 y 9: |
```



Partición de clases equivalentes

TABLA 1:

Requisito Funcional N°6 El programa debe permitir realizar ventas de los productos existentes, actualizando la cantidad disponible al final de la transacción.

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
codigo  cantidadVenta	EC1: codigo = codigo	Válido	1234
	EC2: codigo != codigo	No Válido	1345
	EC3: cantidadVenta= [1, cantidad]	Válido	5
	EC4: cantidadVenta= [1, cantidad]	NO válido	25

## Captura de la pantalla de la ejecución

Descripción de la ejecución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
Ingrese el codigo del producto a vender: 1234

--- INFORMACION DEL PRODUCTO ---
Codigo: 1234
Nombre: chips
Precio Unitario: $1.00
Cantidad Disponible: 20

Ingrese la cantidad a vender: |
```

```
Ingrese el codigo del producto a vender: 1345
Producto no encontrado.
Presione ENTER para continuar...|
```

```
Ingrese el codigo del producto a vender: 1234

--- INFORMACION DEL PRODUCTO ---
Codigo: 1234
Nombre: chips
Precio Unitario: $1.00
Cantidad Disponible: 20

Ingrese la cantidad a vender: 5
Venta registrada.
Total: $5.00
Presione ENTER para continuar...|
```

```
Ingrese el codigo del producto a vender: 1234

--- INFORMACION DEL PRODUCTO ---
Codigo: 1234
Nombre: chips
Precio Unitario: $1.00
Cantidad Disponible: 15

Ingrese la cantidad a vender: 25
Error: No hay suficiente stock disponible.
Ingrese la cantidad a vender: |
```

Partición de clases equivalentes

TABLA 1:

Requisito Funcional N°7 El programa debe permitir visualizar las ventas registradas incluyendo la cantidad vendida, precio unitario y precio total

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
opcion	EC1: opcion= 7	Válido	7
	EC2: opcion != 7	No Válido	20

Captura de la pantalla de la ejecución

Descripción de la elución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
--- HISTORIAL DE VENTAS ---

Codigo: 1234
Nombre: chips
Cantidad: 5
Precio Unitario: $1.00
Total: $5.00

Codigo: 1234
Nombre: chips
Cantidad: 5
Precio Unitario: $1.00
Total: $5.00

Presione ENTER para continuar...|
```

```
--- MENU DE INVENTARIO ---
1. Agregar producto
2. Buscar producto por codigo
3. Mostrar todos los productos
4. Eliminar producto
5. Actualizar producto
6. Realizar venta
7. Ver historial de ventas
8. Cambiar contrasena
9. Salir
Opcion: 20
Opcion invalida. Ingrese un numero entre 1 y 9: |
```

## Partición de clases equivalentes

**TABLA 1:**

Requisito Funcional N°8 El programa debe permitir que el usuario cambie su contraseña a voluntad

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
nueva confirmacion	EC1: nueva= confirmacion	Válido	contraseña1, contraseña1
	EC2: nueva != confirmacion	No Válido	contraseña2, contraseña3

### Captura de la pantalla de la ejecución

Descripción de la elución de acuerdo a los casos de prueba de la tabla 1, caso pass (Ok) y fail (no OK)

```
--- MENU DE INVENTARIO ---
1. Agregar producto
2. Buscar producto por codigo
3. Mostrar todos los productos
4. Eliminar producto
5. Actualizar producto
6. Realizar venta
7. Ver historial de ventas
8. Cambiar contraseña
9. Salir
Opcion: 8
Ingresa la nueva contraseña: contraseña1
Confirme la nueva contraseña: contraseña1
Contraseña cambiada correctamente.
Presione ENTER para continuar...|
```

```
--- MENU DE INVENTARIO ---
1. Agregar producto
2. Buscar producto por codigo
3. Mostrar todos los productos
4. Eliminar producto
5. Actualizar producto
6. Realizar venta
7. Ver historial de ventas
8. Cambiar contraseña
9. Salir
Opcion: 8
Ingresa la nueva contraseña: contraseña2
Confirme la nueva contraseña: ccontraseña3
Las contraseñas no coinciden. No se hizo el cambio.
Presione ENTER para continuar...|
```