

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 contrasena
9. Salir
Opcion: 8
Ingrese la nueva contrasena: contraseña1
Confirme la nueva contrasena: contraseña1
Contrasena 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 contrasena
9. Salir
Opcion: 8
Ingrese la nueva contrasena: contraseña2
Confirme la nueva contrasena: ccontraseña3
Las contrasenas no coinciden. No se hizo el cambio.
Presione ENTER para continuar...|
```