

Universidad Antonio Nariño



Asignatura: Lógica computacional

**Facultad FIMEB Asignatura:
Lógica Computacional
Proyecto de Software – Guía del Usuario**

**Título del proyecto
Sistema de Gestión de Inventario para Tienda Comercial**

Autores

**Shirley Mora
Jennifer Rivera Juan
José Martínez**

2025

Sistema de Gestión de Inventario para Tienda Comercial

Guía del Usuario Versión

1.0

**Proyecto desarrollado en el marco de la asignatura
Lógica Computacional**

IV. Prefacio

Esta guía del usuario proporciona información detallada sobre el sistema de gestión de inventario para una tienda desarrollado como proyecto final de la asignatura Lógica Computacional. El documento contiene los requisitos del sistema, y una explicación general sobre las funcionalidades del programa, Los archivos fuente han sido desarrollados en lenguaje C++, Esta guía te orientará para navegar e interactuar con el sistema de forma eficiente.

V. Introducción

MercaPlus es un sistema de gestión diseñado para ayudar a pequeños negocios a administrar su inventario y ventas de manera eficiente. Su objetivo principal es facilitar el control preciso de los productos, registrar ventas de manera rápida y sencilla, conocer cuándo es necesario reponer productos y obtener reportes o estadísticas sobre las ventas. Es especialmente útil para tiendas que manejan productos de droguería, papelería y supermercado, ya que calcula automáticamente el precio final con el IVA correspondiente según la categoría del producto.

Versión	Fecha	Cambios realizados	Observaciones
1.0	17/05/2025 https://onlinegdb.com/3ay7bEa7A	Implementación inicial del sistema MERCAPLUS con inventario y funciones básicas: visualizar, vender, abastecer, estadísticas.	Versión base funcional

1.1	<p>30/05/2025</p> <p>https://onlinegdb.com/TkJ4TMRe2</p>	<ul style="list-style-type: none"> - Agregar producto nuevo con validaciones - Cálculo automático del precio con IVA según tipo - Mejor manejo de entradas y buffers - Validaciones para cantidades negativas y stock - Visualización mejorada con tablas formateadas - Estadísticas ampliadas (más y menos vendido, totales, promedio) - Mensajes claros y confirmaciones - Estructura de código organizada y funciones separadas - Pantalla de bienvenida y pausa inicial 	<p>Mejoras en usabilidad, validaciones y estructura</p>
-----	--	--	---

VI. Requisitos técnicos previos

Conocimientos mínimos del usuario:

Para utilizar MercaPlus, el usuario debe tener conocimientos básicos de lectura, escritura y estar familiarizado con el uso de computadoras. No se requiere experiencia avanzada, pero es útil que conozca los productos que maneja el negocio para registrar correctamente las ventas y actualizaciones de inventario.

Requisitos técnicos previos:

El sistema puede ejecutarse en computadoras con Windows 7 o versiones más recientes, aunque también funciona en Linux y MacOS. Se recomienda al menos 2 GB de memoria RAM y 500 MB de espacio en disco. Si se desea modificar el código, será necesario un compilador de C++.

Mecanismo para acceder al sistema:

MercaPlus no requiere conexión a internet para su funcionamiento básico. El programa se ejecuta directamente desde la computadora donde esté instalado, lo que lo hace accesible incluso en lugares con conectividad limitada.

VII. Instalación y configuración

El proceso de instalación es sencillo. Solo se necesita copiar el archivo del programa en una carpeta de la computadora. Si se van a realizar modificaciones, debe instalarse un compilador de C++. Una vez listo, el sistema está listo para usarse sin configuraciones adicionales.

VIII. Guía de uso

Funciones principales:

A. Visualizar productos: Esta opción muestra una lista completa de todos los productos en inventario, incluyendo su nombre, tipo, cantidad disponible, cantidad mínima requerida y precios (base y con IVA).

Pasos para acceder:

1. En el menú principal, seleccionar la opción 1.
2. El sistema mostrará automáticamente la lista completa

B. Vender un producto: Permite registrar una venta y actualizar automáticamente el inventario.

Pasos para acceder:

1. Seleccionar opción 2 en el menú principal
2. Escribir el nombre exacto del producto
3. Ingresar la cantidad vendida
4. El sistema confirmará la venta y mostrará el total a pagar

C. Abastecer producto: Si un producto está por agotarse, esta función permite reponerlo.

Pasos para acceder:

1. Seleccionar opción 3 en el menú
2. Escribir el nombre del producto
3. Ingresar la cantidad que se está agregando
4. El sistema mostrará la nueva cantidad disponible

D. Cambiar producto: Si algún dato de un producto necesita modificarse (nombre, tipo, cantidad mínima o precio), esta opción permite editarlo fácilmente.

Pasos para acceder:

1. Seleccionar opción 4
2. Escribir el nombre del producto a modificar

3. Ingresar los nuevos datos (nombre, tipo, cantidad mínima, precio)
4. El sistema actualizará automáticamente la información

E. Ver estadísticas: Muestra información importante sobre las ventas realizadas, como el producto más y menos vendido, entre otros.

Pasos para acceder:

1. Seleccionar opción 5
2. El sistema mostrará:

Producto más vendido

Producto menos vendido

Total de ventas en dinero

Promedio de venta por unidad

F. Agregar nuevo producto: Cuando se incorpora un nuevo artículo al negocio, esta función permite registrarlo en el sistema con todos sus detalles, incluyendo el cálculo automático del precio con IVA.

Pasos para acceder:

1. Seleccionar opción 6
2. Ingresar todos los datos del producto:

Nombre (no debe existir ya)

Tipo (droguería, papelería, supermercado)

Cantidad actual

Cantidad mínima (no menor a 1)

Precio base

El sistema calculará automáticamente el precio con IVA

G. Salir: Finaliza la ejecución del programa de manera segura.

IX. Sección de preguntas frecuentes (FAQ)

1. ¿Qué tipo de productos puedo registrar en MercaPlus?

Puedes registrar cualquier tipo de producto. El sistema incluye soporte específico para productos de droguería, papelería y supermercado, ya que aplica automáticamente el IVA correspondiente según el tipo.

2. ¿Cómo se calcula el precio con IVA?

El sistema calcula automáticamente el precio final con IVA al momento de agregar o modificar un producto:

Droguería: 12%

Papelería: 16%

Supermercado: 4%

Si el tipo ingresado no corresponde a ninguno de estos, se mantendrá el precio base (sin IVA).

3. ¿Qué sucede si ingreso un producto que ya existe?

El sistema no permite duplicar productos. Si intentas registrar un producto con el mismo nombre, mostrará un mensaje de error.

4. ¿Puedo modificar los datos de un producto ya registrado?

Sí. Puedes cambiar el nombre, tipo, cantidad mínima y precio base desde la opción "**Cambiar Producto**". El sistema recalcula automáticamente el nuevo precio con IVA.

5. ¿Qué pasa si intento vender más unidades de las disponibles en inventario?

El sistema no permitirá completar la venta y mostrará un mensaje indicando que no hay suficiente stock.

6. ¿Cómo sé cuándo debo reabastecer un producto?

Puedes revisar la cantidad actual desde la opción "**Visualizar productos**". Si la cantidad es menor o igual al mínimo registrado, se recomienda abastecerlo.

7. ¿Cómo reabastezco un producto?

Usa la opción "**Abastecer Producto**", escribe el nombre del producto y la cantidad que deseas agregar. El inventario se actualizará automáticamente.

8. ¿Puedo eliminar un producto del inventario?

En esta versión no se permite eliminar productos directamente. Se recomienda editar el producto y cambiar su cantidad a cero si ya no se usa.

9. ¿Se guardan los datos cuando cierro el programa?

No. Este sistema no implementa aún un mecanismo de guardado en archivos o bases de datos. Al cerrar el programa, los datos ingresados se pierden.

10. ¿Cómo puedo consultar las ventas realizadas?

La opción "**Calcular Estadísticas**" muestra el producto más vendido, el menos vendido, el total acumulado de ventas y el promedio por unidad. Sin embargo, no muestra el historial detallado de cada venta individual.

11. ¿Qué hacer si el sistema no reconoce un producto que sé que está registrado?

Asegúrate de escribir el nombre exactamente como aparece (respetando mayúsculas, tildes y espacios). Puedes verificar desde "**Visualizar productos**" antes de continuar.

12. ¿Puedo usar el sistema en otros sistemas operativos aparte de Windows?

Sí. MercaPlus funciona en **Windows, Linux y MacOS** siempre que tengas instalado un compilador de C++ compatible.

13. ¿Qué pasa si ingreso datos incorrectos?

El sistema tiene algunas validaciones básicas (por ejemplo, cantidades no negativas o mínimas mayores a 0). Si te equivocas, puedes usar la opción "**Cambiar Producto**" para corregir los datos ingresados.

14. ¿Se puede usar el sistema en red o en múltiples computadoras?

No. Esta versión es de uso **local**, pensada para ejecutarse en una sola computadora. No cuenta con funcionalidad en red ni sincronización entre múltiples dispositivos.

15. ¿Se puede imprimir el inventario o exportarlo?

Actualmente no. MercaPlus muestra la información por pantalla, pero no incluye funciones de exportación o impresión. Puedes copiar la salida desde la consola si lo necesitas.

X. Documentación técnica

a. Diagrama entidad relación:

Entidades principales:

1. Producto:

- nombre (string)
- tipo (string)
- cantidad (int)
- minimo (int)
- precioBase (float)
- precioFinal (float)
- vendidos (int)

2. Venta:

- nombreProducto (string)
- cantidadVendida (int)
- totalVenta (float)

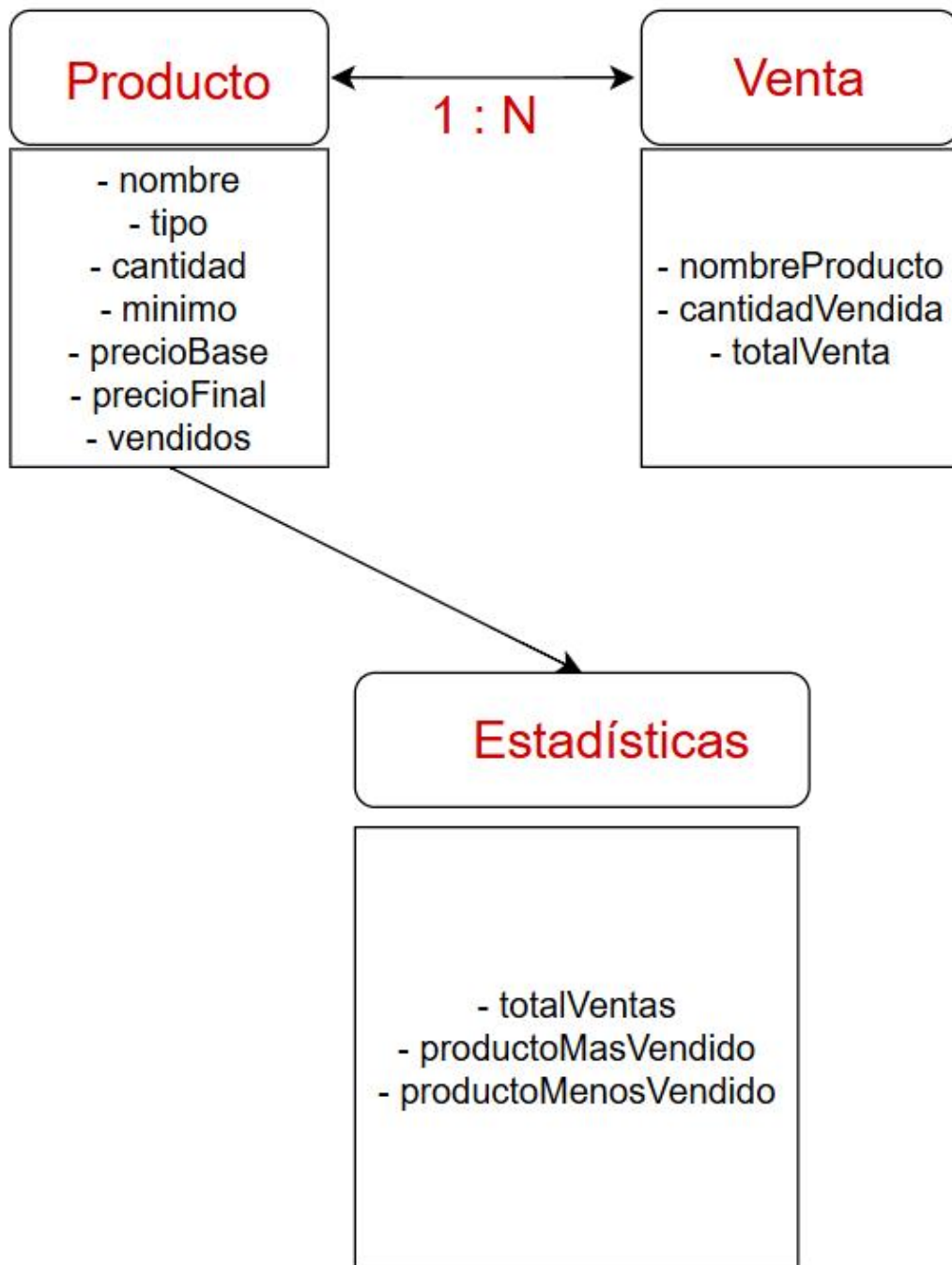
3. Estadísticas:

- totalVentas (float)
- productoMasVendido (string)
- productoMenosVendido (string)

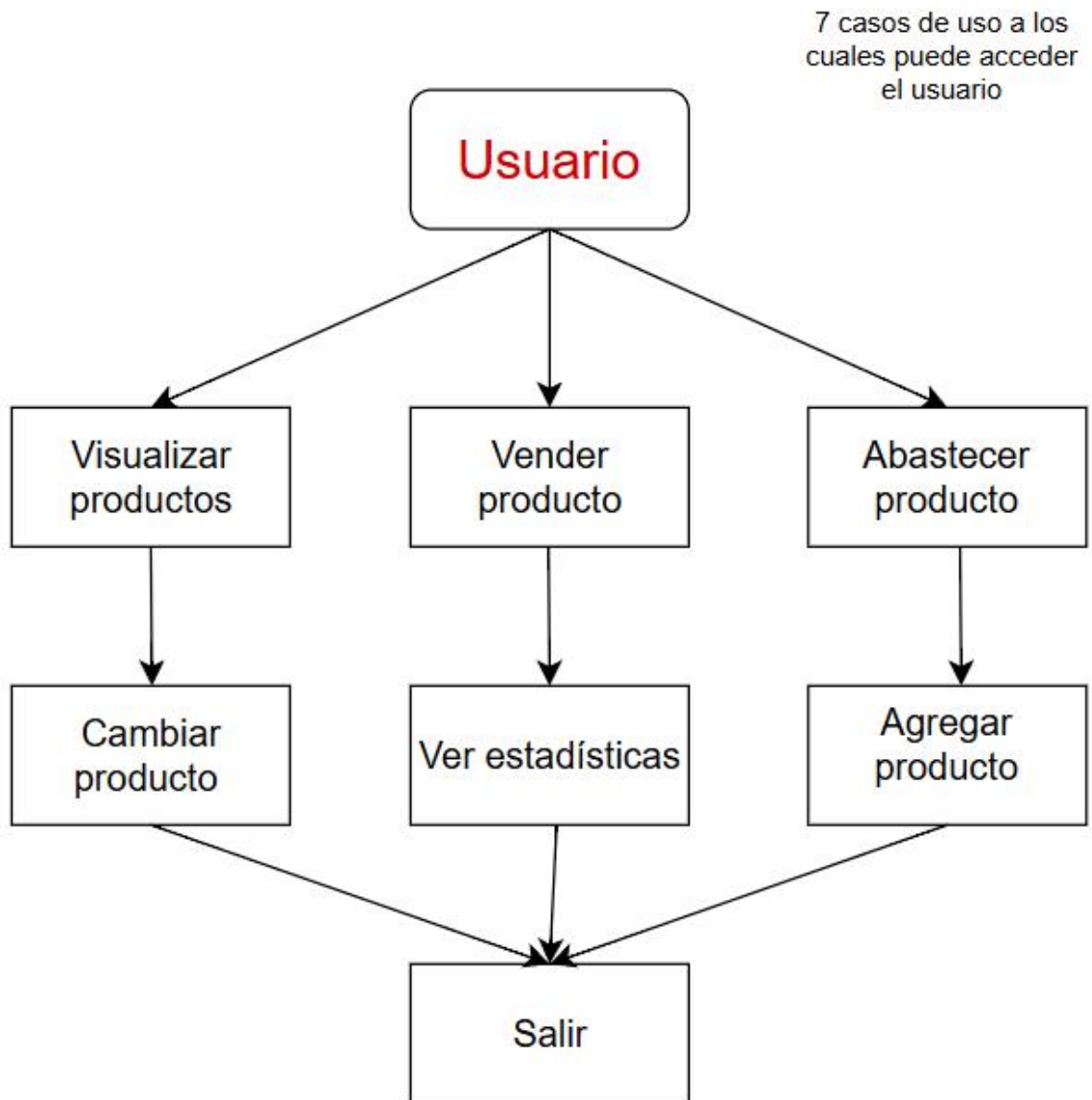
Relaciones:

- **Un Producto** puede generar **muchas Ventas** → 1:N
- **Estadísticas** se calcula **a partir de todos los Productos y Ventas** → la relación es dependiente de ambos.

Gráfico de entidad relación

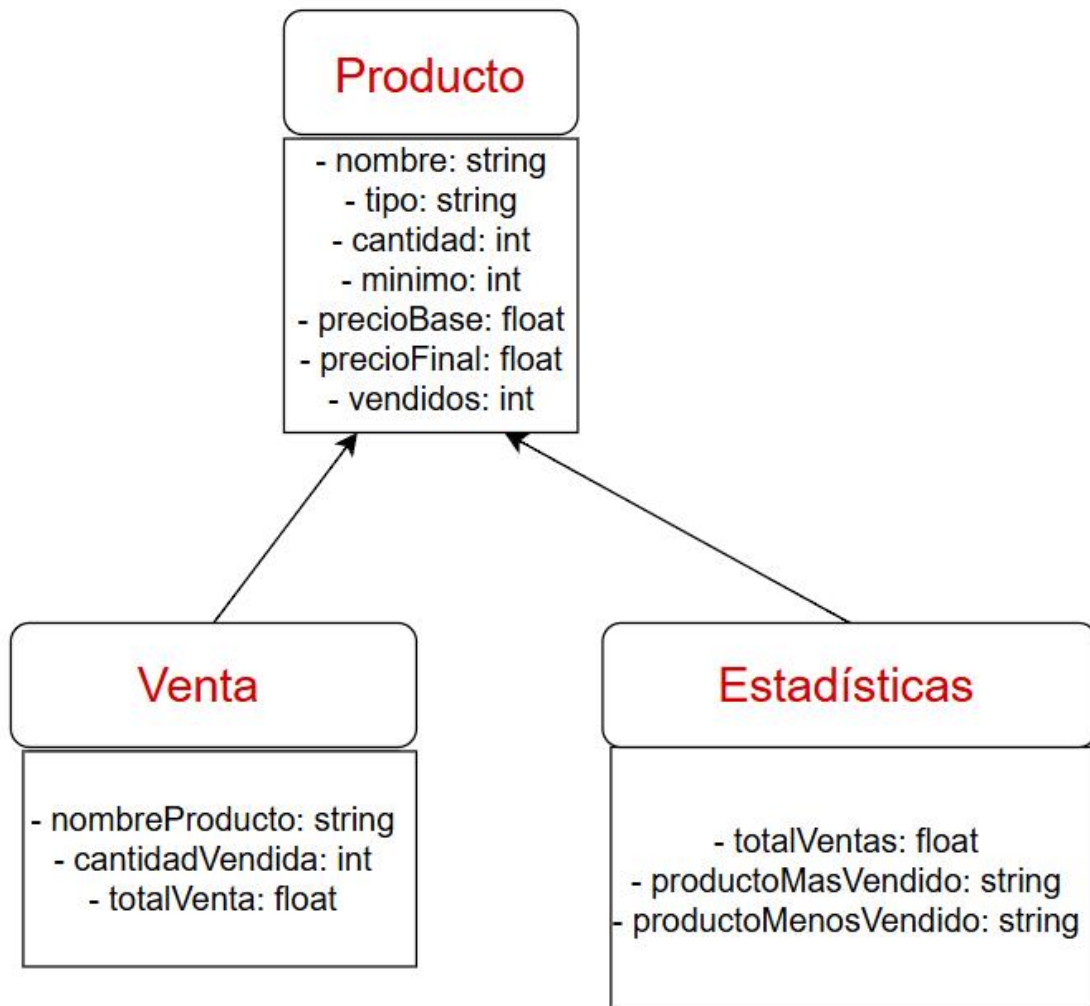


b. Diagrama de casos de uso:



c. Diagrama de clases:

- Producto y Venta están relacionados: cada venta está asociada a un producto (por nombre).
- Estadísticas depende del análisis de todos los Producto.



XI. Datos de contacto de los autores

Autores del proyecto:

Prográma: Ingeniería biomédica

- Shirley Mora

smora630@uan.edu.co

- Jennifer Rivera

jrivera507@uan.edu.co

- Juan José Martínez

Jmartinez351@uan.edu.co

Programa con la última versión generada

https://onlinegdb.com/RV4_r-X0Q

```
#include <iostream>
```

```
#include <vector>
```

```
#include <string> // Para manejar cadenas
```

```
#include <iomanip> // Para formatear la salida
```

```
#include <limits> // Para limpiar el buffer de entrada
```

```
#include <cstdlib> // Para system("cls")
```

```
using namespace std;
```

```
// Estructuras
```

```
struct Producto {
```

```
    string nombre;
```

```
    string tipo;
```

```
    int cantidad;
```

```
int minimo;

float precioBase;

float precioFinal;

int vendidos;

};

struct Venta {

    string nombreProducto;

    int cantidadVendida;

    float totalVenta;

};

struct Estadisticas {

    float totalVentas;

    string productoMasVendido;

    string productoMenosVendido;

};

// Productos

vector<Producto> inventario = {

    // Droguería

    {"Acetaminofen", "drogueria", 100, 20, 2500, 0.0, 0},

    {"Fenilefrina", "drogueria", 50, 10, 7000, 0.0, 0},

    {"Condomes", "drogueria", 60, 15, 7500, 0.0, 0},

    // Supermercado

    {"Arroz", "supermercado", 200, 50, 4200, 0.0, 0},

    {"Leche", "supermercado", 150, 30, 4000, 0.0, 0},

    {"Huevos", "supermercado", 180, 40, 8500, 0.0, 0},

    // Papelería

    {"Cuadernos", "papeleria", 80, 15, 3500, 0.0, 0},
```

```

        {"Esferos", "papeleria", 100, 25, 1500, 0.0, 0},

        {"Carpetas", "papeleria", 60, 10, 2000, 0.0, 0}

    };

vector<Venta> ventas;

Estadisticas estadisticas = {0.0, "", ""};

// Prototipos

void mostrarMenu();

void visualizarProductos();

void venderProducto();

void abastecerProducto();

void cambiarProducto();

void calcularEstadisticas();

void asignarDatosConIVA(Producto& producto);

void agregarProducto();

// Función para calcular el precio final con IVA

void asignarDatosConIVA(Producto& producto) {

    if (producto.tipo == "papeleria") {

        producto.precioFinal = producto.precioBase * 1.16;

    } else if (producto.tipo == "drogueria") {

        producto.precioFinal = producto.precioBase * 1.12;

    } else if (producto.tipo == "supermercado") {

        producto.precioFinal = producto.precioBase * 1.04;

    } else {

        producto.precioFinal = producto.precioBase;

    }

}

// Menú principal

```



```

void mostrarMenu() {

    cout << "\n--- MercaPlus ---" << endl;

    cout << "1. Visualizar Productos" << endl;

    cout << "2. Vender Producto" << endl;

    cout << "3. Abastecer Producto" << endl;

    cout << "4. Cambiar Producto" << endl;

    cout << "5. Calcular Estadisticas" << endl;

    cout << "6. Agregar Producto Nuevo" << endl;

    cout << "7. Salir" << endl;

    cout << "Seleccione una opcion: ";

}

// Visualizar productos

void visualizarProductos() {

    cout << left << setw(20) << "Nombre" << setw(15) << "Tipo"

        << setw(10) << "Cantidad" << setw(10) << "Minimo"

        << setw(16) << "Precio Base (COP)" << setw(18) << "Precio Final (COP)" << endl;

    cout << "-----" << endl;

    for (const auto& p : inventario) {

        cout << left << setw(20) << p.nombre << setw(15) << p.tipo << setw(10) << p.cantidad

            << setw(10) << p.minimo << setw(12) << fixed << setprecision(2) << p.precioBase

            << setw(12) << fixed << setprecision(2) << p.precioFinal << endl;

    }

}

// Vender producto

void venderProducto() {

    string nombreProducto;

    int cantidadAVender;

```

```

cout << "\n--- Vender Producto ---" << endl;

cout << "Nombre del producto: ";

cin.ignore(numeric_limits<streamsize>::max(), '\n');

getline(cin, nombreProducto);

do {

    cout << "Cantidad a vender: ";

    cin >> cantidadAVender;

    if (cantidadAVender < 0) {

        cout << "La cantidad no puede ser negativa.\n";

    }

} while (cantidadAVender < 0);

for (auto& p : inventario) {

    if (p.nombre == nombreProducto) {

        if (p.cantidad >= cantidadAVender) {

            float totalVenta = p.precioFinal * cantidadAVender;

            p.cantidad -= cantidadAVender;

            p.vendidos += cantidadAVender;

            estadisticas.totalVentas += totalVenta;

            ventas.push_back({nombreProducto, cantidadAVender, totalVenta});

            cout << "Venta realizada. Total: $" << fixed << setprecision(2) << totalVenta <<
endl;

            return;

        } else {

            cout << "No hay suficiente stock." << endl;

            return;

        }

    }

}

}

```

```

        cout << "Producto no encontrado." << endl;
    }

// Abastecer producto
void abastecerProducto() {
    string nombreProducto;

    int cantidadAgregar;

    cout << "\n--- Abastecer Producto ---" << endl;

    cout << "Nombre del producto: ";

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    getline(cin, nombreProducto);

    cout << "Cantidad a agregar: ";

    cin >> cantidadAgregar;

    for (auto& p : inventario) {
        if (p.nombre == nombreProducto) {
            p.cantidad += cantidadAgregar;

            cout << "Stock actualizado: " << p.cantidad << endl;

            return;
        }
    }

    cout << "Producto no encontrado." << endl;
}

// Cambiar producto
void cambiarProducto() {
    string nombreProducto;

    cout << "\n--- Cambiar Producto ---" << endl;

    cout << "Nombre del producto a cambiar: ";

```

```

cin.ignore(numeric_limits<streamsize>::max(), '\n');

getline(cin, nombreProducto);


for (auto& p : inventario) {

    if (p.nombre == nombreProducto) {

        cout << "Nuevo nombre: ";

        getline(cin, p.nombre);

        cout << "Nuevo tipo: ";

        getline(cin, p.tipo);

        cout << "Nueva cantidad minima: ";

        cin >> p.minimo;

        cout << "Nuevo precio base: ";

        cin >> p.precioBase;

        asignarDatosConIVA(p);

        cout << "Producto actualizado." << endl;

        return;

    }

}

cout << "Producto no encontrado." << endl;

}

// Calcular estadísticas

void calcularEstadisticas() {

    cout << "\n--- Estadisticas de Ventas ---" << endl;

    if (inventario.empty()) {

        cout << "No hay productos en el inventario." << endl;

        return;

    }

```

```

Producto* masVendido = &inventario[0];

Producto* menosVendido = &inventario[0];

int totalUnidadesVendidas = 0;


for (auto& p : inventario) {

    totalUnidadesVendidas += p.vendidos;

    if (p.vendidos > masVendido->vendidos) {

        masVendido = &p;

    }

    if (p.vendidos < menosVendido->vendidos) {

        menosVendido = &p;

    }

}

cout << "Producto mas vendido: " << masVendido->nombre << " (" << masVendido->vendidos << " unidades)" << endl;

cout << "Producto menos vendido: " << menosVendido->nombre << " (" << menosVendido->vendidos << " unidades)" << endl;

cout << "Total de ventas: $" << fixed << setprecision(2) << estadisticas.totalVentas << endl;

if (totalUnidadesVendidas > 0) {

    cout << "Promedio venta por unidad: $" << fixed << setprecision(2) << estadisticas.totalVentas / totalUnidadesVendidas << endl;

} else {

    cout << "No se han realizado ventas." << endl;

}

}

// Agregar producto

void agregarProducto() {

    Producto p;

```

```
cout << "\n=== Agregar Producto ===\n";

cout << "Nombre: ";

cin.ignore(numeric_limits<streamsize>::max(), '\n');

getline(cin, p.nombre);

for (const auto& prod : inventario) {

    if (prod.nombre == p.nombre) {

        cout << "Ese producto ya existe.\n";

        return;

    }

}

cout << "Tipo: ";

getline(cin, p.tipo);


cout << "Cantidad actual: ";

cin >> p.cantidad;

do {

    cout << "Cantidad mínima (al menos 1): ";

    cin >> p.minimo;

    if (p.minimo < 1) {

        cout << "Error: la cantidad mínima debe ser al menos 1.\n";

    }

} while (p.minimo < 1);


cout << "Precio base por unidad: ";

cin >> p.precioBase;

p.vendidos = 0;

asignarDatosConIVA(p);
```

```

    inventario.push_back(p);

    cout << "\nProducto agregado correctamente.\n";

    cout << "Precio con IVA: $" << p.precioFinal << endl;

}

// Función principal

int main() {

    int opcion;

    // Pantalla de bienvenida

    cout << "===== " << endl;

    cout << "      SISTEMA MERCAPLUS v1.0" << endl;

    cout << "-----" << endl;

    cout << " Proyecto desarrollado por:" << endl;

    cout << " - Shirley Mora" << endl;

    cout << " - Jennifer Rivera" << endl;

    cout << " - Juan José Martínez" << endl;

    cout << " Universidad Antonio Nariño - 2025" << endl;

    cout << "===== " << endl;

    cout << "\nPresione ENTER para continuar...";

    cin.ignore();

    cin.get();

    system("cls"); // Limpiar pantalla

    // Inicializar precios con IVA

    for (auto& producto : inventario) {

        asignarDatosConIVA(producto);

    }

    do {

        mostrarMenu();

```

```
cin >> opcion;

switch (opcion) {

    case 1:

        visualizarProductos();

        break;

    case 2:

        venderProducto();

        break;

    case 3:

        abastecerProducto();

        break;

    case 4:

        cambiarProducto();

        break;

    case 5:

        calcularEstadisticas();

        break;

    case 6:

        agregarProducto();

        break;

    case 7:

        cout << "Saliendo del programa." << endl;

        break;

    default:

        cout << "Opcion no valida. Intente de nuevo." << endl;

}

} while (opcion != 7);
```



```
    return 0;

}
```

Bibliografía:

- Villalobos, J. A. (s.f.). *Fundamentos de programación: Aprendizaje activo basado en casos*. Pearson Educación. Recuperado desde la Biblioteca Virtual Pearson de la Universidad Antonio Nariño.

- Código Facilito. (s.f.). *Curso de C++*.

<https://codigofacilito.com/cursos/c-plus-plus>

- Aprende.org - Fundación Carlos Slim. (s.f.). *Curso de Introducción a la Programación con C++*.

<https://aprende.org>

- Programación ATS. (s.f.). *Curso completo de C++ desde cero – Programación en Español*. Canal de YouTube.

<https://www.youtube.com/@ProgramacionATS>

- Programación en C++ desde cero. (2023). *Estructuras, funciones y ejemplos prácticos*. <https://www.tutorialesprogramacionya.com/cppya/>

- Comunidad Stack Overflow en español. (s.f.). *Ejemplos y solución de errores comunes en C++*.

<https://es.stackoverflow.com/>

- YouTube – Píldoras Informáticas. (s.f.). *Curso completo de C++ en español*. <https://www.youtube.com/playlist?list=PLU8oAlHdN5BmpIQGDSHo5e1r4ZYWQ8m4B>

- ChatGPT (2024). *Asistencia para desarrollo, validación, depuración y explicación del código fuente en C++*, a través de la plataforma ChatGPT de OpenAI.