



Proyecto de Desarrollo del Sistema de Ventas para una Ferretería

Nombre: Juan Carlos Estrada Jiménez

Universidad Politécnica Internacional

Materia: Técnicas de Programación

Luis Felipe Mora Umaña

7/9/2024

Contenido

- Introducción..... 3
 - Objetivos 3
 - Requisitos del Sistema..... 3
 - Gestión de Inventarios..... 3
 - Gestión de Ventas:..... 3
 - Gestión de Clientes:..... 3
 - Reportes y Análisis:..... 4
 - Estructura del Proyecto 4
 - Gestión de Inventarios: 4
 - Gestión de Ventas:..... 5
 - Gestión de Clientes:..... 5
 - Reportes y Análisis:..... 5
- Conclusiones 6

Introducción

En este proyecto, se va a desarrollar un sistema de ventas para una ferretería utilizando el lenguaje C# y aplicando los principios de POO, Clean Code y el patrón MVC. El sistema permitirá gestionar inventarios, procesar ventas, mantener información de clientes y generar reportes.

Objetivos

- Diseñar e implementar un sistema eficiente y organizado.
- Utilizar el patrón Modelo-Vista-Controlador para separar las responsabilidades.
- Cargar datos básicos del sistema desde archivos CSV.
- Generar reportes mediante pantallas.

Requisitos del Sistema

Gestión de Inventarios

- Agregar, editar y eliminar productos: Se implementarán funciones para agregar nuevos productos al inventario, editar detalles existentes (como precio o descripción) y eliminar productos obsoletos.
- Controlar las unidades disponibles: Aquí se llevará un registro de las unidades disponibles para cada producto. Actualiza automáticamente las cantidades después de cada venta o reposición.
- Notificar productos con bajo inventario: Dará alertas para productos que tengan un inventario por debajo de lo seleccionado por la persona.

Gestión de Ventas:

- Procesar ventas.
- Generar facturas y recibos.
- Registrar métodos de pago (efectivo, tarjeta, transferencia bancaria).

Gestión de Clientes:

- Registrar y mantener información de clientes: Crea formularios para ingresar y mantener datos de los clientes, como nombre, dirección y número de teléfono.

- Mantener historial de compras: Asocia las compras realizadas por cada cliente para que puedas rastrear su historial.

Reportes y Análisis:

- Generar reportes de ventas: Genera reportes de ventas diarias, estos informes incluirán también los ingresos y productos.
- Mostrar inventarios. Proporciona una vista general del inventario actual, incluyendo detalles como cantidad disponible y precio.
- Identificar vendedores destacados y clientes relevantes: Analiza los datos para identificar patrones, como los vendedores más productivos o los clientes más frecuentes.

Estructura del Proyecto

Gestión de Inventarios:

1. Agregar productos al inventario.

```
static void AgregarProducto()
{
    Console.WriteLine("==== Agregar Producto =====");
    Console.Write("Nombre del producto: ");
    string nombre = Console.ReadLine();
    Console.Write("Cantidad disponible: ");
    int cantidad = int.Parse(Console.ReadLine());
    Console.Write("Precio: ");
    decimal precio = decimal.Parse(Console.ReadLine());

    Producto nuevoProducto = new Producto
    {
        Id = productoIdCounter++,
        Nombre = nombre,
        CantidadDisponible = cantidad,
        Precio = precio
    };

    inventario.Add(nuevoProducto);
    Console.WriteLine("Producto agregado correctamente.");
}
```

2. Editar productos existentes.

```
static void EditarProducto()
{
    Console.WriteLine("==== Editar Producto =====");
    MostrarInventario();

    Console.Write("Seleccione el ID del producto a editar: ");
    int idProducto = int.Parse(Console.ReadLine());

    Producto productoEditar = inventario.Find(p => p.Id == idProducto);

    if (productoEditar != null)
    {
        Console.Write("Nuevo nombre del producto: ");
        productoEditar.Nombre = Console.ReadLine();

        Console.Write("Nueva cantidad disponible: ");
        productoEditar.CantidadDisponible = int.Parse(Console.ReadLine());

        Console.Write("Nuevo precio: ");
        productoEditar.Precio = decimal.Parse(Console.ReadLine());

        Console.WriteLine("Producto editado correctamente.");
    }
    else
    {
        Console.WriteLine("Producto no encontrado.");
    }
}
```

3. Eliminar productos.

```
1 referencia
2 static void EliminarProducto()
3 {
4     Console.WriteLine("==== Eliminar Producto ====");
5     MostrarInventario();
6
7     Console.WriteLine("Seleccione el ID del producto a eliminar: ");
8     int idProducto = int.Parse(Console.ReadLine());
9
10    Producto productoEliminar = inventario.Find(p => p.Id == idProducto);
11
12    if (productoEliminar != null)
13    {
14        inventario.Remove(productoEliminar);
15        Console.WriteLine("Producto eliminado correctamente.");
16    }
17    else
18    {
19        Console.WriteLine("Producto no encontrado.");
20    }
21 }
```

Gestión de Ventas:

1. Procesar ventas.
2. Generar facturas y recibos.
3. Registrar métodos de pago.

Gestión de Clientes:

1. Registrar información de clientes.
2. Mantener historial de compras.

Reportes y Análisis:

1. Generar reportes.
2. Identificar vendedores y clientes relevantes.

Conclusiones

En este proyecto, he abordado el desarrollo de un sistema de ventas para una ferretería utilizando el lenguaje C# y aplicando principios fundamentales como la programación orientada a objetos (POO), el mantenimiento de un código limpio y el uso del patrón Modelo-Vista-Controlador (MVC). A través de la implementación de las funcionalidades de gestión de inventarios, procesamiento de ventas y seguimiento de clientes, hemos sentado las bases para futuras iteraciones.

La separación de responsabilidades mediante el patrón MVC nos permite mantener un sistema organizado y escalable. Además, la carga de datos desde archivos CSV y la generación de reportes en pantalla contribuyen a la eficiencia y utilidad del sistema.

En resumen, este proyecto no solo cumple con los objetivos iniciales, sino que también establece una estructura sólida para futuras mejoras y optimizaciones. Siguiendo buenas prácticas y manteniendo un código limpio, estamos en el camino correcto para lograr un sistema robusto y eficiente.