Prueba de Caja Negra

"Título proyecto: Caso venta de dulces"

Integrantes:

Collaguazo Pablo
Rodríguez Jeicol
Vilaña Anthony

Fecha: 2025-01-09

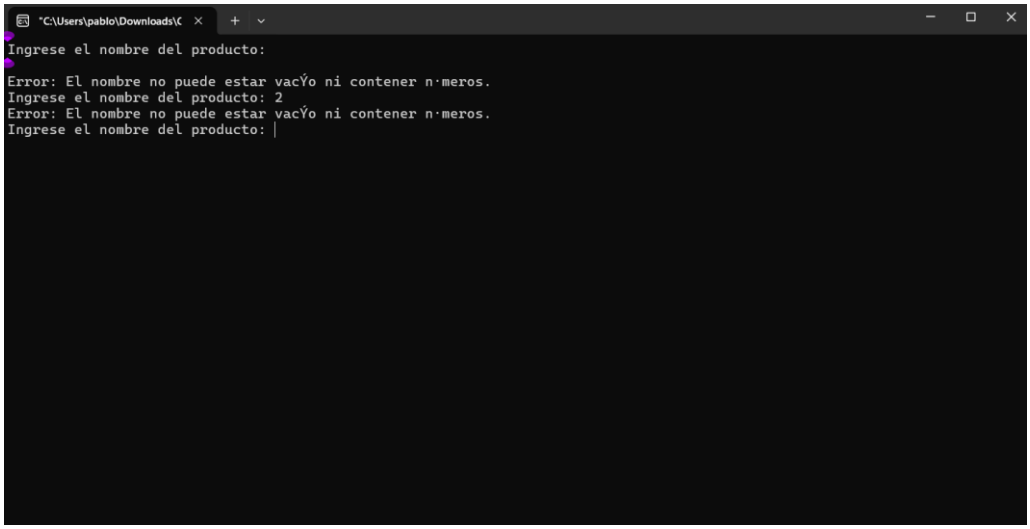
Partición de clases equivalentes

RQ001 Registro de productos

VARIABLE	CLASES DE EQUIVALENCIA	ESTADO	REPRESENTANTE
REGISTRO DE PRODUCTOS	EC: PRODUCTO=PRODUCTO	VALIDO	Dulces
	EC: PRODUCTO! =PRODUCTO	NO VALIDO	“…….”(espacio en blanco)
	EC: PRODUCTO! =PRODUCTO	NO VALIDO	Caracteres especiales: @, +,/
	EC: PRODUCTO! =PRODUCTO	NO VALIDO	1.50, 2, 5 (caracteres numéricos)

```cpp
// Función para validar que el nombre no esté vacío y no contenga números
bool esNombreValido(const string& nombre) {
    if (nombre.empty()) {
        return false;
    }
    for (char c : nombre) {
        if (isdigit(c)) { // Verifica si hay algún número en el nombre
            return false;
        }
    }
    return true;
}
```

Validación para ingresar los productos



En caso de contener número o caracteres especiales se pedirá ingresar nuevamente el nombre.

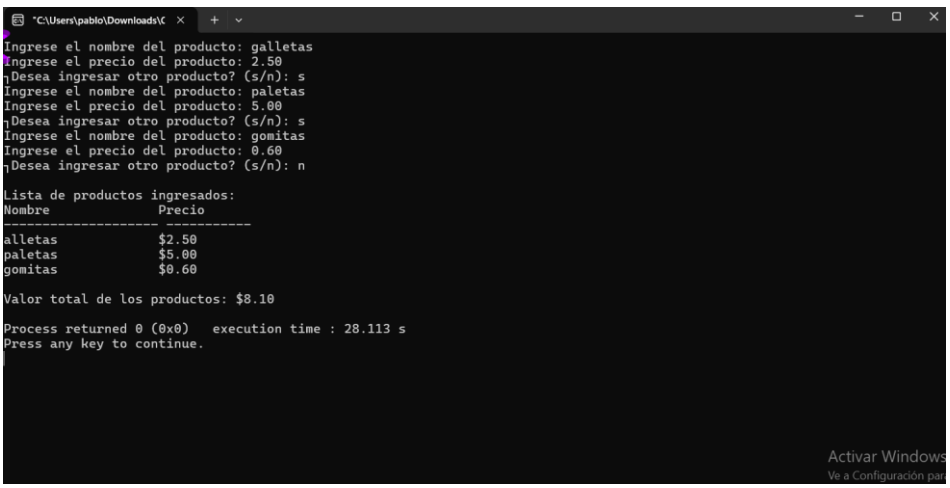
RQ001-1 Lista de productos

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

```
// Mostrar la lista de productos
cout << "\nLista de productos ingresados:" << endl;
cout << setw(20) << left << "Nombre" << setw(10) << "Precio" << endl;
cout << "-------------------- ----------" << endl;

for (const auto& producto : productos) {
    cout << setw(20) << left << producto.nombre << "$" << fixed << setprecision(2) << producto.precio << endl;
}
```

Generar lista automática de los productos



Lista generada

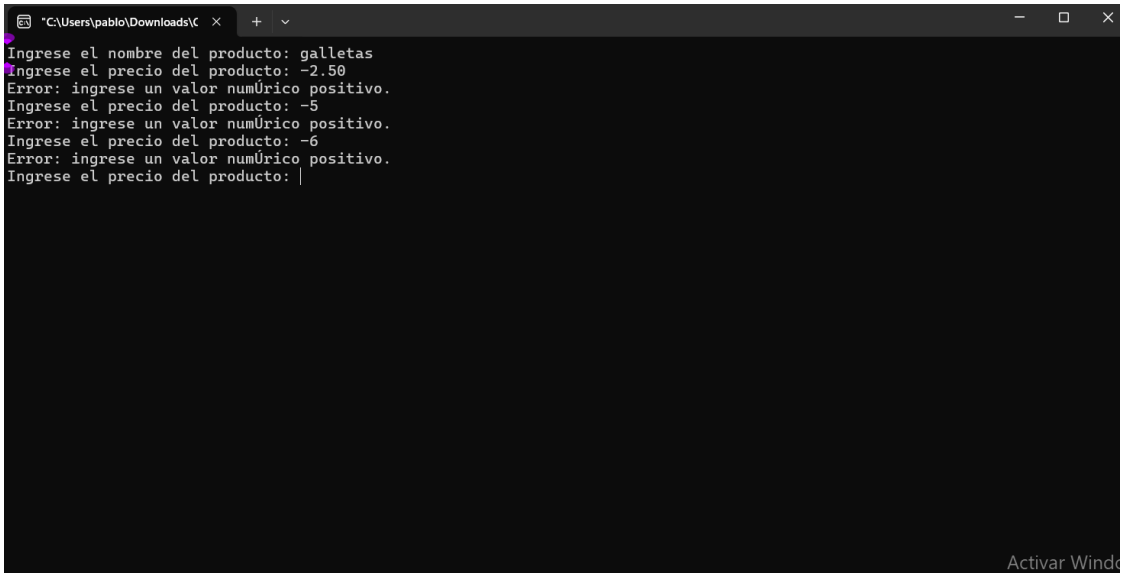
Precios

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)

```cpp
// Función para validar el precio (número positivo y no letras)
float ingresarPrecio() {
    float precio;
    while (true) {
        cout << "Ingrese el precio del producto: ";
        cin >> precio;

        if (cin.fail() || precio < 0) { // Si la entrada falla (no es un número) o es negativa
            cout << "Error: ingrese un valor numérico positivo." << endl;
            cin.clear(); // Limpiar el estado de error de cin
            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Descartar la entrada incorrecta
        } else {
            break; // Si el precio es válido, salir del bucle
        }
    }
    return precio;
}
```

Validación para escribir precios correctos



En caso de escribir número negativo de pedirá que ingrese un valor válido.