Universidad Politécnica Internacional

Curso: 2025.IIIC.G3.Tecnicas de Programación

Profesor: Luis Felipe Mora Umaña

Proyecto: Proyecto #1

Integrantes:

Gerald Elizondo Mora

Allison Solano Solera

III Cuatrimestre 2025

Índice

Introducción 3
Decisiones de Diseño 4
Uso del Programa 5
Aprendizajes 8
Conclusión 9

Introducción

El presente proyecto tiene como objetivo desarrollar la primera versión de un sistema de gestión para compras remotas en las distintas ferias del agricultor. La idea central es ofrecer una plataforma que permita a los usuarios realizar sus compras de forma sencilla y ordenada, integrando funciones como registro de usuarios, manejo de carrito de compra, administración de productos de diversos proveedores y generación de estadísticas básicas.

En esta etapa inicial, el proyecto se enfoca en implementar funciones como:

La gestión de usuarios, que permite el registro de compradores y la existencia de productores para generar estadísticas.

Un carrito de compra, donde cada usuario puede seleccionar productos de distintos proveedores dentro de la feria correspondiente.

Inventarios, asegurando que las cantidades de productos se actualicen de manera correcta al realizar compras.

Facturación, facilitando la generación de comprobantes y la selección del lugar de entrega.

Productos y proveedores, organizando la información de forma intuitiva para el usuario.

Estadística, como resumen de compras, proveedores más frecuentes, meses de mayor consumo y productos más adquiridos.

Para construir este proyecto se emplean los principios de la programación orientada a objetos, junto con buenas prácticas como Clean Code, el patrón Modelo-Vista-Controlador (MVC) y los lineamientos SOLID, todo implementado en C# utilizando Windows Forms. Esto garantiza que el sistema sea extensible, fácil de mantener y preparado para extenderlo, especialmente para el Proyecto #2.

En esta primera versión, el sistema trabaja mediante la carga de archivos JSON, donde se almacena la información de usuarios, proveedores, productos y registros de compras. A partir de estos datos, el sistema debe permitir la creación de carritos personalizados, gestionar inventario, generar facturas, mostrar productos y ofrecer estadísticas útiles como hábitos de compra, meses con mayor consumo y productos más adquiridos.

Decisiones de Diseño

Para esta primera versión del sistema se tomaron varias decisiones de diseño con el fin de que el proyecto fuera fácil de extender, mantener y comprender:

1. **Uso del patrón MVC (Modelo–Vista–Controlador)**
   Se eligió MVC para separar la lógica del negocio, la interfaz gráfica y el control del flujo del sistema. Esto permite modificar cualquier capa sin afectar directamente a las demás, haciendo el proyecto más flexible.

2. **Arquitectura orientada a objetos**
   Las entidades principales como *Usuario*, *Proveedor*, *Producto*, *Carrito*, *Factura* y *Feria* se modelaron como clases independientes. Esto facilita reutilizar código, mantener orden y aplicar principios SOLID.

3. **Aplicación de Clean Code y SOLID**
   Se optó por nombres descriptivos, métodos cortos y responsabilidades únicas. Estas decisiones buscan que el código sea legible y que pueda escalarse sin convertirse en un sistema difícil de gestionar.

4. **Carga de datos desde archivos JSON**
   Para esta versión se decidió utilizar archivos externos como base de datos. Esto permite trabajar con grandes cantidades de registros sin necesidad de implementar una base de datos completa todavía.

5. **Control de inventario en tiempo real**
   Se diseñó el sistema para que cada compra afecte el inventario del proveedor correspondiente. Esto permite reflejar un comportamiento más cercano al de un sistema real sin añadir demasiada complejidad.

6. **Interfaz basada en Windows Forms**
   Se eligió Windows Forms por su facilidad para construir pantallas intuitivas y porque es adecuado para proyectos académicos donde se prioriza la funcionalidad y la rapidez de implementación.

Uso del Programa

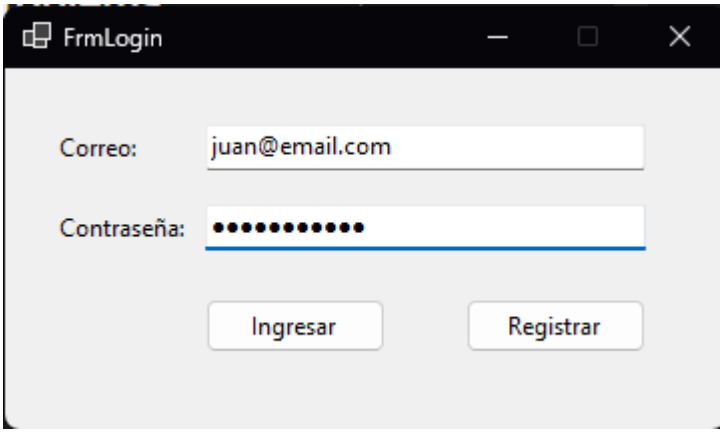
Pantalla de Inicio: Login

Al iniciar, se muestra el formulario

Permite ingresar email, ingresar contraseña,

acceder al sistema, registrar un nuevo usuario

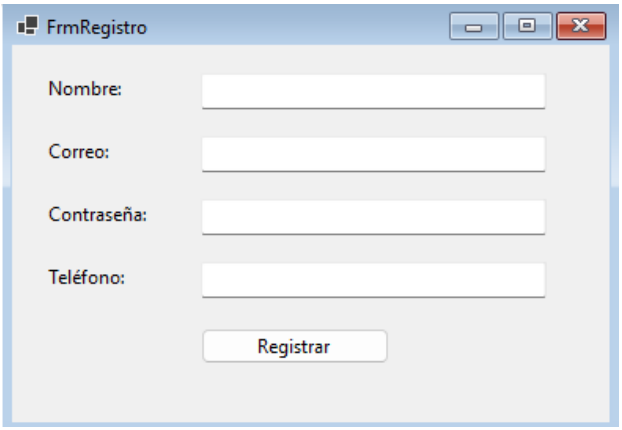
Si el usuario no existe, debe presionar “Registrar”.



Registro de Usuario

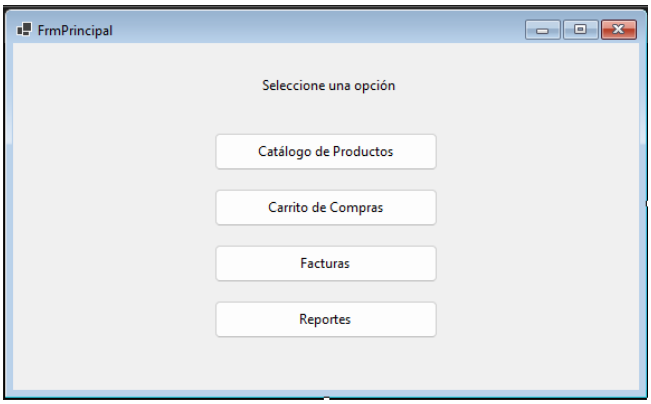
Solicita: Nombre, email, contraseña y teléfono.

Después de registrarse, el usuario puede iniciar sesión.



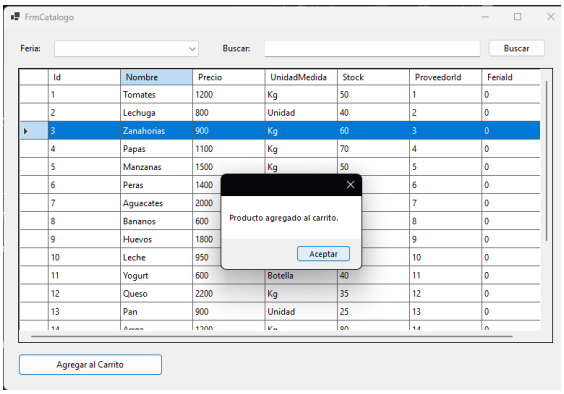
Pantalla Principal

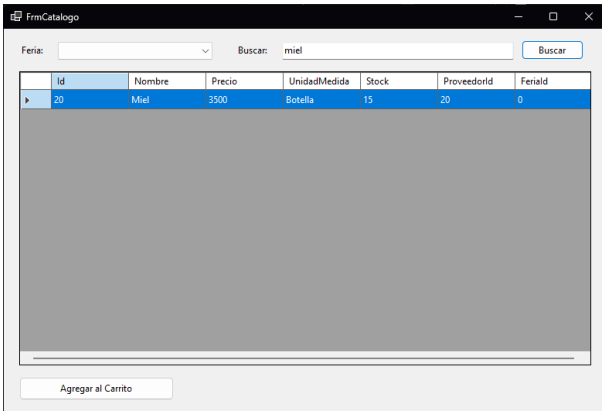
Contiene las opciones: Catálogo, carrito de compras, facturas y reportes.



Catálogo de Productos

Funciones: Seleccionar la feria, mediante un ComboBox se elige una feria disponible, ver productos según la feria seleccionada, los productos se filtran automáticamente, buscar por nombre, filtro por texto, agregar productos al carrito, el usuario selecciona un producto y presiona Agregar.





Carrito de Compras

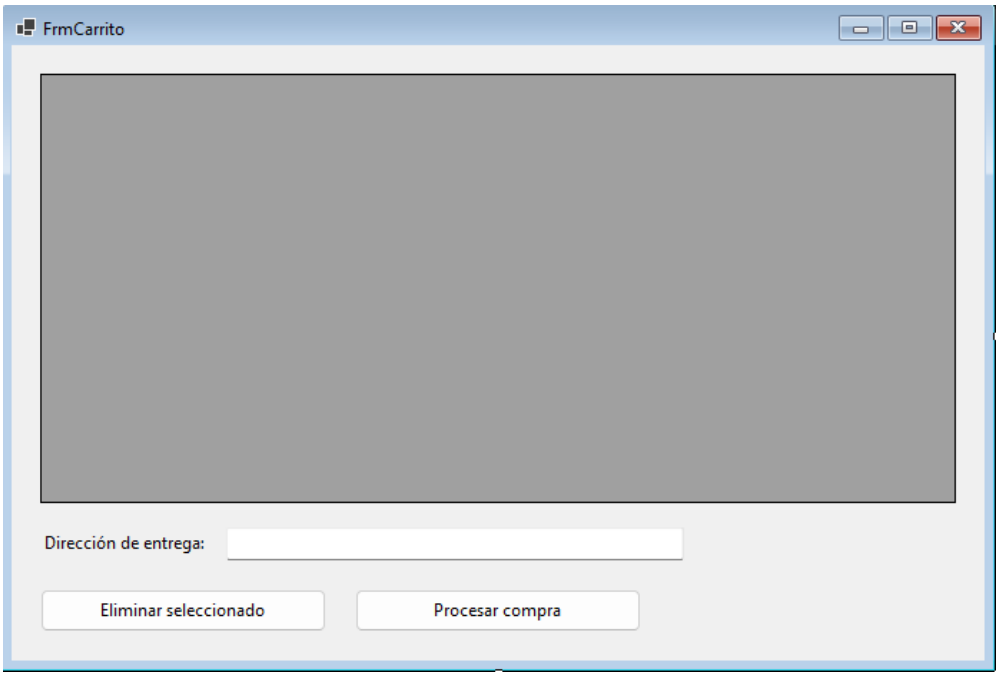
Muestra los productos agregados:

Nombre del producto, cantidad y total por línea.

Permite eliminar productos, ingresar dirección de entrega y procesar la compra.

Al procesar la compra:

Se completa la orden, se genera una factura y se registra en Facturas.json



Facturas y Reportes

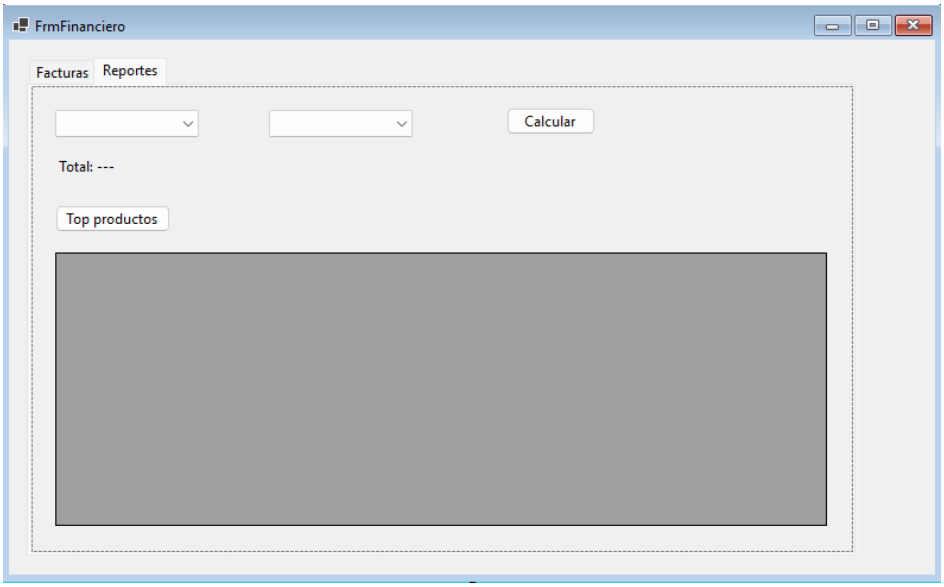
Permite:

Ver todas las facturas del usuario

Código de factura, fecha, total y dirección.

Ver reportes

Total de ventas por mes y productos más vendidos



Aprendizajes

Durante el desarrollo de esta etapa del proyecto se obtuvieron varios aprendizajes importantes.

La importancia de planificar la estructura del sistema desde el inicio para evitar retrabajo, cómo separar responsabilidades dentro de una aplicación mediante MVC y POO, la relevancia de utilizar nombres claros y mantener métodos pequeños para mejorar la lectura del código, manejo básico de archivos CSV/JSON para cargar grandes cantidades de información, cómo organizar entidades, controladores e interfaces para hacer el sistema escalable, la utilidad de aplicar principios SOLID para evitar dependencias entre las clases y el valor de documentar el código y manejar excepciones para evitar errores al momento de la ejecución.

Conclusión

Al utilizar principios de Clean Code, MVC y POO, el proyecto se construyó de manera organizada, facilitando la comprensión del código e ir preparando el sistema para los cambios que se incorporarán en la segunda iteración.

Además, la implementación de funciones como gestión de usuarios, manejo de carrito de compra, inventario, estadísticas básicas y carga de datos desde archivos permitió simular el funcionamiento real de una plataforma de compras para una feria del agricultor. Este trabajo evidencia la importancia de una buena arquitectura y de aplicar buenas prácticas desde el inicio para evitar problemas de mantenimiento y garantizar un crecimiento ordenado del proyecto.