

Instituto Técnico Ricaldone

Primer Año de Desarrollo de Software



“Sistema de gestión de inventario para una Floristería”

Especialidad: Desarrollo de Software 1° Año

Sección: B-2

Docente: Emerson Alexander González Rodríguez

Integrantes:

Allan Adalberto Vásquez Buendía

Rocio Elizabeth Alfaro Duran

José Orlando Zavala López

Anthony Isaac Galdamez Mendoza

Índice

1: Introducción	3
2: Tecnologías y herramientas.....	4
C#	4
Windows Forms	4
SQL Server	4
Canva.....	4
3: Estructura de la base de datos	5
• Diagrama Entidad relación	5
• Diagrama Caso de uso.....	5
• Diagrama de Clases.....	5
• Diagrama de Modelo de dominio.....	5
4: Diccionario de datos	6
5: Arquitectura de software.....	6
6: Estructura del proyecto	9
Estructura del backend.....	9
Estructura del sistema entidades.....	10
Estructura de la capa sistema datos.....	11
Estructura de la capa sistema negocio	12
Estructura del frontend	12
Estructura de la capa sistema presentación	13
○ Formularios principales:	14
○ Modales (ventanas emergentes):	14
○ Recursos gráficos:	15
Estructura de la carpeta Utilidades.....	16
7: Diseño de la aplicación	16
9: Requerimientos de hardware y software	17
9.1 Localmente	17
10. Instalación y configuración.....	19

1: Introducción

El manual técnico ha sido elaborado con el propósito de guiar al usuario en el uso y comprensión de un sistema de gestión de inventario diseñado específicamente para una floristería de flores artificiales. Esta aplicación, desarrollada en **C#** y respaldada por **SQL Server** como motor de base de datos, ofrece una solución práctica y escalable para organizar, controlar y optimizar el stock de productos en un entorno comercial cada vez más dificultoso.

El gestor de inventario no solo permite registrar y actualizar entradas y salidas de mercancía, sino que también facilita la clasificación de los productos por categorías, la consulta de existencias en tiempo real y la generación de reportes personalizados para la toma de decisiones.

En definitiva, lo que aquí encontrará no es solo un instructivo, sino una guía que busca convertir la gestión de inventario en un proceso ordenado, confiable y eficiente para pequeñas y medianas empresas facilitando su gestión.

2: Tecnologías y herramientas

C#

Lenguaje de programación usado para desarrollar la aplicación del gestor de inventario. Permite crear una solución estable y fácil de usar.

Windows Forms

Tecnología usada para construir la interfaz gráfica de usuario, ofreciendo pantallas simples y prácticas para la gestión del sistema.

SQL Server

Base de datos donde se almacena y organiza toda la información de productos, existencias y movimientos de inventario.

Canva

Herramienta utilizada para diseñar diagramas y recursos visuales que apoyan la documentación y la presentación del sistema.

3: Estructura de la base de datos

- **Diagrama Entidad relación**

https://lucid.app/lucidchart/56b28c90-8489-47d4-81db-c46bec4726d1/edit?page=0_0&invitationId=inv_9fb57936-cc34-4645-8bc7-6d2086afdc14#

- **Diagrama Caso de uso**

https://lucid.app/lucidchart/0d3725a6-9dd4-4526-8dbc-6207311dc581/edit?page=0_0&invitationId=inv_d670adee-ee73-4b94-b72b-03b4c9e373ee#

- **Diagrama de Clases**

https://lucid.app/lucidchart/92d0463d-4390-41cb-a567-3084522c87fb/edit?page=0_0&invitationId=inv_8dc79f45-d5e3-4938-b9bd-91abd5421739#

- **Diagrama de Modelo de dominio**

https://lucid.app/lucidchart/79ecf0f2-9652-4b1a-ba8e-1ac66c7fc52f/edit?page=0_0&invitationId=inv_2d867285-6d32-4fab-aa4b-83ffd9f7a11e#

4: Diccionario de datos

x

https://docs.google.com/spreadsheets/d/1CMbVedEqGmTwXG3mYx80ORRSovVEcDW_oInZOvghCl8/edit?gid=0#gid=0

5: Arquitectura de software

Aplicaciones en los diferentes elementos

En nuestra gestión de inventario para una floristería la paleta que usamos basada en los colores rosas claro hasta rosa fuerte se aplican de manera estratégica en los distintos elementos de la plataforma para garantizar una experiencia para el usuario coherente y agradable. a continuación se muestran algunas imágenes para ver cómo se emplea estas paletas en los diferentes componentes Imágenes de nuestra aplicación

Sistema de inventario

Nro Documento

Contraseña

Ingresar

Cancelar

Inicio

Sistema de Inventario

Usuario: IblUsuarios

Usuario

Productos

Ventas

Comprar

Clientes

Proveedores

Reportes

Detalle del usuario

Nro Documento: -1 0

Nombre completo:

Correo:

Contraseña:

Confirmar Contraseña:

ROL:

Estado:

Guardar

Limpiar

Eliminar

Lista de usuarios

Buscar por:

	NumeroDocumento	Nombre Completo	Correo	Rol	Estado
--	-----------------	-----------------	--------	-----	--------

Detalle del Producto

Codigo: -1 0

Nombre:

Descripción:

Categoría:

Estado:

Guardar

Limpiar

Eliminar

Lista de productos

Buscar por:

	Codigo	Nombre	Descripción	Categoría	Stock	Precio Compra	Precio Venta
--	--------	--------	-------------	-----------	-------	---------------	--------------

Registrar Venta

Información Venta

Fecha:

Tipo Documento:

Información Cliente

Número Documento:

Nombre Completo:

Información de Producto

Cod. Producto:

Producto:

Precio:

Stock:

Cantidad:

1

Agregar

Producto	Precio	Cantidad	Sub Total
----------	--------	----------	-----------

Total a Pagar:

0

Paga con:

Cambio:

Crear Venta

Registrar Compra

Informacion de compra

Fecha

Tipo de documento

Informacion Proveedor

Numero de documento

Tipo de documento

0

Informacion de Productos

Cod. Producto:

0

Producto:

Precio Compra:

Precio Venta:

Cantidad:

0

	Producto	Precio Compra	Cantidad	Sub Total	
*					

Total a Pagar:

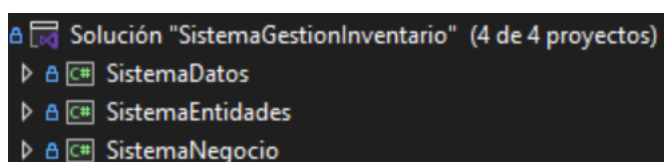
0

Registrar

6: Estructura del proyecto

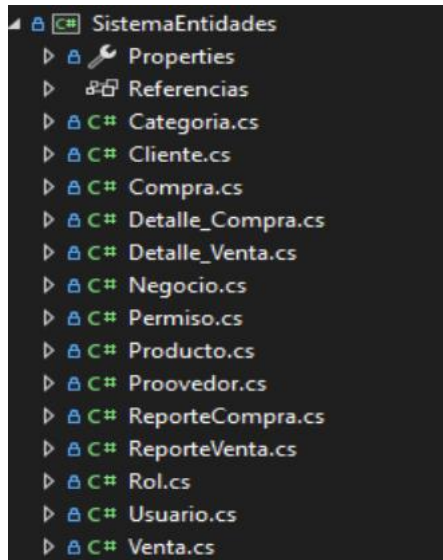
El proyecto se divide en backend y frontend, tomando en cuenta 4 capas de diferentes funciones que intervienen en la ejecución del sistema. Estas capas permiten separar responsabilidades, organizar el código y asegurar que los procesos relacionados con la base de datos, la lógica de negocio y la interfaz gráfica se realicen de manera eficiente, mostraremos las estructuras del backend y frontend:

Estructura del backend



El backend de este sistema de gestión de inventario está conformado por tres capas principales: SistemaEntidades, SistemaDatos y SistemaNegocio. Cada una desempeña un rol esencial en el procesamiento y manejo de los datos.

Estructura del sistema entidades

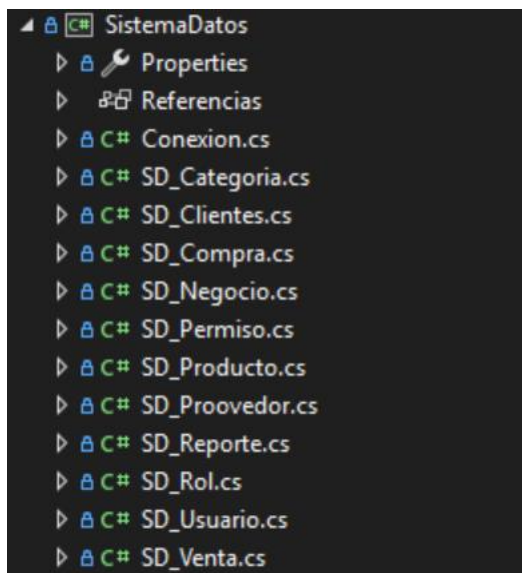


Cada clase define las propiedades que corresponden a los campos de la tabla en SQL Server. De esta manera, la capa de entidades permite que los objetos del sistema puedan manipular la información de una forma estructurada y organizada.

- Ejemplo: Cliente.cs contiene atributos como *Nombre*, *Documento*, *Correo*, *Teléfono*.
- Producto.cs contiene atributos como *IdProducto*, *Nombre*, *Categoría*, *PrecioCompra*, *PrecioVenta*, *Stock*.
- Venta.cs y Compra.cs definen los detalles de las transacciones realizadas.
- ReporteCompra.cs y ReporteVenta.cs se utilizan para la representación de datos específicos de reportes tanto en ventas, como en compras.

Aclarando que esta capa cumple la función de asociar las tablas de la base de datos a clases en C#, de forma que puedan ser reutilizadas en las demás capas del sistema.

Estructura de la capa sistema datos

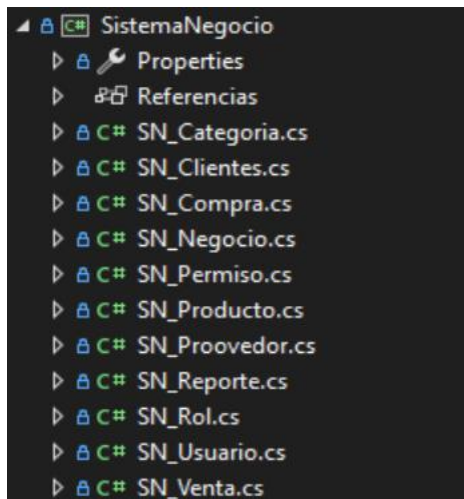


La capa de SistemaDatos contiene las clases responsables de la comunicación directa con la base de datos SQL Server. Cada clase tiene una cierta asociación de una tabla específica, y su objetivo es ejecutar operaciones de tipo CRUD (crear, leer, actualizar y eliminar).

- Conexion.cs: Establece la cadena de conexión con la base de datos, asegurando que las demás clases puedan interactuar con SQL Server.
- Clases como SD_Producto.cs, SD_Cliente.cs, SD_Venta.cs, SD_Compra.cs: ejecutan las consultas SQL necesarias para la gestión de productos, clientes, ventas y compras.
- SD_Reporte.cs: maneja consultas específicas para la obtención de datos que posteriormente se utilizan en la generación de reportes en ventas y compras.

Cada clase en esta capa actúa como un intermediario entre las entidades del sistema y la base de datos real. Su función principal es garantizar que los datos sean correctamente almacenados, actualizados y recuperados, transfiriendo la lógica de negocio a la capa correspondiente.

Estructura de la capa sistema negocio

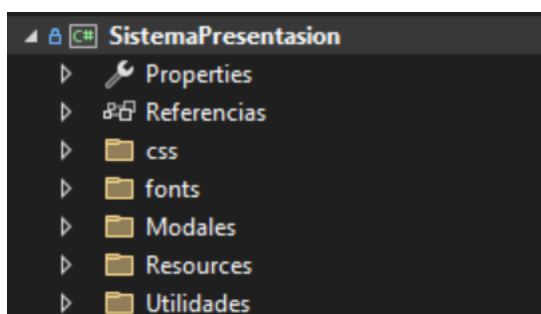


La capa de SistemaNegocio contiene las reglas que restringen el funcionamiento del sistema. Aquí se definen los procesos que deben cumplirse antes de enviar o recibir información de la base de datos.

- Clases como SN_Producto.cs, SN_Usuario.cs, SN_Venta.cs contienen métodos que validan la información antes de guardarla en la base de datos.
- Manejan procesos como: cálculo de totales de ventas, validación de stock disponible, asignación de permisos de usuario, entre otros.
- Actúan como el puente que conecta la capa de presentación (formularios e interfaz gráfica) con la capa de SistemaDatos.

Esta capa asegura que toda la información que entra y sale del sistema cumpla con las normas que establecimos. De esta forma, el backend no solo guarda y recupera datos, sino que también procesa, valida y controla el flujo de la información.

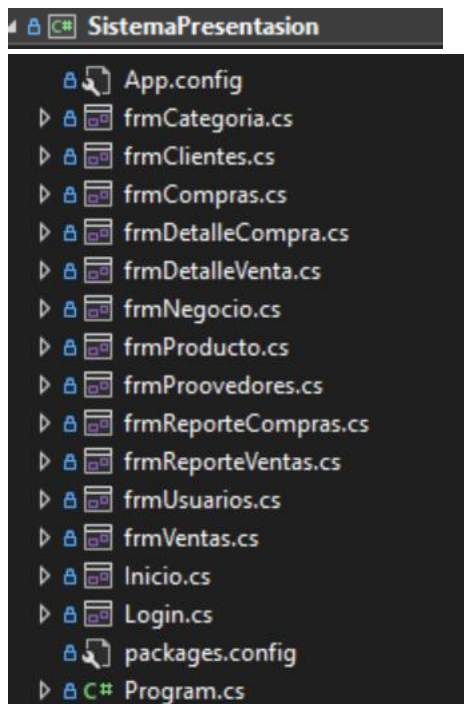
Estructura del frontend



El frontend del sistema corresponde a la capa de SistemaPresentación, que se desarrolla mediante Windows Forms en C#. Es el espacio visual con el que

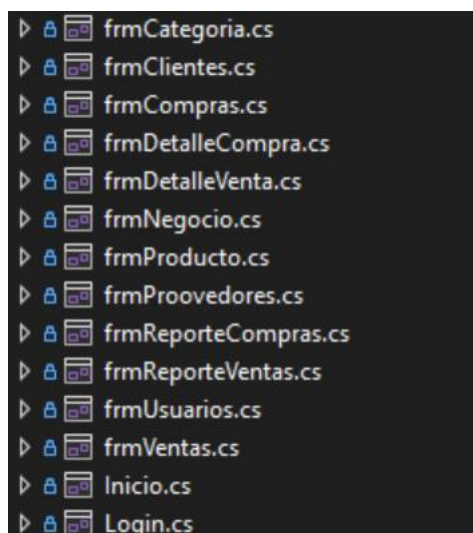
interactúa el usuario, y donde se despliegan los resultados provenientes de las capas del backend.


Estructura de la capa sistema presentación



Esta biblioteca contiene todos los formularios y recursos que conforman la interfaz gráfica. Aquí se agrupan los formularios principales, los formularios emergentes (modales), los recursos gráficos como imágenes e íconos, además de los estilos utilizados en la aplicación.

- **Formularios principales:**

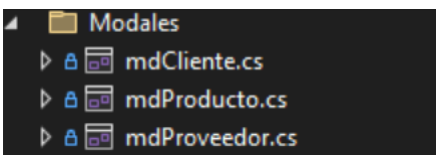




App.config

- frmClientes.cs, frmProductos.cs, frmVentas.cs, frmCompras.cs: permiten el manejo de las operaciones más importantes del sistema.
- frmUsuarios.cs, frmRoles.cs, frmPermisos.cs: se utilizan para administrar la seguridad y los diferentes niveles de acceso al sistema.
- frmReportes.cs: permite visualizar y exportar información organizada en reportes.
- Login.cs y Inicio.cs: el controla el ingreso al sistema y el otro la navegación inicial.
- App.config: archivo de configuración que almacena parámetros necesarios para la ejecución del sistema.

- **Modales (ventanas emergentes):**

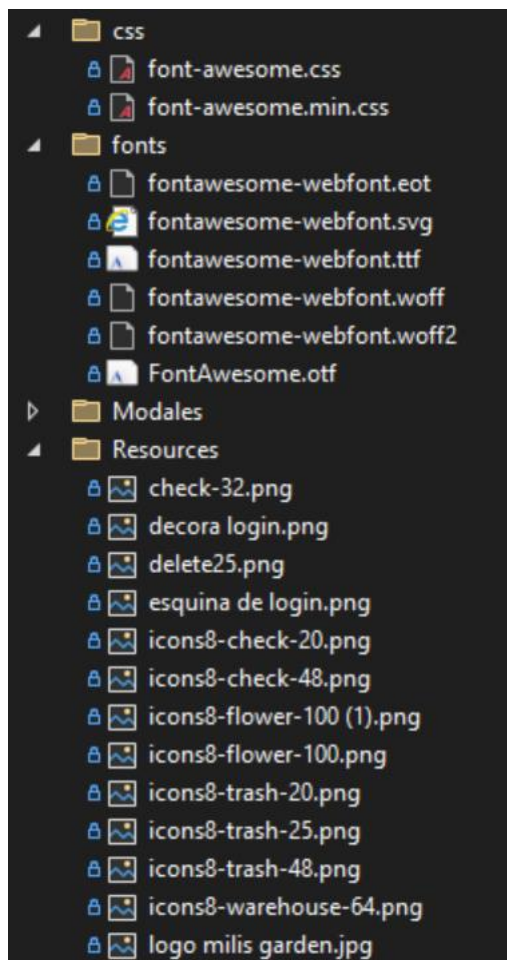


Modales

- mdCliente.cs
- mdProducto.cs
- mdProveedor.cs

- mdCliente.cs, mdProducto.cs, mdProveedor.cs: facilitan operaciones rápidas de alguna gestión específica dentro del sistema, sin necesidad de acceder a las ventanas principales.

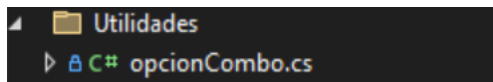
- **Recursos gráficos:**



- Carpeta Resources que almacena íconos e imágenes (ejemplo: íconos de productos, reportes, botones).
- Carpeta **css** y **fonts** que incluyen estilos y tipografías adicionales para mejorar la apariencia visual.

En conclusión, esta capa representa la parte visual y de interacción del sistema, brindando al usuario una interfaz amigable para realizar todas las operaciones del inventario.

Estructura de la carpeta Utilidades



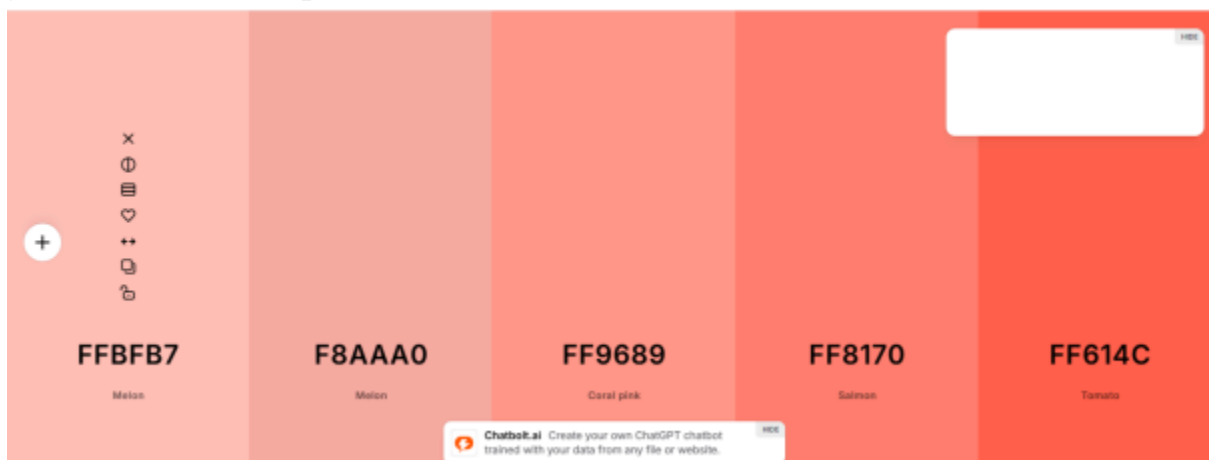
La carpeta de utilidades contiene elementos de apoyo para la capa de presentación, que facilitan la implementación de ciertos procesos en los formularios.

- opcionCombo.cs: permite cargar y gestionar de manera eficiente los combos desplegables que se utilizan en varios formularios del sistema.

Con esta opción, las utilidades funcionan como herramientas complementarias para mejorar las interacciones entre el usuario y la aplicación.

7: Diseño de la aplicación

7 paletas de colores En nuestro proyecto de gestión de inventario hemos usado diferentes tipos de paletas de colores en base de tonalidades de rosado fuertes y claros para mantener una estética coherente y profesional, alineada con la identidad visual de nuestro proyecto. estas paletas de colores fueron seleccionadas cuidadosamente para asegurarnos una experiencia visual agradable y moderna Primera paleta



Tipo de tamaños de fuentes el tipo de fuente que usamos para el proyecto es new roman el cual se emplea con un tamaño estándar de 12. tipos y dimensión de imágenes utilizadas Los tipos de imágenes utilizadas en nuestro sistema son PNG

y JPG. los cuales varían de dimensiones teniendo desde los 18 X 18PX los cuales varían de dimensiones teniendo desde imagen 1 imagen 2 tamaños de pantalla soportados en nuestro proyecto de gestión de inventario no hemos utilizado el soporte bootstrap así que los tamaños que tenemos aseguran a que el sistema de inventario sea adaptado adecuadamente a cualquier resolución, proporcionando una experiencia de usuario óptima en dispositivos móviles, tablets, laptops, y pantallas grandes

9: Requerimientos de hardware y software

9.1 Localmente

Para ejecutar el sistema de manera local en un equipo de desarrollo, se requiere:

Procesador: Intel Core i5 o AMD Ryzen 5 3500 U

500GB de espacio.

8GB de RAM.

Software mínimo:

Sistema operativo: Windows 10/macOS 11+ o superior.

Software instalado:

Visual Studio 2022.

SQL Server Management Studio 20.

9.2 Servidor

Hardware mínimo:

CPU (Procesador)

4 núcleos a 2.5 GHz o superior.

Memoria RAM

8 GB de Memoria RAM.

Almacenamiento (Disco Duro)

SSD de al menos 100 GB.

HDD de 500GB.

Soporte para RAID 1 o RAID 5 para la redundancia y seguridad de datos.

Ancho de banda de red

Conexión mínima de al menos 1 Gbps para la red interna del servidor.

Ancho de banda de mínimo 100/500 Mbps para tráfico hacia internet.

Backup y redundancia

Sistema de backup diario/automático en un disco duro secundario o almacenamiento en la nube.

10. Instalación y configuración

Pasos para la instalación

Clonar el proyecto directamente desde el repositorio de GitHub o descargar el código fuente.

Configurar la base de datos.

Configurar el archivo de conexión a la base de datos.