

# Prueba de Caja Negra

---

*“Título proyecto: Caso venta de dulces”*

**Integrantes:**

**Collaguazo Pablo  
Rodríguez Jeicol  
Vilaña Anthony**

**Fecha: 2025-01-09**

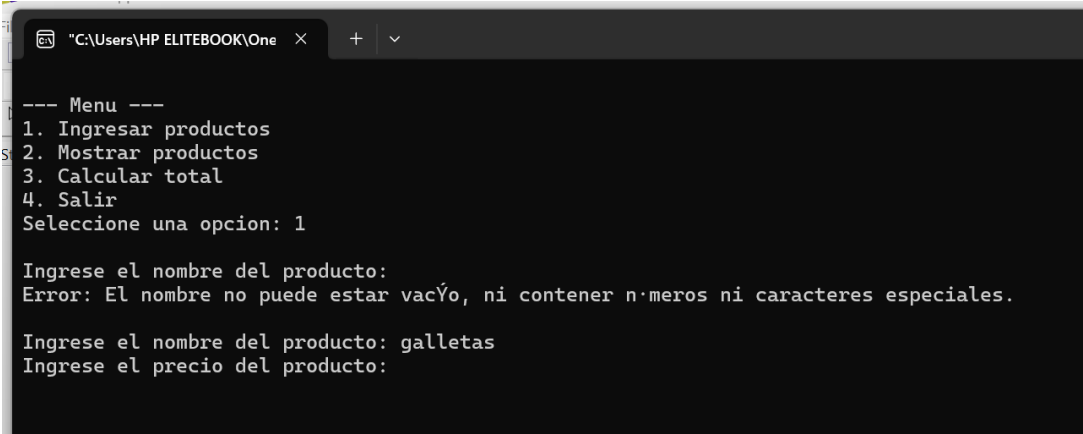
# Partición de clases equivalentes

## RQ001 Registro de productos

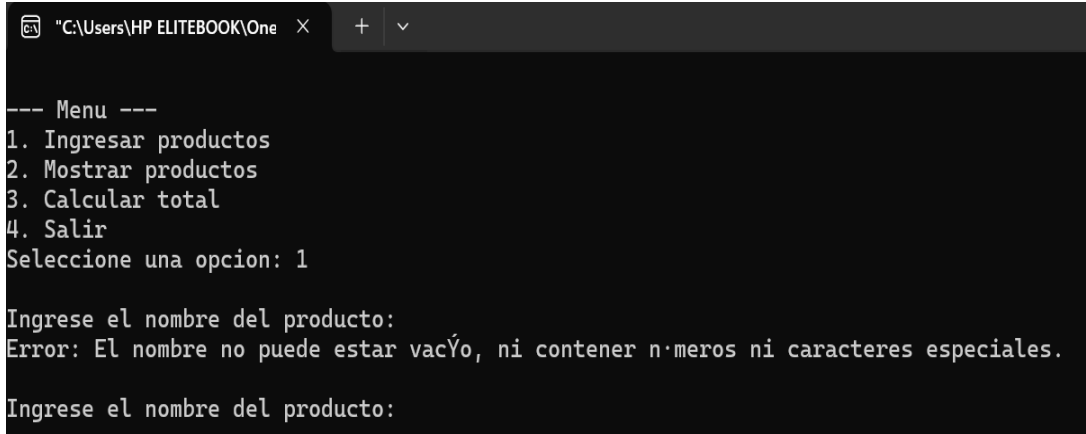
VARIABLE	CLASES DE EQUIVALENCIA	ESTADO	REPRESENTANTE
REGISTRO DE PRODUCTOS	EC: NOMBRE VALIDO	VALIDO	Dulces
	EC: NOMBRE VACIO	NO VALIDO	"....."(espacio en blanco)
	EC: CONTIENE CARACTERES ESPECIALES	NO VALIDO	Caracteres especiales: @, +,/
	EC: CONTIENE NUMEROS	NO VALIDO	1.50, 2, 5 (caracteres numéricos)

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <iomanip> // Para formatear la salida
5  #include <limits> // Para controlar la entrada de datos
6  #include <cctype> // Para funciones relacionadas con caracteres
7
8  using namespace std; // Usar el espacio de nombres estándar
9
10 // Estructura para representar un producto
11 struct Producto {
12     string nombre;
13     float precio;
14 };
15
16 // Función para validar que el nombre no esté vacío, no contenga números ni caracteres especiales
17 bool esNombreValido(const string& nombre) {
18     if (nombre.empty()) {
19         return false;
20     }
21     for (char c : nombre) {
22         if (isdigit(c) || !isalnum(c) && !isspace(c)) { // Verifica si hay números o caracteres no alfanuméricos
23             return false;
24         }
25     }
26     return true;
27 }
```

### Nombre valido



### NOMBRE VACIO



Contiene caracteres especiales

```
"C:\Users\HP ELITEBOOK\One  X + v

--- Menu ---
1. Ingresar productos
2. Mostrar productos
3. Calcular total
4. Salir
Seleccione una opcion: 1

Ingrese el nombre del producto: &$
Error: El nombre no puede estar vacío, ni contener números ni caracteres especiales.
Ingrese el nombre del producto:
```

Contiene números

```
--- Menu ---
1. Ingresar productos
2. Mostrar productos
3. Calcular total
4. Salir
Seleccione una opcion: 1

Ingrese el nombre del producto: 1232
Error: El nombre no puede estar vacío, ni contener números ni caracteres especiales.
Ingrese el nombre del producto: |
```

RQ001-1 Lista de productos

VARIABLE	CLASES DE EQUIVALENCIA	ESTADO	REPRESENTANTE
LISTA PRODUCTOS	EC: LISTA=LISTA	VALIDO	Galletas Chocolate Agua
	EC: ¡LISTA! =LISTA	NO VALIDO	2514 "....." Agua
	EC: ¡LISTA! =LISTA	NO VALIDO	Caracteres especiales: @, +,/

```
do {
    Producto p;
    string nombreProducto;

    // Validar la entrada del nombre del producto
    cin.ignore(); // Limpiar el buffer de entrada
    do {
        cout << "\nIngrese el nombre del producto: ";
        getline(cin, nombreProducto);

        if (!esNombreValido(nombreProducto)) {
            cout << "Error: El nombre no puede estar vacío, ni contener números ni caracteres especiales." << endl;
        }
    } while (!esNombreValido(nombreProducto)); // Repetir hasta que el nombre sea válido

    p.nombre = nombreProducto;

    // Ingresar y validar el precio del producto
    p.precio = ingresarPrecio();

    // Agregar el producto a la lista
    productos.push_back(p);

    cout << "Producto agregado correctamente." << endl;

    // Preguntar si desea agregar otro producto
    cout << "¿Desea agregar otro producto? (s/n): ";
    cin >> continuar;

} while (continuar == 's' || continuar == 'S');
```

--- Menu ---

1. Ingresar productos
2. Mostrar productos
3. Calcular total
4. Salir

Seleccione una opcion: 2

Lista de productos ingresados:

Nombre	Precio
galletas	\$50.00
chocolates	\$1.00
agua	\$50.00

## Precios

VARIABLE	CLASES DE EQUIVALENCIA	ESTADO	REPRESENTANTE
COSTO DEL PRODUCTO	EC: PPRECIO=PRECIO	VALIDO	1.50, 2.00, 0.50
	EC: PPRECIO!=PRECIO	NO VALIDO	-12,1, -5. -1. (valores menores a 0)
	EC: PPRECIO!=PRECIO	NO VALIDO	"....."(espacio en blanco)
	EC: PPRECIO!=PRECIO	NO VALIDO	F A L J (caracteres)

```

29 // Función para validar el precio (número mayor que 0)
30 float ingresarPrecio() {
31     float precio;
32     while (true) {
33         cout << "Ingrese el precio del producto: ";
34         cin >> precio;
35
36         if (cin.fail() || precio <= 0) { // Si la entrada falla (no es un número) o es menor o igual a 0
37             cout << "Error: ingrese un valor mayor que 0." << endl;
38             cin.clear(); // Limpiar el estado de error de cin
39             cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Descartar la entrada incorrecta
40         } else {
41             break; // Si el precio es válido, salir del bucle
42         }
43     }
44     return precio;
45 }

```

```

Ingrese el nombre del producto: galletas
Ingrese el precio del producto: 50
Producto agregado correctamente.
¿Desea agregar otro producto? (s/n): s

Ingrese el nombre del producto: chocolates
Ingrese el precio del producto:
1
Producto agregado correctamente.
¿Desea agregar otro producto? (s/n): s

Ingrese el nombre del producto: agua
Ingrese el precio del producto: 50
Producto agregado correctamente.
¿Desea agregar otro producto? (s/n): n

--- Menu ---
1. Ingresar productos
2. Mostrar productos
3. Calcular total
4. Salir
Seleccione una opcion: 2

Lista de productos ingresados:
Nombre          Precio
-----
galletas        $50.00
chocolates      $1.00
agua            $50.00

```