



UNIVERSIDAD DE LAS AMÉRICAS (UDLA)  
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS  
CARRERA DE INGENIERÍA EN CIBERSEGURIDAD

**ASIGNATURA:** PROGRAMACION 1  
**DOCENTE:** EDDY MAURICIO ARMAS

**ESTUDIANTES:** Jeremy Vasquez, Israel zurita

**FECHA:** 02/10/2025

## 1. Introducción

El presente informe tiene como propósito exponer los aprendizajes adquiridos en la asignatura de “Algoritmos”, a través de la implementación en el lenguaje C.

El ejercicio propuesto consiste en diseñar un sistema para un pequeño negocio que permita gestionar la venta de un único producto dentro de la tienda. Dicho sistema cuenta con funcionalidades como registro, modificación, venta, reabastecimiento y consulta de información.

Este programa resulta bastante práctico en la actualidad, ya que facilita la administración de manera más eficiente gracias a la automatización de procesos. Además, el desarrollo de este proyecto nos brinda la oportunidad de reforzar y aplicar conocimientos previos, preparándonos para futuros desafíos en el área.

## 2. Desarrollo del Código

Para dar solución al problema se planteó un algoritmo en C utilizando una estructura (struct) para representar el producto, con atributos como: ID, nombre, stock, precio unitario y ganancia acumulada. Este enfoque permite que el sistema sea sencillo de mantener y extender en cuanto a funciones y características.

Entre los aspectos técnicos más relevantes se encuentran:

**Validación de datos:** se implementaron funciones que garantizan que las entradas del usuario sean válidas (valores positivos y de tipo correcto).

**Estructuras de control:** se usaron condicionales if y switch para manejar las restricciones y gestionar las diferentes opciones del menú.

**Operaciones matemáticas:** se programó la lógica para calcular descuentos, actualizar existencias después de cada venta y acumular las ganancias.

**Bucles de repetición:** mediante un while el programa permanece en ejecución hasta que el usuario decida finalizarlo.

A continuación, se presentan algunos fragmentos representativos del código desarrollado (el programa completo está disponible en el repositorio de GitHub).

## Imagen del código

```
1  #include <stdio.h>
2  #include <cadena.h>
3
4  definición de tipo estructura{
5      entero identificación;
6      carbonizarse nombre[50];
7      entero existencias;
8      flotar precio;
9      flotar ganancias;
10 }Producto;
11
12 vacio registrarProducto(Producto *pag) {
13     printf("Ingrese el ID del producto: ");
14     escaneo("%d",&pag->identificación);
15     obtener carácter();// limpiar buffer
16
17     printf("Ingrese el nombre del producto: ");
18     se olvida(pag->nombre,tamaño de(pag->nombre),entrada estándar);
19     pag->nombre[strcspn(pag->nombre,"\\norte")] = '\\0';// quitar salto de línea
20
21     printf("Ingrese la cantidad en stock: ");
22     escaneo("%d",&pag->existencias);
23
24     printf("Ingrese el precio unitario: ");
25     escaneo("%F",&pag->precio);
26
27     pag->ganancias = 0.0;
28     printf("\\nProducto registrado correctamente.\\n\\n");
29 }
30
31
32 vacio venderProducto(Producto *pag) {
33     entero cantidad;
34     printf("Ingrese la cantidad a vender: ");
35     escaneo("%d",&cantidad);
36 }
```

```

37     si(cantidad<=0) {
38         printf("Cantidad no válida.\n");
39     }demás si(cantidad > pag->existencias) {
40         printf("Stock insuficiente. Solo hay %d unidades disponibles.\n",pag->existencias);
41     }demás{
42         pag->existencias -= cantidad;
43         flotar venta = cantidad * pag->precio;
44         pag->ganancias += venta;
45         printf("Venta realizada con éxito. Ganancia: $%.2f\n",venta);
46     }
47 }
48 }
49
50 vacio reabastecerProducto(Producto *pag) {
51     entero cantidad;
52     printf("Ingrese la cantidad a agregar al stock: ");
53     escaneo("%d",&cantidad);
54
55     si(cantidad > 0) {
56         pag->existencias += cantidad;
57         printf("Reabastecimiento exitoso. Nuevo stock: %d\n",pag->existencias);
58     }demás{
59         printf("Cantidad no válida.\n");
60     }
61 }
62 }
63
64 vacio consultarProducto(Producto pag) {
65     printf("\n--- Información del Producto ---\n");
66     printf("Identificación: %d\n",pag.identificación);
67     printf("Nombre: %s\n",pag.nombre);
68     printf("Existencias: %d\n",pag.existencias);
69     printf("Precio unitario: $%.2f\n",pag.precio);
70     printf("Ganancias acumuladas: $%.2f\n\n",pag.ganancias);

```

---

```

71     }
72
73     entero principal() {
74         Producto producto;
75         entero opción;
76         entero registrado = 0;
77
78         hacer{
79             printf("\n===== MENÚ =====\n");
80             printf("1. Registrar producto\n");
81             printf("2. Vender producto\n");
82             printf("3. Reabastecer producto\n");
83             printf("4. Consultar información\n");
84             printf("5. Mostrar ganancias\n");
85             printf("6. Salir");
86             printf("Seleccione una opción: ");
87             escaneo("%d",&opción);
88
89
90             cambiar(opción) {
91                 caso 1:
92                     registradorProducto(&producto);
93                     registrado = 1;
94                     romper;
95                 caso 2:
96                     si(registrado)venderProducto(&producto);
97                     demás printf("Primero debe registrar el producto.\n");
98                     romper;
99                 caso 3:
100                     si(registrado)reabastecerProducto(&producto);
101                     demás printf("Primero debe registrar el producto.\n");
102                     romper;
103                 caso 4:
104                     si(registrado)consultarProducto(producto);
105                     demás printf("Primero debe registrar el producto.\n");
106                     romper;
107                 caso 5:
108                     si(registrado)printf("Ganancias acumuladas: $%.2f\n",producto.ganancias);
109                     demás printf("Primero debe registrar el producto.\n");
110                     romper;
111                 caso 6:
112                     printf("Saliendo del programa...\n");
113                     romper;
114                 por defecto:
115                     printf("Opción no válida.\n");
116             }
117         }mientras(opción != 6);
118
119         devolver 0;
120     }

```

---

**Enlace al repositorio GitHub:**

[https://github.com/JeremyVasquez/taller\\_n-1\\_vasquez\\_zurita.git](https://github.com/JeremyVasquez/taller_n-1_vasquez_zurita.git)

### Verificación y Pruebas

Con el fin de garantizar el buen desempeño del sistema, se llevaron a cabo diferentes pruebas de validación, entre las cuales se incluyen:

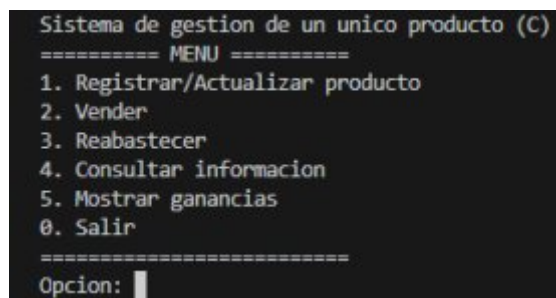
**Venta sin stock suficiente:** el sistema impide realizar la operación y notifica al usuario cuántas unidades se encuentran disponibles.

**Reabastecimiento:** al añadir nuevas cantidades, el inventario se incrementa de forma correcta y este cambio se refleja en las consultas posteriores.

**Cálculo de utilidades:** en cada transacción de venta el programa descuenta las unidades vendidas, aplica el descuento en caso de que corresponda y actualiza el monto total de las ganancias acumuladas.

**Control de entradas incorrectas:** se comprobó que el sistema rechaza valores negativos, cantidades excesivamente elevadas y datos no numéricos.

### Imágenes de la ejecución del código



```
Sistema de gestion de un unico producto (C)
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: █
```



```
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: 1
```

```
Opcion: 1

=== Registro/Actualizacion de Producto ===
ID (entero): 20070412
Nombre (max 99 chars): RTX 3016 NVIDIA 8GB
Cantidad en stock (entero >= 0): 20
Precio unitario (>= 0): 400
Producto registrado/actualizado correctamente.
```

## Ejecución del programa prueba 2

```
Sistema de gestion de un unico producto (C)
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: █
```

```
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: 2
```

```
=====
Opcion: 2

=== Venta ===
Unidades a vender (>= 1): 10
Descuento (%) para esta venta (0-100): 2
Venta realizada: 10 unidad(es) de 'RTX 3016 NVIDIA 8GB'
Precio bruto: $4000.00 | Descuento: $80.00 | Total: $3920.00
Stock restante: 10
```

### Ejecución del programa prueba 3

```
Sistema de gestion de un unico producto (C)
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: █
```

```
===== MENU =====
1. Registrar/Actualizar producto
2. Vender
3. Reabastecer
4. Consultar informacion
5. Mostrar ganancias
0. Salir
=====
Opcion: 3
```

```
=====
Opcion: 3

=== Reabastecer ===
Unidades a agregar (>= 1): 25
Reabastecimiento exitoso. Stock actual: 35
```



## Conclusiones

La elaboración de este programa nos permitió afianzar conocimientos en programación con C, especialmente en el uso de estructuras, funciones, validaciones y controles de flujo.

Entre los aprendizajes más relevantes se pueden mencionar:

La relevancia de realizar una validación adecuada de las entradas del usuario.

La ventaja de dividir el código en funciones específicas, lo cual mejora la comprensión y el mantenimiento del sistema.

La aplicación práctica de estructuras como struct para representar y organizar los datos.

Los principales desafíos que se presentaron fueron:

Diseñar la lógica matemática necesaria para calcular correctamente las ganancias y aplicar descuentos en cada transacción.

Mantener el programa estructurado y entendible, apoyándose en menús y mensajes claros para el usuario.

En síntesis, este proyecto no solo permitió dar cumplimiento a las indicaciones planteadas, sino que también representó una experiencia enriquecedora para fortalecer las habilidades de resolución de problemas mediante la programación.