

Sistema Integral de Control de Stock

Trabajo práctico integrador - Tecnologías de Desarrollo de Software
IDE



Integrantes del grupo:

- Sello, Valentino;
- Tineo, Tomás Iván - 51379

Comisión 3k1

Año: 2025

Introducción

El sistema es una solución integral de control de stock y gestión de entidades, desarrollada sobre .NET 8 y C# 12, con una arquitectura por capas y aplicaciones cliente que consumen una API REST. La solución incluye una aplicación de escritorio basada en WinForms para operaciones administrativas, y proyectos web modernos (Blazor).

En cuanto a su propósito, el sistema busca centralizar la administración de información comercial (clientes, productos, movimientos de stock), ofrecer una interfaz eficaz para consulta y edición, y garantizar seguridad mediante autenticación por token. Todos los módulos de entidades tienen un funcionamiento y apariencia similar, incluyen grillas donde se muestran los datos de las clases del dominio y un panel de carga y edición. La búsqueda es reactiva en tiempo real a través del evento `TextChanged`, filtrando en memoria sobre, por ejemplo, el nombre y actualizando el origen de datos mostrado para evitar bloqueos o inconsistencias visuales.

Flujo de usuario

Bienvenida

Al ejecutar el programa se recibe al usuario con una pantalla de bienvenida donde deberá iniciar sesión o registrarse.

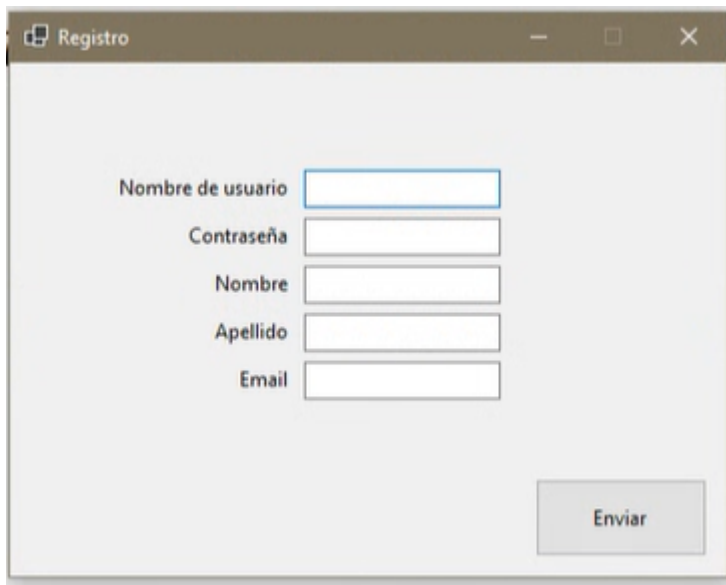


Inicio de sesión

Los nuevos usuarios creados serán “operadores”. Esto implica que tendrán limitados los permisos de uso del sistema.

Estos permisos de operador son:

- Reservas: sería su principal función, por lo tanto tendría todos los permisos necesarios para llevar a cabo las operaciones de alta, lectura, modificación, baja.
- Clientes: alta, lectura, modificación, baja.
- Productos: solo lectura y actualización del campo 'stock' a la hora de crear reserva.
- Eventos: solo lectura.

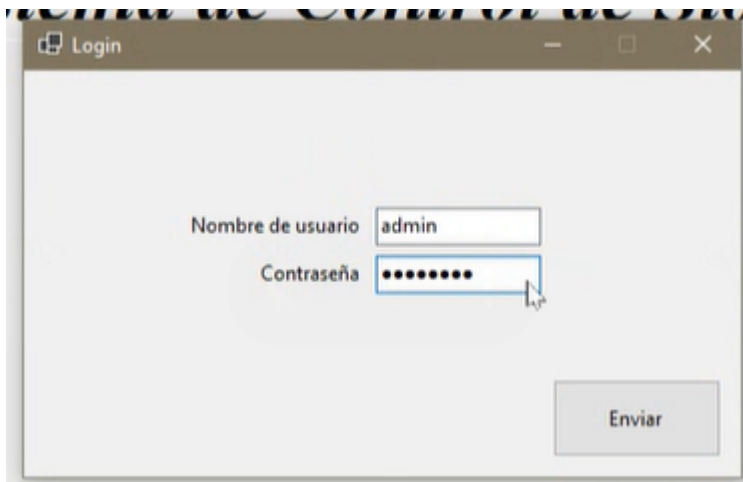
A screenshot of a web application window titled "Registro". It contains five input fields stacked vertically, each with a label to its left: "Nombre de usuario", "Contraseña", "Nombre", "Apellido", and "Email". The "Nombre de usuario" field is highlighted with a blue border. At the bottom right of the form is a button labeled "Enviar".

El administrador, que es único por el momento, debe iniciar sesión con las credenciales preestablecidas:

Usuario: admin

Contraseña: admin123

El administrador tiene todos los permisos dentro de la aplicación.

A screenshot of a web application window titled "Login". It contains two input fields: "Nombre de usuario" with the text "admin" entered, and "Contraseña" with masked characters (dots). The "Contraseña" field is highlighted with a blue border. At the bottom right is a button labeled "Enviar".

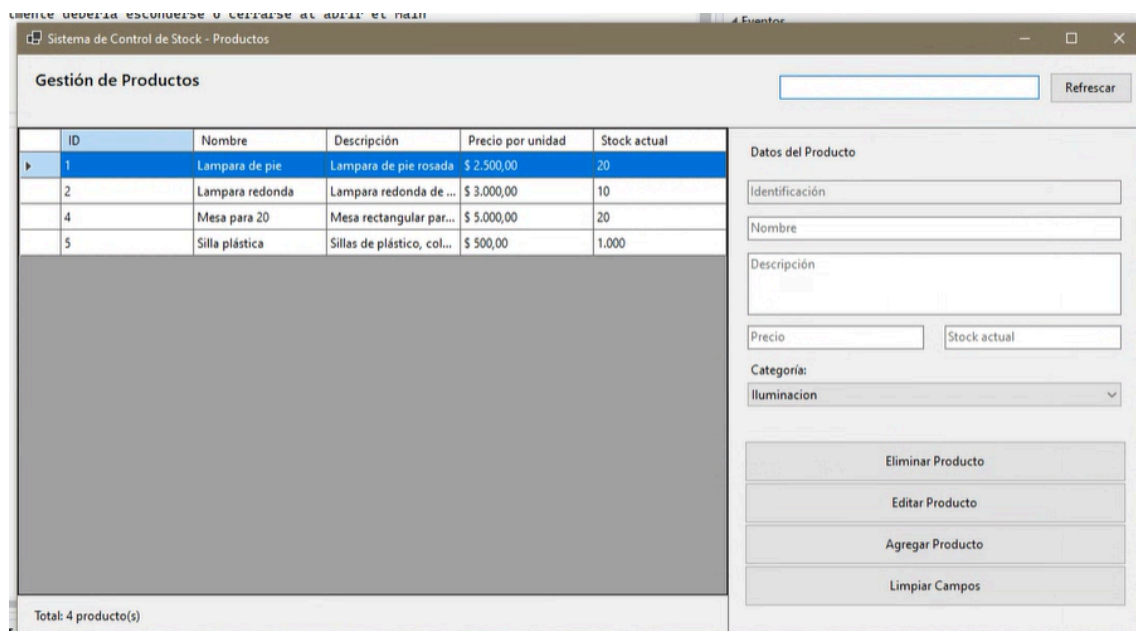
Menú principal

El menú principal es la base de redireccionamiento hacia los módulos CRUD de las entidades de la aplicación.



Gestión de productos

Aquí el administrador puede cargar los productos que tiene el respectivo negocio en stock. Además podrá modificar sus atributos si es necesario.



Gestión de categorías

Cada producto tiene una categoría. En este módulo se gestionan esas categorías.

The screenshot shows the 'Gestión de Categorías' window. It features a table with 2 columns: 'ID' and 'Nombre'. The table contains 4 rows of data. To the right of the table is a 'Datos de la Categoría' section with two input fields: 'Identificación' and 'Nombre de la Categoría'. Below these fields are four buttons: 'Eliminar Categoría', 'Editar Categoría', 'Agregar Categoría', and 'Limpiar Campos'. A 'Refrescar' button is located at the top right. At the bottom left, a status bar indicates 'Total: 4 categoría(s)'.

ID	Nombre
1	Iluminacion
2	Mesas
3	CandyBar
4	Asientos plásticos

Total: 4 categoría(s)

Gestión de eventos

The screenshot shows the 'Gestión de Eventos' window. It features a table with 4 columns: 'ID', 'Nombre del Evento', and 'Fecha'. The table contains 3 rows of data. To the right of the table is a 'Datos del Evento' section with three input fields: 'Identificación', 'Nombre del Evento', and 'Fecha del Evento:'. Below these fields are four buttons: 'Eliminar Evento', 'Editar Evento', 'Agregar Evento', and 'Limpiar Campos'. A 'Refrescar' button is located at the top right. At the bottom left, a status bar indicates 'Total: 3 evento(s)'.

ID	Nombre del Evento	Fecha
1	Casamiento	27/10/2025
3	Cumpleaños Valentino	12/11/2025
4	Casamiento Juan	26/11/2025

Total: 3 evento(s)

Gestión de reservas

El módulo principal del sistema. Desde aquí el usuario operador puede gestionar las reservas, asignándole un cliente, evento, una fecha de finalización de reserva, un estado y los productos que requiere el cliente.

Además se mostrará un monto total de la reserva, con el precio de cada producto.

La grilla principal mostrará los datos de las reservas creadas.

Sistema de Control de Stock - Reservas

Gestión de Reservas

ID	Cliente	Evento	Fecha Creación	Fecha Finalización	Estado	Total
----	---------	--------	----------------	--------------------	--------	-------

Total: 0 reserva(s)

Datos de la Reserva

ID (auto)

Cliente:

Evento:

Casamiento

Fecha Finalización:

7/12/2025

Estado:

Pendiente

Productos:

Producto	Cantidad	Precio Unit.	Subtotal
----------	----------	--------------	----------

Lampara de pie

1

Agregar

Quitar

Total: **\$0.00**

Eliminar Reserva

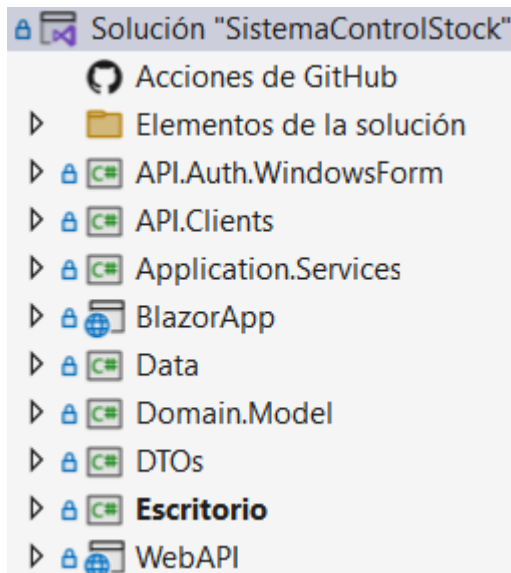
Editar Reserva

Crear Reserva

Limpiar Campos

Arquitectura de carpetas

La solución “SistemaControlStock” está estructurada en capas con diferentes proyectos. De esta forma se logra la separación de responsabilidades y el orden.



La arquitectura por capas se estructura en Presentación, Aplicación/Servicios, Dominio e Infraestructura.

- Domain.Model: modelos de dominio, entidades y reglas básicas.
- Data: AppDbContext, configuraciones EF Core, migraciones y repositorios.
- Application.Services: casos de uso, servicios de negocio, orquestación entre datos y dominio.
- DTOs: contratos de datos para transporte entre capas y clientes (incluye ReportesDTO).
- WebAPI: endpoints REST, configuración Program.cs, appsettings.json, inyección de dependencias y mapeos.
- API.Clients: clientes HTTP para consumir la WebAPI desde otras apps.
- BlazorApp: UI interactiva en Blazor, componentes, páginas, HttpClient para integrar con API.

- API.Auth.WindowsForm: utilidad de autenticación/escritorio para pruebas de login.
- Escritorio: aplicación de escritorio (WinForms), consume servicios y muestra datos.

Tecnologías utilizadas

- .NET 8 y C# 12
- ASP.NET Core Web API (WebAPI)
- Entity Framework Core (contexto Data\AppDbContext.cs)
- Windows Forms (API.Auth.WindowsForm, Escritorio) y Blazor (cliente web en BlazorApp)
- Configuración via appsettings.json (JSON)
- SQL Server 2022 express
- [ASP.Net](#) core
- Visual Studio 2022
- Swagger