# 

DOCUMENTO MANUAL TECNICO DEL SISTEMA DE VENTAS DE PRODUCTOS

1. Curso de Construcción de Software

(Farfán Sánchez Abraham Andrés, Agurto Pincay José Andrés,

Pincay Álvarez Pablo Salvador, Vivanco García Ángel Enrique)

**Tabla de contenido**

[Introducción 5](#_Toc187326018)

[Objetivo del sistema 6](#_Toc187326019)

[Alcances del sistema 6](#_Toc187326020)

[Importancia del Manual 7](#_Toc187326021)

[Manual de Normas, Políticas y Procedimientos de la Organización en las que se basa el Sistema para su Implementación 8](#_Toc187326022)

[Normas y Políticas 8](#_Toc187326023)

[Procedimientos Operativos 9](#_Toc187326024)

[Recomendaciones 9](#_Toc187326025)

[Mapeo AS IS Registrar Producto 10](#_Toc187326026)

[Mapeo TO BE Registrar Producto 11](#_Toc187326027)

[Mapeo AS IS Registrar Compra 12](#_Toc187326028)

[Mapeo TO BE Registrar Compra 13](#_Toc187326029)

[Mapeo AS IS Registrar Venta 14](#_Toc187326030)

[Mapeo TO BE Registrar Venta 15](#_Toc187326031)

[Descripción de base de datos y diagramas de relación 16](#_Toc187326032)

[Descripción de las tablas 16](#_Toc187326033)

[Tabla Rol 16](#_Toc187326034)

[Tabla Permiso 16](#_Toc187326035)

[Tabla Usuario 16](#_Toc187326036)

[Tabla Proveedor 17](#_Toc187326037)

[Tabla Transportista 17](#_Toc187326038)

[Tabla Cliente 17](#_Toc187326039)

[Tabla Categoría 18](#_Toc187326040)

[Tabla Unidad de medida 18](#_Toc187326041)

[Tabla Producto 18](#_Toc187326042)

[Tabla Zona de almacén 18](#_Toc187326043)

[Tabla Inventario 19](#_Toc187326044)

[Tabla Oferta 19](#_Toc187326045)

[Tabla Compra 19](#_Toc187326046)

[Tabla Detalle Compra 20](#_Toc187326047)

[Tabla Venta 20](#_Toc187326048)

[Tabla Detalle Venta 20](#_Toc187326049)

[Tabla Negocio 21](#_Toc187326050)

[Tabla Sucursal 21](#_Toc187326051)

[Descripción de los procedimientos almacenados 22](#_Toc187326052)

[PA\_REGISTRAR\_USUARIO 22](#_Toc187326053)

[PA\_EDITAR\_USUARIO 22](#_Toc187326054)

[PA\_ELIMINAR\_USUARIO 23](#_Toc187326055)

[PA\_REGISTRAR\_CATEGORIA 23](#_Toc187326056)

[PA\_EDITAR\_CATEGORIA 23](#_Toc187326057)

[PA\_ELIMINAR\_CATEGORIA 24](#_Toc187326058)

[PA\_REGISTRAR\_PRODUCTO 24](#_Toc187326059)

[PA\_EDITAR\_PRODUCTO 24](#_Toc187326060)

[PA\_ELIMINAR\_PRODUCTO 25](#_Toc187326061)

[PA\_REGISTRAR\_UNIDAD\_MEDIDA 25](#_Toc187326062)

[PA\_EDITAR\_UNIDAD\_MEDIDA 26](#_Toc187326063)

[PA\_ELIMINAR\_UNIDAD\_MEDIDA 26](#_Toc187326064)

[PA\_REGISTRAR\_CLIENTE 26](#_Toc187326065)

[PA\_EDITAR\_CLIENTE 27](#_Toc187326066)

[PA\_ELIMINAR\_CLIENTE 27](#_Toc187326067)

[PA\_REGISTRAR\_PROVEEDOR 28](#_Toc187326068)

[PA\_EDITAR\_PROVEEDOR 28](#_Toc187326069)

[PA\_ELIMINAR\_PROVEEDOR 29](#_Toc187326070)

[PA\_REGISTRAR\_COMPRA 29](#_Toc187326071)

[PA\_REGISTRAR\_VENTA 30](#_Toc187326072)

[EDetalle\_Compra y EDetalle\_Venta 30](#_Toc187326073)

[PA\_REPORTE\_COMPRA 31](#_Toc187326074)

[PA\_REPORTE\_VENTA 31](#_Toc187326075)

[PA\_REGISTRAR\_OFERTA 32](#_Toc187326076)

[PA\_EDITAR\_OFERTA 32](#_Toc187326077)

[PA\_ELIMINAR\_OFERTA 33](#_Toc187326078)

[PA\_REGISTRAR\_SUCURSAL 33](#_Toc187326079)

[PA\_EDITAR\_SUCURSAL 33](#_Toc187326080)

[PA\_ELIMINAR\_SUCURSAL 34](#_Toc187326081)

[PA\_REGISTRAR\_TRANSPORTISTA 34](#_Toc187326082)

[PA\_EDITAR\_TRANSPORTISTA 35](#_Toc187326083)

[PA\_ELIMINAR\_TRANSPORTISTA 35](#_Toc187326084)

[PA\_EDITAR\_NEGOCIO 36](#_Toc187326085)

[Configuración de la Conexión en app.config 37](#_Toc187326086)

[Implementación en el Código 37](#_Toc187326087)

[Diagrama modelo relacional 38](#_Toc187326088)

[Diseño de reportes y pantallas 39](#_Toc187326089)

[Diseño de reportes 39](#_Toc187326090)

[Reporte de Compras realizadas 39](#_Toc187326091)

[Diseño de la ventana Reportes Compras 40](#_Toc187326092)

[Reporte de Ventas realizadas 42](#_Toc187326093)

[Diseño de la ventana Reportes Ventas 43](#_Toc187326094)

[Diseño de pantallas 45](#_Toc187326095)

[Pantalla de Inicio 46](#_Toc187326096)

[Diseño de la ventana Login 46](#_Toc187326097)

[Diseño de la ventana Recuperar Clave 47](#_Toc187326098)

[Diseño de la ventana Menú 48](#_Toc187326099)

[Pantalla de Gestión 50](#_Toc187326100)

[Diseño de la ventana Producto 50](#_Toc187326101)

[Diseño de la ventana Categoría 52](#_Toc187326102)

[Diseño de la ventana Unidad de Medida 54](#_Toc187326103)

[Diseño de la ventana Zona de Almacén 56](#_Toc187326104)

[Diseño de la ventana Oferta 58](#_Toc187326105)

[Pantalla de Consulta 60](#_Toc187326106)

[Diseño de la ventana Usuario 60](#_Toc187326107)

[Diseño de la ventana Proveedor 62](#_Toc187326108)

[Diseño de la ventana Transportista 64](#_Toc187326109)

[Diseño de la ventana Cliente 66](#_Toc187326110)

[Diseño de la ventana Sucursal 68](#_Toc187326111)

[Pantalla Transaccionales 70](#_Toc187326112)

[Diseño de la ventana Compra 70](#_Toc187326113)

[Diseño de la ventana Detalle Compra 72](#_Toc187326114)

[Diseño de la ventana Venta 74](#_Toc187326115)

[Diseño de la ventana Detalle Venta 76](#_Toc187326116)

[Pantalla de Configuración 78](#_Toc187326117)

[Diseño de la ventana Información Negocio 78](#_Toc187326118)

[Diseño de la ventana Información Programa 80](#_Toc187326119)

[Pantalla de Informe y Analítica 82](#_Toc187326120)

[Diseño de la ventana Graficas Compras 82](#_Toc187326121)

[Diseño de la ventana Graficas Ventas 84](#_Toc187326122)

[Pantalla Modales 86](#_Toc187326123)

[Diseño de la ventana Modal Productos 86](#_Toc187326124)

[Diseño de la ventana Modal Proveedores 88](#_Toc187326125)

[Diseño de la ventana Modal Transportistas 90](#_Toc187326126)

[Diseño de la ventana Modal Clientes 92](#_Toc187326127)

[Diseño de la ventana Modal Ofertas 94](#_Toc187326128)

**Manual técnico**

# Introducción

El presente manual técnico del sistema de ventas para el supermercado ha sido diseñado como una guía detallada y exhaustiva para administradores, técnicos y personal encargado de la implementación y mantenimiento del software. Su propósito es proporcionar toda la información técnica necesaria para garantizar el correcto funcionamiento del sistema, abarcando desde su instalación y configuración inicial hasta la resolución de problemas comunes.

En este documento se incluyen especificaciones técnicas del sistema, requisitos de hardware y software, estructura de la base de datos, descripción de los módulos principales y su integración, así como pasos detallados para la configuración y actualización del sistema. Además, se proporciona información sobre las medidas de seguridad, gestión de usuarios y mantenimiento preventivo y correctivo, asegurando un desempeño óptimo a largo plazo.

El manual está estructurado de manera lógica para facilitar la consulta y servir como una referencia confiable tanto durante la puesta en marcha como en las operaciones diarias del sistema. Este documento es esencial para garantizar que el sistema de ventas sea implementado, administrado y actualizado correctamente, permitiendo una operación eficiente y confiable dentro del supermercado.

# Objetivo del sistema

Desarrollar un sistema de ventas que permita al supermercado registrar y gestionar de manera eficiente sus ventas diarias, compras, inventario y otros procesos relacionados. El sistema debe ser fácil de usar, preciso y capaz de proporcionar información valiosa para la toma de decisiones.

# Alcances del sistema

Nuestra solución permitirá:

* Registrar y gestionar ventas diarias de manera precisa y rápida
* Mantener un inventario actualizado y automático
* Gestionar compras y proveedores de forma eficiente
* Generar reportes y análisis de ventas y compras
* Administrar información de clientes, proveedores y transportistas

# Importancia del Manual

La importancia de un manual técnico para el sistema de ventas del supermercado radica en los siguientes aspectos clave:

* **Guía para la Instalación y Configuración**  
  El manual técnico proporciona instrucciones detalladas sobre cómo instalar y configurar correctamente el sistema en los equipos del supermercado, asegurando que cumpla con los requisitos específicos de hardware, software y red.
* **Resolución de Problemas**  
  Actúa como una referencia para identificar y solucionar problemas técnicos, reduciendo el tiempo de inactividad del sistema y asegurando la continuidad operativa del supermercado.
* **Optimización de Recursos Técnicos**  
  Permite que el equipo de soporte técnico y los administradores comprendan el funcionamiento interno del sistema, como la estructura de la base de datos, las conexiones entre módulos y los procesos automatizados, optimizando el uso de los recursos tecnológicos disponibles.
* **Mantenimiento Preventivo y Correctivo**  
  Detalla procedimientos para realizar mantenimiento preventivo y correctivo, asegurando la longevidad y el buen desempeño del sistema a lo largo del tiempo.
* **Capacitación Técnica**  
  Facilita la capacitación de nuevos técnicos o administradores encargados del sistema, acelerando su integración y capacidad para gestionar el software de manera independiente.
* **Seguridad del Sistema**  
  Proporciona lineamientos para la gestión de usuarios, permisos, y medidas de protección de datos, minimizando riesgos relacionados con accesos no autorizados o pérdida de información.

# Manual de Normas, Políticas y Procedimientos de la Organización en las que se basa el Sistema para su Implementación

## Normas y Políticas

1. **Capacitación Continua**  
   Todos los empleados deben participar en los programas de capacitación relacionados con el sistema de ventas. Estos programas incluirán el uso eficiente del software, la gestión de inventarios y el manejo de datos de ventas y compras. La capacitación continua asegura un manejo adecuado del sistema y maximiza la productividad en las operaciones diarias.
2. **Comportamiento Profesional**  
   Los empleados deben mantener una actitud profesional y respetuosa hacia clientes, proveedores y compañeros de trabajo. El uso del sistema debe realizarse de manera eficiente, evitando errores, y con un alto nivel de responsabilidad para mantener la calidad del servicio ofrecido.
3. **Confidencialidad de la Información**  
   Toda la información almacenada en el sistema, como datos de ventas, clientes y proveedores, debe ser manejada con estricta confidencialidad. Está prohibido compartir o utilizar esta información para fines no relacionados con las operaciones comerciales del supermercado.
4. **Cumplimiento de Tiempos y Entregas**  
   El personal encargado de gestionar las compras y ventas debe cumplir con los plazos establecidos para actualizar inventarios y generar reportes. La puntualidad en estas actividades es esencial para garantizar un flujo de trabajo ágil y eficiente.
5. **Manejo Adecuado del Inventario**  
   El sistema de gestión de inventarios debe actualizarse en tiempo real con cada movimiento de productos (ventas, compras, devoluciones). Los empleados son responsables de garantizar que el inventario refleje la disponibilidad real de los productos, evitando inconsistencias.
6. **Solución de Problemas y Mejora Continua**  
   Es obligatorio notificar cualquier anomalía en el sistema, ya sea en el registro de ventas, inventarios o la interacción con proveedores. Esto permitirá tomar medidas correctivas oportunas y fomentar la mejora continua del sistema. La comunicación clara y efectiva entre los departamentos involucrados es fundamental para este proceso.

## Procedimientos Operativos

1. **Registro de Ventas**

* Los cajeros o encargados de ventas deben registrar las transacciones diarias con precisión utilizando el sistema.
* El sistema actualiza automáticamente el inventario tras cada venta y genera un recibo de pago para el cliente.

1. **Gestión de Inventarios**

* Cada vez que se realice una venta, el sistema ajustará automáticamente el stock de productos.
* Los empleados deben verificar regularmente los niveles de inventario, especialmente para productos de alta rotación, y tomar las medidas necesarias para evitar desabastecimientos o excesos.

1. **Procesamiento de Compras y Proveedores**

* Los empleados podrán gestionar las compras de manera eficiente utilizando el sistema, que permite generar órdenes de compra directamente.
* El sistema está integrado con los proveedores registrados, lo que facilita la emisión de pedidos y el control de las entregas.

1. **Generación de Reportes**

* El sistema generará automáticamente reportes sobre las ventas diarias y las compras realizadas.
* Estos informes serán utilizados por los administradores para analizar el rendimiento del supermercado y tomar decisiones estratégicas relacionadas con el stock, precios y promociones.

1. **Gestión de Datos de Clientes, Proveedores y Transportistas**

* La información relevante sobre clientes, proveedores y transportistas se almacenará en una base de datos centralizada.
* Es responsabilidad de los empleados ingresar los datos correctamente y mantenerlos actualizados para garantizar una comunicación efectiva y un seguimiento adecuado de las operaciones.

## Recomendaciones

* Los empleados deben tener acceso a soporte técnico en caso de problemas con el sistema.
* Es fundamental que los inconvenientes sean resueltos con rapidez para evitar interrupciones en las operaciones diarias del supermercado.

## Mapeo AS IS Registrar Producto

*Diagrama

Descripción generada automáticamente*

## Mapeo TO BE Registrar Producto

Diagrama

Descripción generada automáticamente

## Mapeo AS IS Registrar Compra

**Diagrama

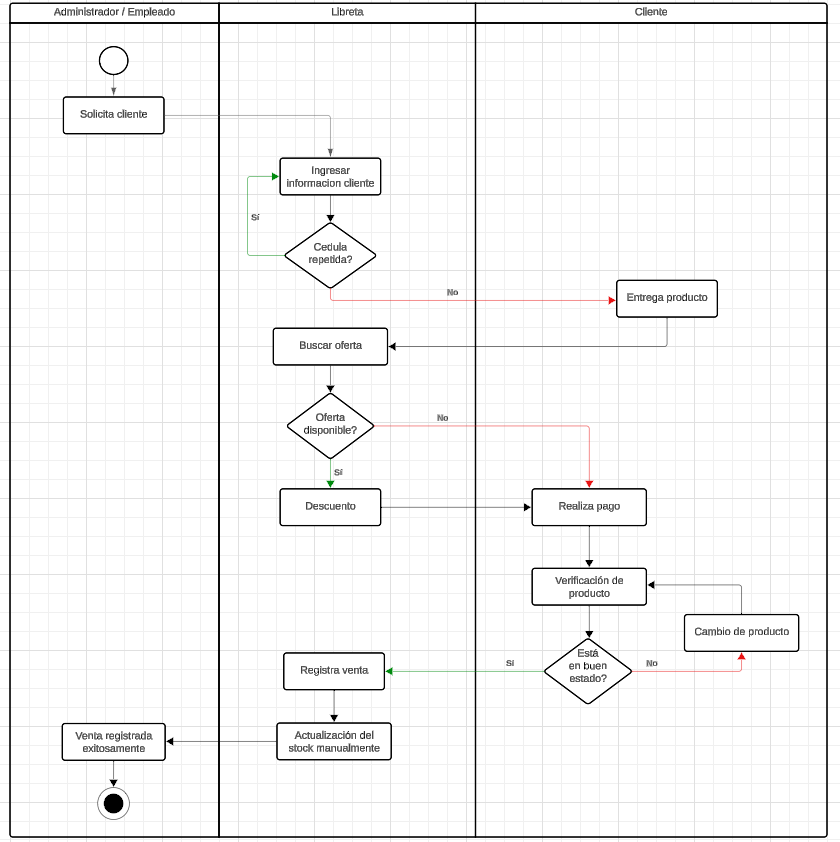
Descripción generada automáticamente**

## Mapeo TO BE Registrar Compra

Diagrama

Descripción generada automáticamente

## Mapeo AS IS Registrar Venta



## Mapeo TO BE Registrar VentDiagrama, Esquemático Descripción generada automáticamentea

# Descripción de base de datos y diagramas de relación

Para el sistema de ventas de productos se ha creado un total de 18 tablas y 35 procedimientos almacenados.

## Descripción de las tablas

### Tabla Rol

**Propósito:** Define los roles del sistema, como administrador, cajero, etc.

**Columnas:**

* ID\_ROL: Identificador único del rol (clave primaria).
* DESCRIPCION: Breve descripción del rol.
* FECHA\_CREACION: Fecha y hora de creación del rol.

### Tabla Permiso

**Propósito:** Asigna permisos a los roles del sistema para acceder a diferentes menús o funcionalidades.

**Columnas:**

* ID\_PERMISO: Identificador único del permiso (clave primaria).
* ID\_ROL: Relación con la tabla ROL, especifica a qué rol pertenece el permiso.
* NOMBRE\_MENU: Nombre del menú o funcionalidad asociada al permiso.
* FECHA\_CREACION: Fecha y hora de creación del permiso.

### Tabla Usuario

**Propósito:** Gestiona los usuarios del sistema.

**Columnas:**

* ID\_USUARIO: Identificador único del usuario (clave primaria).
* CODIGO: Código interno del usuario.
* NOMBRE\_COMPLETO: Nombre completo del usuario.
* CORREO\_ELECTRONICO: Dirección de correo electrónico.
* CLAVE: Contraseña del usuario.
* ID\_ROL: Relación con la tabla ROL.
* ESTADO: Estado activo/inactivo del usuario.
* FECHA\_CREACION: Fecha y hora de creación.

### Tabla Proveedor

**Propósito:** Almacena información sobre los proveedores del supermercado.

**Columnas:**

* ID\_PROVEEDOR: Identificador único del proveedor (clave primaria).
* CODIGO: Código interno del proveedor.
* NOMBRES: Nombres del proveedor.
* APELLIDOS: Apellidos del proveedor.
* CEDULA: Identificación única del proveedor.
* TELEFONO: Número telefónico del proveedor.
* CORREO\_ELECTRONICO: Dirección de correo electrónico.
* ESTADO: Estado activo/inactivo del proveedor.
* FECHA\_REGISTRO: Fecha y hora de registro.

### Tabla Transportista

**Propósito:** Almacena información de los transportistas que colaboran con el supermercado.

**Columnas**

* ID\_TRANSPORTISTA: Identificador único del transportista (clave primaria).
* CODIGO: Código interno del transportista.
* NOMBRES: Nombres del transportista.
* APELLIDOS: Apellidos del transportista.
* CEDULA: Identificación única del transportista.
* TELEFONO: Número telefónico del transportista.
* CORREO\_ELECTRONICO: Dirección de correo electrónico.
* IMAGEN: Foto del transportista (almacenada en formato binario).
* ESTADO: Estado activo/inactivo del transportista.
* FECHA\_REGISTRO: Fecha y hora de registro.

### Tabla Cliente

**Propósito:** Registra los datos de los clientes.

**Columnas:**

* ID\_CLIENTE: Identificador único del cliente (clave primaria).
* CODIGO: Código interno del cliente.
* NOMBRES: Nombres del cliente.
* APELLIDOS: Apellidos del cliente.
* CEDULA: Identificación única del cliente.
* TELEFONO: Número telefónico del cliente.
* CORREO\_ELECTRONICO: Dirección de correo electrónico.
* ESTADO: Estado activo/inactivo del cliente.
* FECHA\_REGISTRO: Fecha y hora de registro.

### Tabla Categoría

**Propósito:** Define categorías para agrupar productos.

**Columnas:**

* ID\_CATEGORIA: Identificador único (clave primaria).
* CODIGO: Código interno.
* DESCRIPCION: Descripción de la categoría.
* ESTADO: Estado activo/inactivo.
* FECHA\_CREACION: Fecha y hora de creación.

### Tabla Unidad de medida

**Propósito:** Gestiona las unidades de medida para los productos.

**Columnas:**

* ID\_UNIDAD\_MEDIDA: Identificador único (clave primaria).
* CODIGO: Código interno.
* DESCRIPCION: Descripción de la unidad.
* SIMBOLO: Símbolo representativo (ej. "kg", "lt").
* ESTADO: Estado activo/inactivo.
* FECHA\_CREACION: Fecha y hora de creación.

### Tabla Producto

**Propósito:** Contiene los detalles de los productos.

**Columnas:**

* ID\_PRODUCTO: Identificador único (clave primaria).
* CODIGO: Código interno.
* DESCRIPCION: Breve descripción.
* NOMBRE\_PRODUCTO: Nombre del producto.
* ID\_CATEGORIA: ID de la categoría (llave foránea).
* ID\_UNIDAD\_MEDIDA: ID de la unidad de medida (llave foránea).
* PAIS\_ORIGEN: País de origen.
* STOCK: Cantidad disponible.
* PRECIO\_COMPRA: Costo del producto.
* PRECIO\_VENTA: Costos del producto.
* ESTADO: Estado activo/inactivo.

### Tabla Zona de almacén

**Propósito:** Define zonas para almacenar productos.

**Columnas:**

* ID\_ZONA: Identificador único (clave primaria).
* NOMBRE\_ZONA: Nombre de la zona.
* LIMITE\_ESPACIOS: Capacidad máxima.
* ESTADO: Estado activo/inactivo.
* FECHA\_REGISTRO: Fecha de registro.

### Tabla Inventario

**Propósito:** Registra el inventario de productos.

**Columnas:**

* ID\_INVENTARIO: Identificador único (clave primaria).
* ID\_PRODUCTO: ID del producto (llave foránea).
* ID\_ZONA: ID de la zona del almacén (llave foránea).
* CANTIDAD: Cantidad disponible.
* FECHA\_INGRESO: Fecha de registro.

### Tabla Oferta

**Descripción**: Administra las ofertas o promociones aplicadas a productos específicos.

**Campos**:

* ID\_OFERTA: Identificador único de la oferta (clave primaria).
* CODIGO: Código único de la oferta.
* NOMBRE\_OFERTA: Nombre de la oferta.
* ID\_PRODUCTO: ID del producto (llave foránea).
* DESCRIPCION: Descripción de la oferta.
* FECHA\_INICIO: Fecha de inicio de la oferta.
* FECHA\_FIN: Fecha de finalización de la oferta.
* DESCUENTO: Porcentaje de descuento aplicado en la oferta.
* ESTADO: Estado de la oferta (activa/inactiva).
* FECHA\_CREACION: Fecha de creación de la oferta.

### Tabla Compra

**Descripción**: Esta tabla registra todas las compras realizadas por el sistema.

**Campos**:

* ID\_COMPRA: Identificador único para cada compra (clave primaria).
* ID\_USUARIO: ID del usuario (llave foránea).
* ID\_PROVEEDOR: ID del proveedor (llave foránea).
* ID\_TRANSPORTISTA: ID del transportista (llave foránea).
* TIPO\_DOCUMENTO: Tipo de documento de compra (factura, recibo, etc.).
* NUMERO\_DOCUMENTO: Número del documento asociado a la compra.
* MONTO\_TOTAL: Total del monto de la compra.
* FECHA\_COMPRA: Fecha en la que se realizó la compra (por defecto la fecha actual).

### Tabla Detalle Compra

**Descripción**: Registra los detalles específicos de cada compra, como productos adquiridos.

**Campos**:

* ID\_DETALLE\_COMPRA: Identificador único para cada detalle de compra (clave primaria).
* ID\_COMPRA: ID de la compra (llave foránea).
* ID\_PRODUCTO: ID del producto (llave foránea).
* PRECIO\_COMPRA: Precio unitario del producto adquirido.
* PRECIO\_VENTA: Precio sugerido para la venta del producto.
* CANTIDAD: Cantidad de productos comprados.
* MONTO\_TOTAL: Subtotal del detalle de la compra (cantidad x precio).
* FECHA\_REGISTRO: Fecha de registro del detalle de compra.

### Tabla Venta

**Descripción**: Almacena información sobre las ventas realizadas en el sistema.

**Campos**:

* ID\_VENTA: Identificador único para cada venta (clave primaria).
* ID\_USUARIO: ID del usuario (llave foránea).
* TIPO\_DOCUMENTO: Tipo de documento emitido (boleta, factura, etc.).
* ID\_CLIENTE: ID del cliente (llave foránea).
* NUMERO\_DOCUMENTO: Número del documento emitido por la venta.
* MONTO\_PAGO: Monto pagado por el cliente.
* MONTO\_CAMBIO: Cambio devuelto al cliente.
* MONTO\_TOTAL: Total de la venta antes del descuento.
* DESCUENTO: Monto del descuento aplicado.
* FECHA\_VENTA: Fecha de la venta (por defecto la fecha actual).

### Tabla Detalle Venta

**Descripción**: Contiene los detalles de cada venta, incluyendo productos vendidos.

**Campos**:

* ID\_DETALLE\_VENTA: Identificador único para cada detalle de venta (clave primaria).
* ID\_VENTA: ID de la venta (llave foránea).
* ID\_PRODUCTO: ID el producto (llave foránea).
* PRECIO\_VENTA: Precio unitario del producto vendido.
* CANTIDAD\_PRODUCTO: Cantidad de productos vendidos.
* SUBTOTAL: Subtotal del detalle de venta (cantidad x precio).
* DESCUENTO: Descuento aplicado en el detalle.
* FECHA\_REGISTRO: Fecha de registro del detalle de venta.

### Tabla Negocio

**Descripción**: Contiene información básica sobre el negocio propietario del sistema.

**Campos**:

* ID\_NEGOCIO: Identificador único del negocio (clave primaria).
* NOMBRE: Nombre del negocio.
* TELEFONO: Teléfono del negocio.
* RUC: Registro Único de Contribuyente del negocio.
* DIRECCION: Dirección física del negocio.
* CORREO\_ELECTRONICO: Correo electrónico del negocio.
* LOGO: Imagen del logo del negocio (almacenada en formato binario).

### Tabla Sucursal

**Descripción**: Registra las sucursales asociadas al negocio.

**Campos**:

* ID\_SUCURSAL: Identificador único de la sucursal (clave primaria).
* CODIGO: Código único de la sucursal.
* NOMBRE\_SUCURSAL: Nombre de la sucursal.
* DIRECCION\_SUCURSAL: Dirección física de la sucursal.
* LATITUD\_SUCURSAL: Coordenadas de latitud de la sucursal.
* LONGITUD\_SUCURSAL: Coordenadas de longitud de la sucursal.
* CIUDAD\_SUCURSAL: Ciudad donde se encuentra la sucursal.
* ESTADO: Estado de la sucursal (abierto/cerrado).

## Descripción de los procedimientos almacenados

### PA\_REGISTRAR\_USUARIO

**Propósito:** Registrar un nuevo usuario en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único del usuario.
* @Nombre\_Completo: Nombre completo del usuario.
* @Correo\_Electronico: Correo electrónico del usuario.
* @Clave: Contraseña del usuario.
* @Id\_Rol: ID del rol asignado al usuario.
* @Estado: Estado del usuario (activo/inactivo).

**Parámetros de salida:**

* @Id\_Usuario\_Resultado: ID del usuario registrado.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicia una transacción.
2. Inicializa los parámetros de salida.
3. Verifica si el código del usuario ya existe en la tabla USUARIO.
4. Si el código no existe, inserta el nuevo usuario y obtiene el ID generado.
5. Si el código ya existe, establece un mensaje de error.
6. Maneja errores y realiza un rollback en caso de excepción.

### PA\_EDITAR\_USUARIO

**Propósito:** Editar la información de un usuario existente.

**Parámetros de entrada:**

* @Id\_Usuario: ID del usuario a editar.
* @Codigo: Nuevo código del usuario.
* @Nombre\_Completo: Nuevo nombre completo del usuario.
* @Correo\_Electronico: Nuevo correo electrónico del usuario.
* @Clave: Nueva contraseña del usuario.
* @Id\_Rol: Nuevo ID del rol asignado al usuario.
* @Estado: Nuevo estado del usuario (activo/inactivo).

**Parámetros de salida:**

* @Respuesta: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicia una transacción.
2. Inicializa los parámetros de salida.
3. Verifica si el nuevo código ya está en uso por otro usuario.
4. Si el código está en uso, establece un mensaje de error.
5. Si el código no está en uso, verifica si el usuario con el ID proporcionado existe.
6. Si el usuario existe, actualiza la información del usuario.
7. Si el usuario no existe, establece un mensaje de error.
8. Maneja errores y realiza un rollback en caso de excepción.

### PA\_ELIMINAR\_USUARIO

**Propósito:** Eliminar un usuario de la base de datos.

**Parámetros de entrada:**

* @Id\_Usuario: ID del usuario a eliminar.

**Parámetros de salida:**

* @Respuesta: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicia una transacción.
2. Inicializa los parámetros de salida.
3. Verifica si el usuario está relacionado con alguna compra.
4. Verifica si el usuario está relacionado con alguna venta.
5. Si el usuario no está relacionado con compras o ventas, lo elimina.
6. Si el usuario está relacionado con compras o ventas, establece un mensaje de error.
7. Maneja errores y realiza un rollback en caso de excepción.

### PA\_REGISTRAR\_CATEGORIA

**Propósito:** Registrar una nueva categoría en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único de la categoría.
* @Descripcion: Descripción de la categoría.
* @Estado: Estado de la categoría (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID de la categoría registrada.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el código de la categoría ya existe en la tabla CATEGORIA.
3. Si el código no existe, inserta la nueva categoría y obtiene el ID generado.
4. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_CATEGORIA

**Propósito:** Editar la información de una categoría existente.

**Parámetros de entrada:**

* @Id\_Categoria: ID de la categoría a editar.
* @Codigo: Nuevo código de la categoría.
* @Descripcion: Nueva descripción de la categoría.
* @Estado: Nuevo estado de la categoría (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el nuevo código ya está en uso por otra categoría.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, actualiza la información de la categoría.
5. Si la categoría no existe, establece un mensaje de error.

### PA\_ELIMINAR\_CATEGORIA

**Propósito:** Eliminar una categoría de la base de datos.

**Parámetros de entrada:**

* @Id\_Categoria: ID de la categoría a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si la categoría está relacionada con algún producto.
3. Si la categoría no está relacionada con productos, la elimina.
4. Si la categoría está relacionada con productos, establece un mensaje de error.

### PA\_REGISTRAR\_PRODUCTO

**Propósito:** Registrar un nuevo producto en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único del producto.
* @Nombre\_Producto: Nombre del producto.
* @Descripcion: Descripción del producto.
* @Id\_Categoria: ID de la categoría a la que pertenece el producto.
* @Id\_Unidad\_Medida: ID de la unidad de medida del producto.
* @Pais\_Origen: País de origen del producto.
* @Estado: Estado del producto (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID del producto registrado.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el código del producto ya existe en la tabla PRODUCTO.
3. Si el código no existe, inserta el nuevo producto y obtiene el ID generado.
4. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_PRODUCTO

**Propósito:** Editar la información de un producto existente.

**Parámetros de entrada:**

* @Id\_Producto: ID del producto a editar.
* @Codigo: Nuevo código del producto.
* @Nombre\_Producto: Nuevo nombre del producto.
* @Descripcion: Nueva descripción del producto.
* @Id\_Categoria: Nuevo ID de la categoría del producto.
* @Id\_Unidad\_Medida: Nuevo ID de la unidad de medida del producto.
* @Pais\_Origen: Nuevo país de origen del producto.
* @Estado: Nuevo estado del producto (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el nuevo código ya está en uso por otro producto.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, actualiza la información del producto.
5. Si la actualización es exitosa, establece el indicador de éxito.

### PA\_ELIMINAR\_PRODUCTO

**Propósito:** Eliminar un producto de la base de datos.

**Parámetros de entrada:**

* @Id\_Producto: ID del producto a eliminar.

**Parámetros de salida:**

* @Respuesta: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el producto está relacionado con alguna compra.
3. Verifica si el producto está relacionado con alguna venta.
4. Verifica si el producto está relacionado con el inventario.
5. Verifica si el producto está relacionado con alguna oferta.
6. Si el producto no está relacionado con compras, ventas, inventario u ofertas, lo elimina.
7. Si el producto está relacionado con alguna de estas entidades, establece un mensaje de error.

### PA\_REGISTRAR\_UNIDAD\_MEDIDA

**Propósito:** Registrar una nueva unidad de medida en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único de la unidad de medida.
* @Descripcion: Descripción de la unidad de medida.
* @Simbolo: Símbolo de la unidad de medida.
* @Estado: Estado de la unidad de medida (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID de la unidad de medida registrada.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el código de la unidad de medida ya existe en la tabla UNIDAD\_MEDIDA.
3. Si el código no existe, inserta la nueva unidad de medida y obtiene el ID generado.
4. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_UNIDAD\_MEDIDA

**Propósito:** Editar la información de una unidad de medida existente.

**Parámetros de entrada:**

* @Id\_Unidad\_Medida: ID de la unidad de medida a editar.
* @Codigo: Nuevo código de la unidad de medida.
* @Descripcion: Nueva descripción de la unidad de medida.
* @Simbolo: Nuevo símbolo de la unidad de medida.
* @Estado: Nuevo estado de la unidad de medida (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el nuevo código ya está en uso por otra unidad de medida.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, actualiza la información de la unidad de medida.
5. Si la actualización es exitosa, establece el indicador de éxito y un mensaje de confirmación.

### PA\_ELIMINAR\_UNIDAD\_MEDIDA

**Propósito:** Eliminar una unidad de medida de la base de datos.

**Parámetros de entrada:**

* @Id\_Unidad\_Medida: ID de la unidad de medida a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si la unidad de medida está asociada a algún producto.
3. Si la unidad de medida no está asociada a productos, la elimina.
4. Si la eliminación es exitosa, establece el indicador de éxito y un mensaje de confirmación.
5. Si la unidad de medida está asociada a productos, establece un mensaje de error.

### PA\_REGISTRAR\_CLIENTE

**Propósito:** Registrar un nuevo cliente en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único del cliente.
* @Nombre\_Cliente: Nombre del cliente.
* @Apellido\_Cliente: Apellido del cliente.
* @Cedula: Cédula del cliente.
* @Telefono: Teléfono del cliente.
* @Correo\_Electronico: Correo electrónico del cliente.
* @Estado: Estado del cliente (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID del cliente registrado.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el código del cliente ya existe en la tabla CLIENTE.
3. Si el código no existe, verifica si la cédula ya existe.
4. Si la cédula no existe, inserta el nuevo cliente y obtiene el ID generado.
5. Si la cédula ya existe, establece un mensaje de error.
6. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_CLIENTE

**Propósito:** Editar la información de un cliente existente.

**Parámetros de entrada:**

* @Id\_Cliente: ID del cliente a editar.
* @Codigo: Nuevo código del cliente.
* @Nombre\_Cliente: Nuevo nombre del cliente.
* @Apellido\_Cliente: Nuevo apellido del cliente.
* @Cedula: Nueva cédula del cliente.
* @Telefono: Nuevo teléfono del cliente.
* @Correo\_Electronico: Nuevo correo electrónico del cliente.
* @Estado: Nuevo estado del cliente (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el nuevo código ya está en uso por otro cliente.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, verifica si la nueva cédula ya está en uso por otro cliente.
5. Si la cédula está en uso, establece un mensaje de error.
6. Si la cédula no está en uso, actualiza la información del cliente.
7. Si la actualización es exitosa, establece un mensaje de confirmación.

### PA\_ELIMINAR\_CLIENTE

**Propósito:** Eliminar un cliente de la base de datos.

**Parámetros de entrada:**

* @Id\_Cliente: ID del cliente a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Elimina el cliente con el ID especificado.
3. Si la eliminación es exitosa, establece un mensaje de confirmación.
4. Si no se encuentra el cliente con el ID especificado, establece un mensaje de error.

### PA\_REGISTRAR\_PROVEEDOR

**Propósito:** Registrar un nuevo proveedor en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único del proveedor.
* @Nombres\_Proveedor: Nombres del proveedor.
* @Apellidos\_Proveedor: Apellidos del proveedor.
* @Cedula: Cédula del proveedor.
* @Telefono: Teléfono del proveedor.
* @Correo\_Electronico: Correo electrónico del proveedor.
* @Estado: Estado del proveedor (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID del proveedor registrado.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el código del proveedor ya existe en la tabla PROVEEDOR.
3. Si el código no existe, verifica si la cédula ya existe.
4. Si la cédula no existe, inserta el nuevo proveedor y obtiene el ID generado.
5. Si la cédula ya existe, establece un mensaje de error.
6. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_PROVEEDOR

**Propósito:** Editar la información de un proveedor existente.

**Parámetros de entrada:**

* @Id\_Proveedor: ID del proveedor a editar.
* @Codigo: Nuevo código del proveedor.
* @Nombres\_Proveedor: Nuevos nombres del proveedor.
* @Apellidos\_Proveedor: Nuevos apellidos del proveedor.
* @Cedula: Nueva cédula del proveedor.
* @Telefono: Nuevo teléfono del proveedor.
* @Correo\_Electronico: Nuevo correo electrónico del proveedor.
* @Estado: Nuevo estado del proveedor (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el nuevo código ya está en uso por otro proveedor.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, verifica si la nueva cédula ya está en uso por otro proveedor.
5. Si la cédula está en uso, establece un mensaje de error.
6. Si la cédula no está en uso, actualiza la información del proveedor.
7. Si la actualización es exitosa, establece un mensaje de confirmación.

### PA\_ELIMINAR\_PROVEEDOR

**Propósito:** Eliminar un proveedor de la base de datos.

**Parámetros de entrada:**

* @Id\_Proveedor: ID del proveedor a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el proveedor está relacionado con alguna compra.
3. Si el proveedor no está relacionado con compras, lo elimina.
4. Si la eliminación es exitosa, establece un mensaje de confirmación.
5. Si el proveedor está relacionado con compras, establece un mensaje de error.

### PA\_REGISTRAR\_COMPRA

**Propósito:** Registrar una nueva compra en la base de datos.

**Parámetros de entrada:**

* @Id\_Usuario: ID del usuario que realiza la compra.
* @Id\_Proveedor: ID del proveedor.
* @Id\_Transportista: ID del transportista.
* @Tipo\_Documento: Tipo de documento de la compra.
* @Numero\_Documento: Número del documento de la compra.
* @Monto\_Total: Monto total de la compra.
* @Detalle\_Compra: Tabla de detalles de la compra (tipo EDetalle\_Compra).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Inicia una transacción.
3. Inserta la compra en la tabla COMPRA y obtiene el ID generado.
4. Inserta los detalles de la compra en la tabla DETALLE\_COMPRA utilizando los datos de la tabla @Detalle\_Compra.
5. Actualiza el stock, precio de compra y precio de venta de los productos en la tabla PRODUCTO.
6. Actualiza la cantidad de los productos en la tabla INVENTARIO.
7. Si todo es exitoso, realiza un commit de la transacción.
8. Si ocurre un error, realiza un rollback de la transacción y establece el mensaje de error.

### PA\_REGISTRAR\_VENTA

**Propósito:** Registrar una nueva venta en la base de datos.

**Parámetros de entrada:**

* @Id\_Usuario: ID del usuario que realiza la venta.
* @Tipo\_Documento: Tipo de documento de la venta.
* @Numero\_Documento: Número del documento de la venta.
* @Id\_Cliente: ID del cliente.
* @Monto\_Pago: Monto del pago realizado.
* @Monto\_Cambio: Monto del cambio devuelto.
* @Monto\_Total: Monto total de la venta.
* @Descuento: Descuento aplicado a la venta.
* @Detalle\_Venta: Tabla de detalles de la venta (tipo EDetalle\_Venta).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Inicia una transacción.
3. Inserta la venta en la tabla VENTA y obtiene el ID generado.
4. Inserta los detalles de la venta en la tabla DETALLE\_VENTA utilizando los datos de la tabla @Detalle\_Venta.
5. Actualiza la cantidad de los productos en la tabla INVENTARIO.
6. Si todo es exitoso, realiza un commit de la transacción.
7. Si ocurre un error, realiza un rollback de la transacción y establece el mensaje de error.

### EDetalle\_Compra y EDetalle\_Venta

**Propósito:** Definir tipos de tabla para los detalles de compra y venta.

**Estructura de EDetalle\_Compra:**

* Id\_Producto: ID del producto.
* Precio\_Compra: Precio de compra del producto.
* Precio\_Venta: Precio de venta del producto.
* Cantidad: Cantidad de productos comprados.
* Monto\_Total: Monto total de la compra.

**Estructura de EDetalle\_Venta:**

* IdProducto: ID del producto.
* PrecioVenta: Precio de venta del producto.
* Cantidad: Cantidad de productos vendidos.
* SubTotal: Subtotal de la venta.
* Descuento: Descuento aplicado a la venta.

### PA\_REPORTE\_COMPRA

**Propósito:** Generar un reporte de compras en la base de datos basado en un rango de fechas, proveedor y transportista.

**Parámetros de entrada:**

* @Fecha\_Inicio: Fecha de inicio del rango (formato dd/mm/yyyy).
* @Fecha\_Fin: Fecha de fin del rango (formato dd/mm/yyyy).
* @Id\_Proveedor: ID del proveedor (0 para todos los proveedores).
* @Id\_Transportista: ID del transportista (0 para todos los transportistas).

**Lógica:**

1. Establece el formato de fecha a dmy (día/mes/año).
2. Realiza una consulta SELECT para obtener los detalles de las compras dentro del rango de fechas especificado.
3. La consulta incluye información de las tablas COMPRA, USUARIO, PROVEEDOR, TRANSPORTISTA, DETALLE\_COMPRA, PRODUCTO y CATEGORIA.
4. Filtra las compras por el rango de fechas (@Fecha\_Inicio y @Fecha\_Fin).
5. Filtra las compras por el ID del proveedor y el ID del transportista, si se especifican (si @Id\_Proveedor o @Id\_Transportista son 0, no se aplica el filtro).

### PA\_REPORTE\_VENTA

**Propósito:** Generar un reporte de ventas en la base de datos basado en un rango de fechas.

**Parámetros de entrada:**

* @fecha\_Inicio: Fecha de inicio del rango (formato dd/mm/yyyy).
* @fecha\_Fin: Fecha de fin del rango (formato dd/mm/yyyy).

**Lógica:**

1. Establece el formato de fecha a dmy (día/mes/año).
2. Realiza una consulta SELECT para obtener los detalles de las ventas dentro del rango de fechas especificado.
3. La consulta incluye información de las tablas VENTA, USUARIO, DETALLE\_VENTA, PRODUCTO, CATEGORIA y CLIENTE.
4. Filtra las ventas por el rango de fechas (@fecha\_Inicio y @fecha\_Fin).

### PA\_REGISTRAR\_OFERTA

**Propósito:** Registrar una nueva oferta en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único de la oferta.
* @Nombre\_Oferta: Nombre de la oferta.
* @Id\_Producto: ID del producto asociado a la oferta.
* @Descripcion: Descripción de la oferta.
* @Fecha\_Inicio: Fecha de inicio de la oferta.
* @Fecha\_Fin: Fecha de fin de la oferta.
* @Descuento: Descuento aplicado en la oferta.
* @Estado: Estado de la oferta (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID de la oferta registrada.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el código de la oferta ya existe en la tabla OFERTA.
3. Si el código no existe, inserta la nueva oferta y obtiene el ID generado.
4. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_OFERTA

**Propósito:** Editar la información de una oferta existente.

**Parámetros de entrada:**

* @Id\_Oferta: ID de la oferta a editar.
* @Codigo: Nuevo código de la oferta.
* @Nombre\_Oferta: Nuevo nombre de la oferta.
* @Id\_Producto: Nuevo ID del producto asociado a la oferta.
* @Descripcion: Nueva descripción de la oferta.
* @Fecha\_Inicio: Nueva fecha de inicio de la oferta.
* @Fecha\_Fin: Nueva fecha de fin de la oferta.
* @Descuento: Nuevo descuento aplicado en la oferta.
* @Estado: Nuevo estado de la oferta (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el nuevo código ya está en uso por otra oferta.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, actualiza la información de la oferta.
5. Si la actualización es exitosa, establece el indicador de éxito y un mensaje de confirmación.

### PA\_ELIMINAR\_OFERTA

**Propósito:** Eliminar una oferta de la base de datos.

**Parámetros de entrada:**

* @Id\_Oferta: ID de la oferta a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Elimina la oferta con el ID especificado.
3. Si la eliminación es exitosa, establece el indicador de éxito y un mensaje de confirmación.
4. Si no se encuentra la oferta con el ID especificado, establece un mensaje de error.

### PA\_REGISTRAR\_SUCURSAL

**Propósito:** Registrar una nueva sucursal en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único de la sucursal.
* @Nombre\_Sucursal: Nombre de la sucursal.
* @Direccion\_Sucursal: Dirección de la sucursal.
* @Latitud\_Sucursal: Latitud de la ubicación de la sucursal.
* @Longitud\_Sucursal: Longitud de la ubicación de la sucursal.
* @Ciudad\_Sucursal: Ciudad donde se encuentra la sucursal.
* @Estado: Estado de la sucursal (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID de la sucursal registrada.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el código de la sucursal ya existe en la tabla SUCURSAL.
3. Si el código no existe, inserta la nueva sucursal y obtiene el ID generado.
4. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_SUCURSAL

**Propósito:** Editar la información de una sucursal existente.

**Parámetros de entrada:**

* @Id\_Sucursal: ID de la sucursal a editar.
* @Codigo: Nuevo código de la sucursal.
* @Nombre\_Sucursal: Nuevo nombre de la sucursal.
* @Direccion\_Sucursal: Nueva dirección de la sucursal.
* @Latitud\_Sucursal: Nueva latitud de la ubicación de la sucursal.
* @Longitud\_Sucursal: Nueva longitud de la ubicación de la sucursal.
* @Ciudad\_Sucursal: Nueva ciudad de la sucursal.
* @Estado: Nuevo estado de la sucursal (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Verifica si el nuevo código ya está en uso por otra sucursal.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, actualiza la información de la sucursal.
5. Si la actualización es exitosa, establece el indicador de éxito y un mensaje de confirmación.

### PA\_ELIMINAR\_SUCURSAL

**Propósito:** Eliminar una sucursal de la base de datos.

**Parámetros de entrada:**

* @Id\_Sucursal: ID de la sucursal a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa el parámetro de salida @Resultado a 0.
2. Elimina la sucursal con el ID especificado.
3. Si la eliminación es exitosa, establece el indicador de éxito y un mensaje de confirmación.
4. Si no se encuentra la sucursal con el ID especificado, establece un mensaje de error.

### PA\_REGISTRAR\_TRANSPORTISTA

**Propósito:** Registrar un nuevo transportista en la base de datos.

**Parámetros de entrada:**

* @Codigo: Código único del transportista.
* @Nombres: Nombres del transportista.
* @Apellidos: Apellidos del transportista.
* @Cedula: Cédula del transportista.
* @Telefono: Teléfono del transportista.
* @Correo\_Electronico: Correo electrónico del transportista.
* @Imagen: Imagen del transportista (opcional).
* @Estado: Estado del transportista (activo/inactivo).

**Parámetros de salida:**

* @Resultado: ID del transportista registrado.
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el código del transportista ya existe en la tabla TRANSPORTISTA.
3. Si el código no existe, verifica si la cédula ya existe.
4. Si la cédula no existe, inserta el nuevo transportista y obtiene el ID generado.
5. Si la cédula ya existe, establece un mensaje de error.
6. Si el código ya existe, establece un mensaje de error.

### PA\_EDITAR\_TRANSPORTISTA

**Propósito:** Editar la información de un transportista existente.

**Parámetros de entrada:**

* @Id\_Transportista: ID del transportista a editar.
* @Codigo: Nuevo código del transportista.
* @Nombres: Nuevos nombres del transportista.
* @Apellidos: Nuevos apellidos del transportista.
* @Cedula: Nueva cédula del transportista.
* @Telefono: Nuevo teléfono del transportista.
* @Correo\_Electronico: Nuevo correo electrónico del transportista.
* @Imagen: Nueva imagen del transportista (opcional).
* @Estado: Nuevo estado del transportista (activo/inactivo).

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Verifica si el nuevo código ya está en uso por otro transportista.
3. Si el código está en uso, establece un mensaje de error.
4. Si el código no está en uso, verifica si la nueva cédula ya está en uso por otro transportista.
5. Si la cédula está en uso, establece un mensaje de error.
6. Si la cédula no está en uso, actualiza la información del transportista.
7. Si la actualización es exitosa, establece un mensaje de confirmación.

### PA\_ELIMINAR\_TRANSPORTISTA

**Propósito:** Eliminar un transportista de la base de datos.

**Parámetros de entrada:**

* @Id\_Transportista: ID del transportista a eliminar.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida.
2. Elimina el transportista con el ID especificado.
3. Si la eliminación es exitosa, establece el indicador de éxito y un mensaje de confirmación.
4. Si no se encuentra el transportista con el ID especificado, establece un mensaje de error.

### PA\_EDITAR\_NEGOCIO

**Propósito:** Editar la información de un negocio existente en la base de datos.

**Parámetros de entrada:**

* @Id\_Negocio: ID del negocio a editar.
* @Nombre: Nuevo nombre del negocio.
* @Telefono: Nuevo teléfono del negocio.
* @Ruc: Nuevo RUC (Registro Único de Contribuyentes) del negocio.
* @Direccion: Nueva dirección del negocio.
* @Correo\_Electronico: Nuevo correo electrónico del negocio.

**Parámetros de salida:**

* @Resultado: Indicador de éxito (1) o fracaso (0).
* @Mensaje: Mensaje de resultado.

**Lógica:**

1. Inicializa los parámetros de salida @Resultado a 1 y @Mensaje a una cadena vacía.
2. Realiza una actualización (UPDATE) en la tabla NEGOCIO para modificar los campos NOMBRE, TELEFONO, RUC, DIRECCION y CORREO\_ELECTRONICO del negocio con el ID\_NEGOCIO especificado.
3. Establece el mensaje de salida @Mensaje a "La información del negocio fue actualizada exitosamente."

## Configuración de la Conexión en app.config

En el archivo app.config del proyecto, se ha agregado una sección específica para configurar la conexión a la base de datos. Esta sección se llama <connectionStrings> y contiene la información necesaria para establecer la conexión con la base de datos.

Esta configuración incluye:

* name: El nombre identificador de la cadena de conexión ("ConexionBD").
* connectionString: La cadena que especifica los detalles de la conexión, como el servidor, la base de datos y el método de autenticación.
* providerName: El proveedor utilizado para la conexión (en este caso, SQL Server).

### Implementación en el Código

Para utilizar esta configuración en el código, se ha creado una clase llamada Conexion dentro del namespace Datos. Esta clase contiene un método llamado ConexionBD() que se encarga de establecer la conexión con la base de datos utilizando la información proporcionada en el archivo app.config.

El método ConexionDB() realiza las siguientes acciones:

1. Intenta obtener la cadena de conexión desde el archivo app.config.
2. Crea una nueva instancia de SqlConnection usando esa cadena.
3. Abre la conexión.
4. Si ocurre algún error durante el proceso, lo captura y lo registra.
5. Devuelve la conexión abierta.

Además, para conocer las herramientas utilizadas y la instalación del programa, por favor, lee los siguientes documentos:

* DOCUMENTO DE INSTRUCCIONES DEL PROYECTO IDE
* DOCUMENTO DE ALCANCE

## Diagrama modelo relacional

Diagrama, Esquemático

Descripción generada automáticamente

# Diseño de reportes y pantallas

El sistema de ventas para el supermercado cuenta con dos reportes principales:

1. Reporte de Compras Realizadas
2. Reporte de Ventas Realizadas

Ambos reportes se generan para un rango de fechas específico, definido por una fecha de inicio y una fecha final.

## Diseño de reportes

### Reporte de Compras realizadas

**Objetivo:** Este reporte proporciona un resumen detallado de todas las compras realizadas dentro del rango de fechas seleccionado.

**Estructura:**

El reporte incluye las siguientes columnas:

1. FechaRegistro
2. TipoDocumento
3. NumeroDocumento
4. MontoTotal
5. UsuarioRegistro
6. DocumentoProveedor
7. NombreProveedor
8. DocumentoTransportista
9. NombreTransportista
10. CodigoProducto
11. NombreProducto
12. Categoria
13. PrecioCompra
14. PrecioVenta
15. Cantidad
16. SubTotal

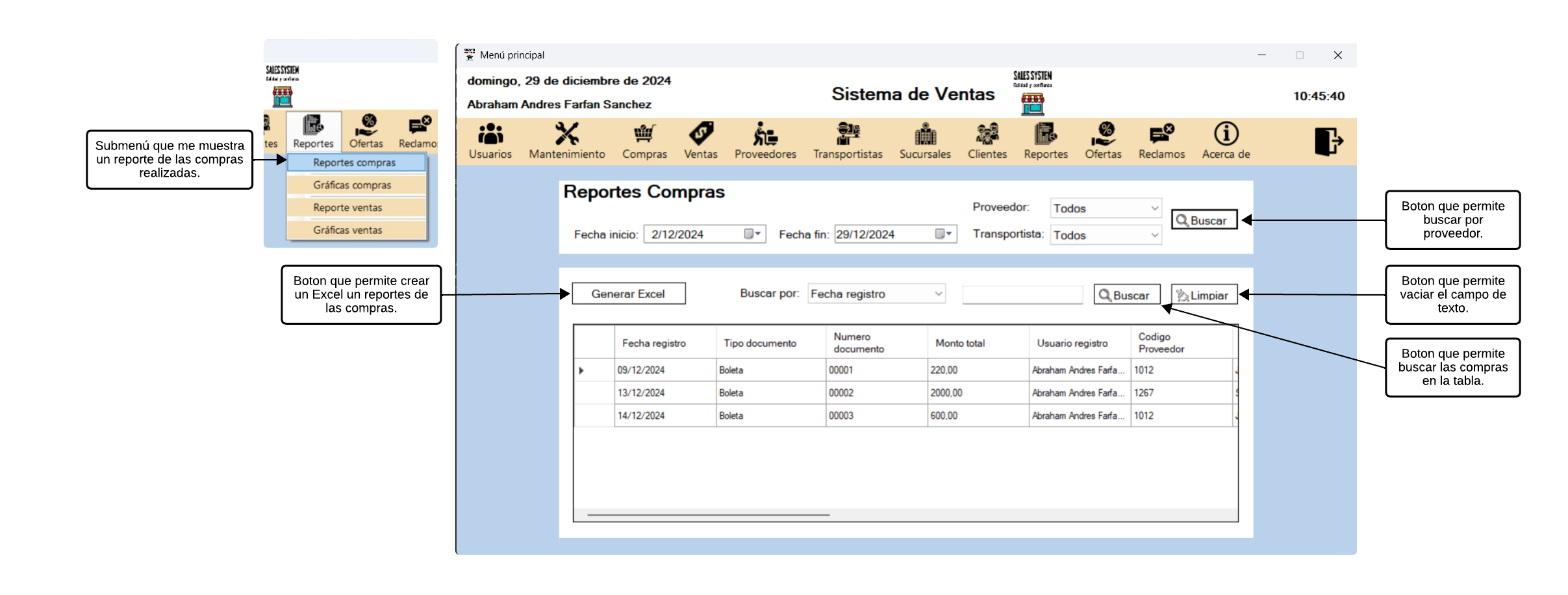
**Filtros y Parámetros:**

El reporte permite filtrar por:

* Proveedor (incluyendo la opción de seleccionar "Todos")
* Transportista (incluyendo la opción de seleccionar "Todos")

**Formatos de Salida:** El reporte se puede visualizar en pantalla y exportar a Excel (.xlsx).

#### Diseño de la ventana Reportes Compras



**Descripción General:**

Esta pantalla permite al usuario generar y visualizar los reportes de compras.

**Elementos Clave:**

1. Campos para seleccionar el rango de fechas (Fecha Inicio y Fecha Fin).
2. Combobox para seleccionar el proveedor (con opción "Todos").
3. Combobox para seleccionar el transportista (con opción "Todos").
4. Botón "Buscar" para generar el reporte.
5. Tabla (DataGridView) para visualizar los resultados del reporte.
6. Botones adicionales:
   * "Generar Excel" para exportar el reporte a formato Excel.
   * "Limpiar" para resetear la búsqueda.
   * "Buscar" en la tabla para filtrar los resultados visibles.

**Funcionalidades:**

* Generación dinámica de reportes basados en los filtros seleccionados.
* Visualización detallada de los datos en una tabla.
* Filtro adicional en la tabla para buscar información específica.
* Exportación a Excel para análisis posterior o presentaciones.

### Reporte de Ventas realizadas

**Objetivo:** Este reporte proporciona un resumen detallado de todas las ventas realizadas dentro del rango de fechas seleccionado.

**Estructura:**

El reporte incluye las siguientes columnas:

1. FechaRegistro
2. TipoDocumento
3. NumeroDocumento
4. MontoTotal
5. UsuarioRegistro
6. CedulaCliente
7. NombreCliente
8. CodigoProducto
9. NombreProducto
10. Categoria
11. PrecioVenta
12. Cantidad
13. SubTotal
14. Descuento

**Filtros y Parámetros:** El reporte permite filtrar por rango de fechas.

**Formatos de Salida:** El reporte se puede visualizar en pantalla y exportar a Excel (.xlsx).

#### Diseño de la ventana Reportes Ventas

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Descripción General:**

Esta pantalla permite al usuario generar y visualizar los reportes de ventas.

**Elementos Clave:**

1. Campos para seleccionar el rango de fechas (Fecha Inicio y Fecha Fin).
2. Botón "Buscar" para generar el reporte.
3. Tabla (DataGridView) para visualizar los resultados del reporte.
4. Combobox para seleccionar la columna por la cual filtrar los resultados.
5. Campo de texto para ingresar el criterio de búsqueda.
6. Botones adicionales:
   * "Generar Excel" para exportar el reporte a formato Excel.
   * "Limpiar" para resetear la búsqueda.

**Funcionalidades:**

* Generación dinámica de reportes basados en el rango de fechas seleccionado.
* Visualización detallada de los datos en una tabla.
* Filtro adicional en la tabla para buscar información específica.
* Exportación a Excel para análisis posterior o presentaciones.

## Diseño de pantallas

El sistema cuenta con varias pantallas principales que facilitan la interacción del usuario con la aplicación. Estas pantallas están diseñadas para ser intuitivas y eficientes, permitiendo al usuario realizar diversas tareas relacionadas con la gestión de compras y ventas.

Para el sistema de ventas, hemos clasificado las pantallas de la siguiente manera:

1. Pantalla de Inicio
   * Pantalla de Login
   * Pantalla de Recuperar Clave
   * Pantalla Menu
2. Pantalla de Gestión
   * Pantalla de Producto
   * Pantalla de Categoria
   * Pantalla de Unidad de Medida
   * Pantalla de Inventario
   * Pantalla de Oferta
3. Pantalla de Consulta
   * Pantalla de Usuario
   * Pantalla de Proveedor
   * Pantalla de Transportista
   * Pantalla de Cliente
   * Pantalla de Sucursal
4. Pantallas Transaccionales
   * Pantalla Compra
   * Pantalla Detalle Compra
   * Pantalla Venta
   * Pantalla Detalle Venta
5. Pantalla de Configuracion
   * Pantalla Informacion Negocio
   * Pantalla Informacion Programa
6. Pantallas de Informes y Analítica
   * Pantalla Reporte Compra
   * Pantalla Grafica Compra
   * Pantalla Reporte Venta
   * Pantalla Grafica Venta
7. Pantallas Modales
   * Pantalla Modal Producto
   * Pantalla Modal Proveedor
   * Pantalla Modal Transportista
   * Pantalla Modal Cliente

### Pantalla de Inicio

#### Diseño de la ventana Login

Diagrama

Descripción generada automáticamente

**Descripción General:** Esta pantalla es el punto de entrada principal para los usuarios del sistema de ventas del supermercado. Permite la autenticación de los empleados antes de acceder a las funcionalidades principales.

**Elementos Clave:**

* Campo para ingresar el código de usuario
* Campo para ingresar la clave
* Botón para mostrar/ocultar la contraseña
* Botón para iniciar sesión
* Enlace para recuperar la contraseña olvidada
* Botón para salir de la aplicación
* Mostrador de hora actual

**Funcionalidades:**

* Autenticación de usuarios mediante código y clave
* Visualización de la hora actual
* Opción para mostrar/ocultar la contraseña durante el ingreso
* Enlace a la función de recuperación de contraseña
* Posibilidad de salir de la aplicación desde esta pantalla

**Consideraciones de Diseño:**

* La pantalla sigue un diseño minimalista y centrado, enfocado en la tarea principal de autenticación.
* Se utiliza un esquema de colores coherente con la marca del supermercado.
* Los campos de entrada son grandes y fáciles de leer.
* El botón de iniciar sesión es prominente y fácil de identificar.

#### Diseño de la ventana Recuperar Clave

Diagrama

Descripción generada automáticamente

**Descripción General:** Esta pantalla auxiliar permite a los usuarios recuperar su contraseña olvidada, mejorando la experiencia del usuario y reduciendo la necesidad de intervención del departamento de TI.

**Elementos Clave:**

* Campo para ingresar el correo electrónico asociado a la cuenta
* Botón para enviar instrucciones de recuperación
* Botón para regresar a la pantalla de login

**Funcionalidades:**

* Envío de instrucciones de recuperación al correo electrónico del usuario
* Posibilidad de regresar directamente a la pantalla de login

**Consideraciones de Diseño:**

* La pantalla mantiene el estilo visual coherente con la pantalla de login.
* El proceso es simple y requiere solo una pieza de información del usuario.
* Se proporciona una opción clara para regresar al login si el usuario cambia de opinión.

#### Diseño de la ventana Menú

Diagrama

Descripción generada automáticamente

**Descripción General:** Esta pantalla actúa como el menú principal del sistema, proporcionando acceso a todas las funcionalidades disponibles para el usuario logueado. Sirve como punto central de navegación y control de acceso basado en permisos.

**Elementos Clave:**

* Menú principal con opciones organizadas jerárquicamente
* Panel de contenido para mostrar formularios secundarios
* Etiqueta mostrando el nombre completo del usuario logueado
* Mostrador de fecha y hora actual
* Botón para salir del sistema

**Funcionalidades:**

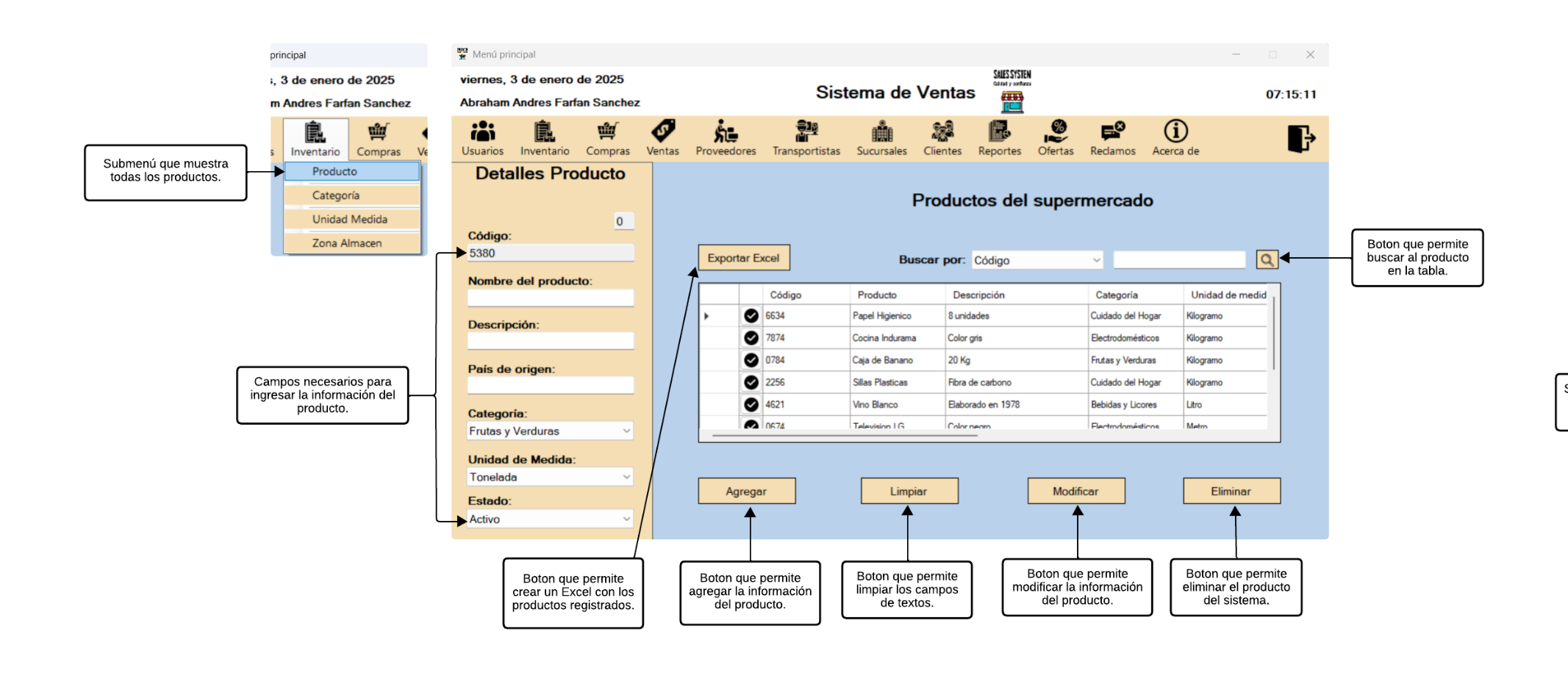
1. Navegación entre diferentes módulos del sistema
2. Control de acceso basado en permisos de usuario
3. Visualización dinámica de formularios secundarios dentro del panel principal
4. Mostrador en tiempo real de fecha y hora
5. Opción para cerrar sesión o salir completamente del sistema

**Consideraciones de Diseño:**

1. Organización Jerárquica: El menú está estructurado de manera jerárquica para facilitar el acceso rápido a las diferentes funcionalidades del sistema.
2. Control de Acceso: Se implementa un sistema de permisos dinámicos, donde los elementos del menú se ocultan o muestran según los derechos del usuario actual.
3. Contenedor Dinámico: El panel central utiliza una técnica de contenedores para mostrar diferentes formularios sin necesidad de abrir ventanas separadas, mejorando la eficiencia y reduciendo la complejidad visual.
4. Información Contextual: Se muestra el nombre completo del usuario logueado, proporcionando contexto sobre la sesión activa.
5. Elementos de Tiempo Real: La pantalla incluye un mostrador de fecha y hora actualizado constantemente, lo cual puede ser útil para operaciones sensibles al tiempo.
6. Opción de Salida Segura: Se proporciona una opción explícita para salir del sistema, con una confirmación adicional para evitar salidas accidentales.
7. Flexibilidad: La estructura permite fácilmente agregar o modificar opciones de menú sin alterar significativamente la interfaz principal.

### Pantalla de Gestión

#### Diseño de la ventana Producto



**Descripción General:** Esta pantalla permite gestionar los productos del sistema, incluyendo operaciones como agregar, modificar y eliminar productos. Proporciona una interfaz completa para visualizar, filtrar y exportar información de productos.

**Elementos Clave:**

* Tabla para mostrar lista de productos
* Campos de texto para ingresar/editar detalles del producto
* ComboBoxes para seleccionar categoría, unidad de medida y estado
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para exportar a Excel

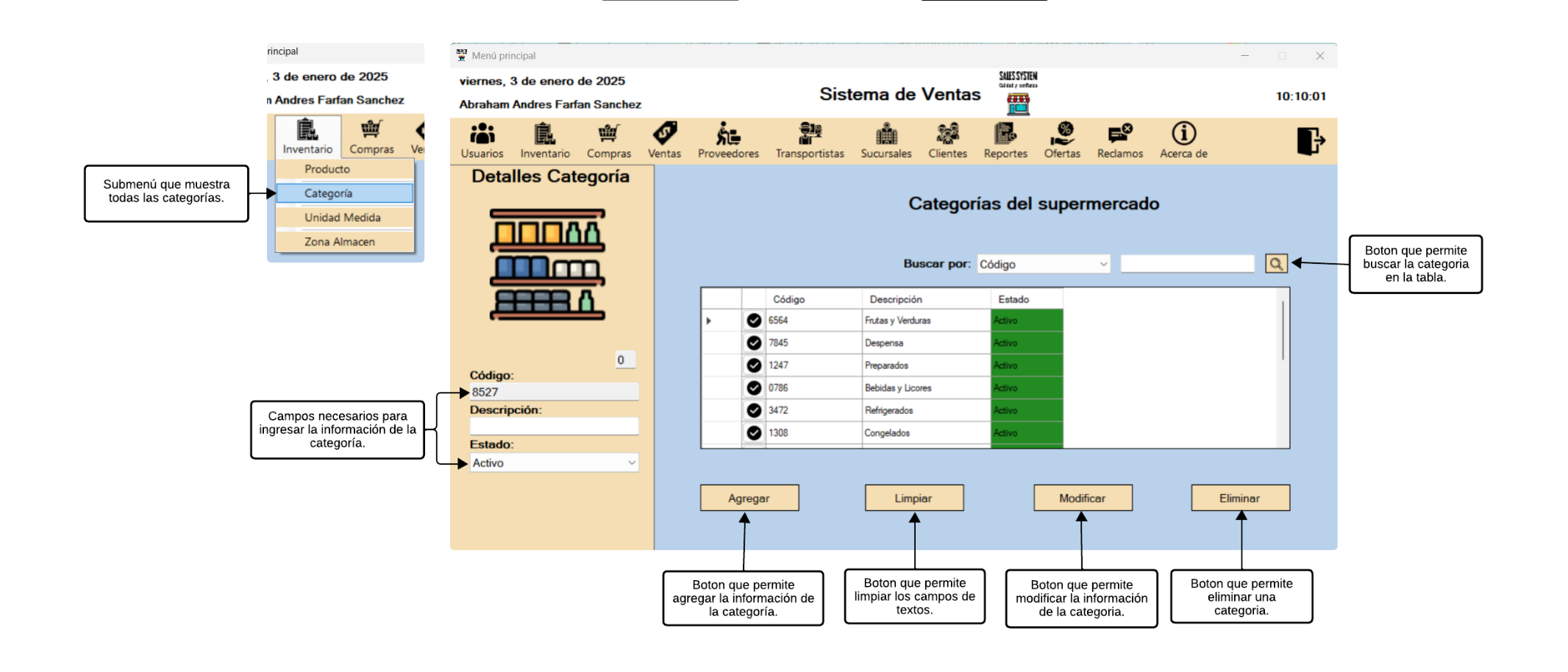
**Funcionalidades:**

1. Listar productos existentes en una tabla interactiva
2. Agregar nuevos productos al sistema
3. Modificar información de productos existentes
4. Eliminar productos del sistema
5. Filtrar productos por diferentes criterios
6. Exportar lista de productos a archivo Excel
7. Generar códigos automáticos para los productos
8. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño:**

1. Interfaz Intuitiva: La pantalla está diseñada para ser fácil de usar, con botones claros para cada acción principal y campos etiquetados para la entrada de datos.
2. Validación de Datos: Se implementan controles para asegurar que los campos obligatorios sean llenados antes de permitir la creación o modificación de un producto.
3. Manejo de Combobox Dinámicos: Los combobox para categoría y unidad de medida se cargan dinámicamente desde la base de datos, permitiendo mantener la consistencia de datos.
4. Generación Automática de Código: Se implementa un método para generar códigos de producto automáticamente, evitando duplicados y simplificando el proceso de ingreso.
5. Funcionalidad de Búsqueda Avanzada: Permite filtrar productos por diferentes columnas, mejorando la eficiencia al buscar información específica.
6. Exportación a Excel: Incluye la opción de exportar la lista de productos a Excel, facilitando la generación de reportes y análisis externos.
7. Manejo de Estados: Utiliza un combobox para manejar el estado del producto (activo/no activo), permitiendo una fácil desactivación temporal de productos.
8. Interfaz Responsiva: La tabla de productos utiliza colores condicionales para representar el estado de los productos, mejorando la visualización y comprensión rápida de datos.
9. Seguridad: Implementa una confirmación antes de eliminar un producto, previniendo eliminaciones accidentales.

#### Diseño de la ventana Categoría



**Descripción General:** Esta pantalla permite gestionar las categorías del sistema, permitiendo agregar, modificar y eliminar categorías. Ofrece una interfaz para visualizar, filtrar y editar información de categorías existentes.

**Elementos Clave:**

* Tabla para mostrar lista de categorías
* Campos de texto para ingresar/editar detalles de la categoría
* ComboBox para seleccionar el estado de la categoría
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

**Funcionalidades:**

1. Listar categorías existentes en una tabla interactiva
2. Agregar nuevas categorías al sistema
3. Modificar información de categorías existentes
4. Eliminar categorías del sistema
5. Filtrar categorías por diferentes criterios
6. Generar códigos automáticos para las categorías
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño:**

1. Interfaz Intuitiva: La pantalla está diseñada para ser fácil de usar, con botones claros para cada acción principal y campos etiquetados para la entrada de datos.
2. Validación de Datos: Se implementan controles para asegurar que los campos obligatorios sean llenados antes de permitir la creación o modificación de una categoría.
3. Manejo de Combobox Dinámico: El combobox para el estado se carga con opciones predefinidas (activo/no activo), facilitando la selección rápida.
4. Generación Automática de Código: Se implementa un método para generar códigos de categoría automáticamente, evitando duplicados y simplificando el proceso de ingreso.
5. Funcionalidad de Búsqueda Avanzada: Permite filtrar categorías por diferentes columnas, mejorando la eficiencia al buscar información específica.
6. Colorización de Estado: Utiliza colores condicionales para representar el estado de las categorías en la tabla, mejorando la visualización y comprensión rápida de datos.
7. Seguridad: Implementa una confirmación antes de eliminar una categoría, previniendo eliminaciones accidentales.
8. Manejo de Selección: Incluye una columna adicional como botón para seleccionar rápidamente una categoría específica en la tabla.

#### Diseño de la ventana Unidad de Medida

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

**Descripción General:** Esta pantalla permite gestionar las unidades de medida del sistema, permitiendo agregar, modificar y eliminar unidades de medida. Ofrece una interfaz para visualizar, filtrar y editar información de unidades de medida existentes.

**Elementos Clave:**

* Tabla para mostrar lista de unidades de medida
* Campos de texto para ingresar/editar detalles de la unidad de medida
* ComboBox para seleccionar el estado de la unidad de medida
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

**Funcionalidades:**

1. Listar unidades de medida existentes en una tabla interactiva
2. Agregar nuevas unidades de medida al sistema
3. Modificar información de unidades de medida existentes
4. Eliminar unidades de medida del sistema
5. Filtrar unidades de medida por diferentes criterios
6. Generar códigos automáticos para las unidades de medida
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño:**

1. Interfaz Intuitiva: La pantalla está diseñada para ser fácil de usar, con botones claros para cada acción principal y campos etiquetados para la entrada de datos.
2. Validación de Datos: Se implementan controles para asegurar que los campos obligatorios sean llenados antes de permitir la creación o modificación de una unidad de medida.
3. Manejo de Combobox Dinámico: El combobox para el estado se carga con opciones predefinidas (activo/no activo), facilitando la selección rápida.
4. Generación Automática de Código: Se implementa un método para generar códigos de unidad de medida automáticamente, evitando duplicados y simplificando el proceso de ingreso.
5. Funcionalidad de Búsqueda Avanzada: Permite filtrar unidades de medida por diferentes columnas, mejorando la eficiencia al buscar información específica.
6. Colorización de Estado: Utiliza colores condicionales para representar el estado de las unidades de medida en la tabla, mejorando la visualización y comprensión rápida de datos.
7. Seguridad: Implementa una confirmación antes de eliminar una unidad de medida, previniendo eliminaciones accidentales.
8. Manejo de Selección: Incluye una columna adicional como botón para seleccionar rápidamente una unidad de medida específica en la tabla.

#### Diseño de la ventana Zona de Almacén

#### Diseño de la ventana Oferta

Interfaz de usuario gráfica

Descripción generada automáticamente

**Descripción General:** Esta pantalla permite gestionar las ofertas del sistema, permitiendo agregar, modificar y eliminar ofertas. Ofrece una interfaz para visualizar, filtrar y editar información de ofertas existentes.

**Elementos Clave:**

* Tabla para mostrar lista de ofertas
* Campos de texto para ingresar/editar detalles de la oferta
* ComboBox para seleccionar el producto y el estado de la oferta
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

**Funcionalidades:**

1. Listar ofertas existentes en una tabla interactiva
2. Agregar nuevas ofertas al sistema
3. Modificar información de ofertas existentes
4. Eliminar ofertas del sistema
5. Filtrar ofertas por diferentes criterios
6. Generar códigos automáticos para las ofertas
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño:**

1. Interfaz Intuitiva: La pantalla está diseñada para ser fácil de usar, con botones claros para cada acción principal y campos etiquetados para la entrada de datos.
2. Validación de Datos: Se implementan controles para asegurar que los campos obligatorios sean llenados antes de permitir la creación o modificación de una oferta.
3. Manejo de Combobox Dinámico: Los comboboxes para el producto y el estado se cargan con opciones predefinidas, facilitando la selección rápida.
4. Generación Automática de Código: Se implementa un método para generar códigos de oferta automáticamente, evitando duplicados y simplificando el proceso de ingreso.
5. Funcionalidad de Búsqueda Avanzada: Permite filtrar ofertas por diferentes columnas, mejorando la eficiencia al buscar información específica.
6. Colorización de Estado: Utiliza colores condicionales para representar el estado de las ofertas en la tabla, mejorando la visualización y comprensión rápida de datos.
7. Seguridad: Implementa una confirmación antes de eliminar una oferta, previniendo eliminaciones accidentales.
8. Manejo de Selección: Incluye una columna adicional como botón para seleccionar rápidamente una oferta específica en la tabla.
9. Validación de Producto Activo: Verifica si el producto seleccionado está activo antes de proceder con la operación.
10. Gestión de Fechas: Utiliza controles de fecha personalizados para manejar fechas de inicio y fin de la oferta.

### Pantalla de Consulta

#### Diseño de la ventana Usuario

Interfaz de usuario gráfica

Descripción generada automáticamente

**Descripción General:** Esta pantalla permite gestionar los usuarios del sistema, permitiendo agregar, modificar y eliminar usuarios. Ofrece una interfaz para visualizar, filtrar y editar información de usuarios existentes.

**Elementos Clave:**

* Tabla para mostrar lista de los usuarios
* Campos de texto para ingresar/editar detalles del usuario
* ComboBox para seleccionar el rol del usuario
* ComboBox para seleccionar el estado del usuario
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

**Funcionalidades:**

1. Listar usuarios existentes en una tabla interactiva
2. Agregar nuevos usuarios al sistema
3. Modificar información de los usuarios existentes
4. Eliminar usuarios del sistema
5. Filtrar usuarios por diferentes criterios
6. Generar códigos automáticos para los usuarios
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Colorización Visual: Se aplica colorización condicional a las celdas de estado en la tabla de usuarios.
4. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de texto.
5. Automatización: Se genera un código automático para nuevos usuarios.
6. Flexibilidad: El sistema permite agregar, modificar y eliminar múltiples usuarios.
7. Seguridad: Se solicita confirmación antes de eliminar un usuario.
8. Información Adicional: Se muestra información relevante sobre el usuario mientras se busca en la tabla.
9. Flexibilidad de Filtros: Se permite filtrar la lista de usuarios por diferentes columnas.
10. Manejo de Selección: Se implementa un botón de selección para ver detalles de un usuario específico.

#### Diseño de la ventana Proveedor

Interfaz de usuario gráfica

Descripción generada automáticamente

**Descripción General:** Esta pantalla permite gestionar los proveedores del sistema, permitiendo agregar, modificar y eliminar proveedores. Ofrece una interfaz para visualizar, filtrar y editar información de proveedores existentes.

**Elementos Clave:**

* Tabla para mostrar lista de los proveedores
* Campos de texto para ingresar/editar detalles de los proveedores
* ComboBox para seleccionar el estado de los proveedores
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

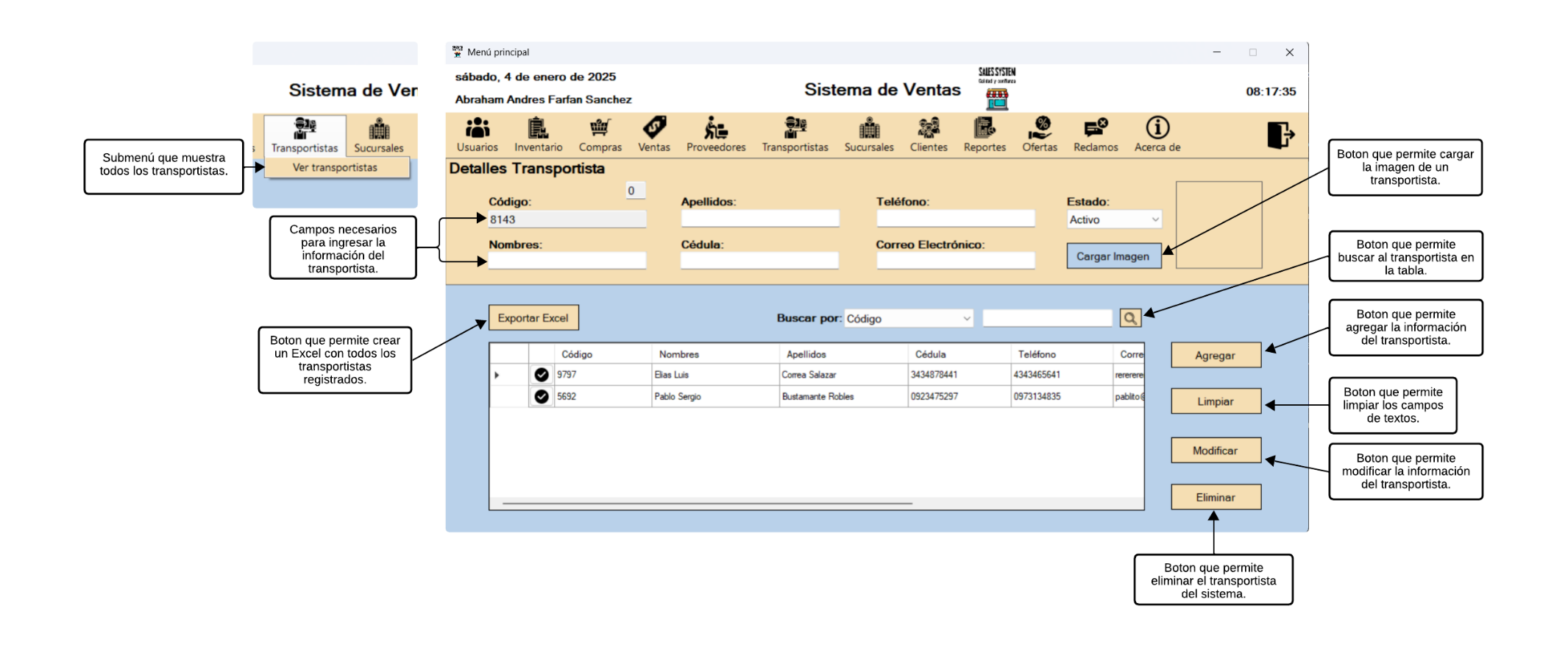
**Funcionalidades:**

1. Listar proveedores existentes en una tabla interactiva
2. Agregar nuevos proveedores al sistema
3. Modificar información de los proveedores existentes
4. Eliminar proveedores del sistema
5. Filtrar proveedores por diferentes criterios
6. Generar códigos automáticos para los proveedores
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Colorización Visual: Se aplica colorización condicional a las celdas de estado en la tabla de los proveedores.
4. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de texto.
5. Automatización: Se genera un código automático para nuevos proveedores.
6. Flexibilidad: El sistema permite agregar, modificar y eliminar múltiples proveedores.
7. Seguridad: Se solicita confirmación antes de eliminar un proveedor.
8. Información Adicional: Se muestra información relevante sobre el proveedor mientras se busca en la tabla.
9. Flexibilidad de Filtros: Se permite filtrar la lista de proveedores por diferentes columnas.
10. Manejo de Selección: Se implementa un botón de selección para ver detalles de un proveedor específica.

#### Diseño de la ventana Transportista



**Descripción General:** Esta pantalla permite gestionar los transportistas del sistema, permitiendo agregar, modificar y eliminar transportistas. Ofrece una interfaz para visualizar, filtrar y editar información de transportistas existentes.

**Elementos Clave:**

* Tabla para mostrar lista de los transportistas
* Campos de texto para ingresar/editar detalles del transportista
* ComboBox para seleccionar el estado del transportista
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

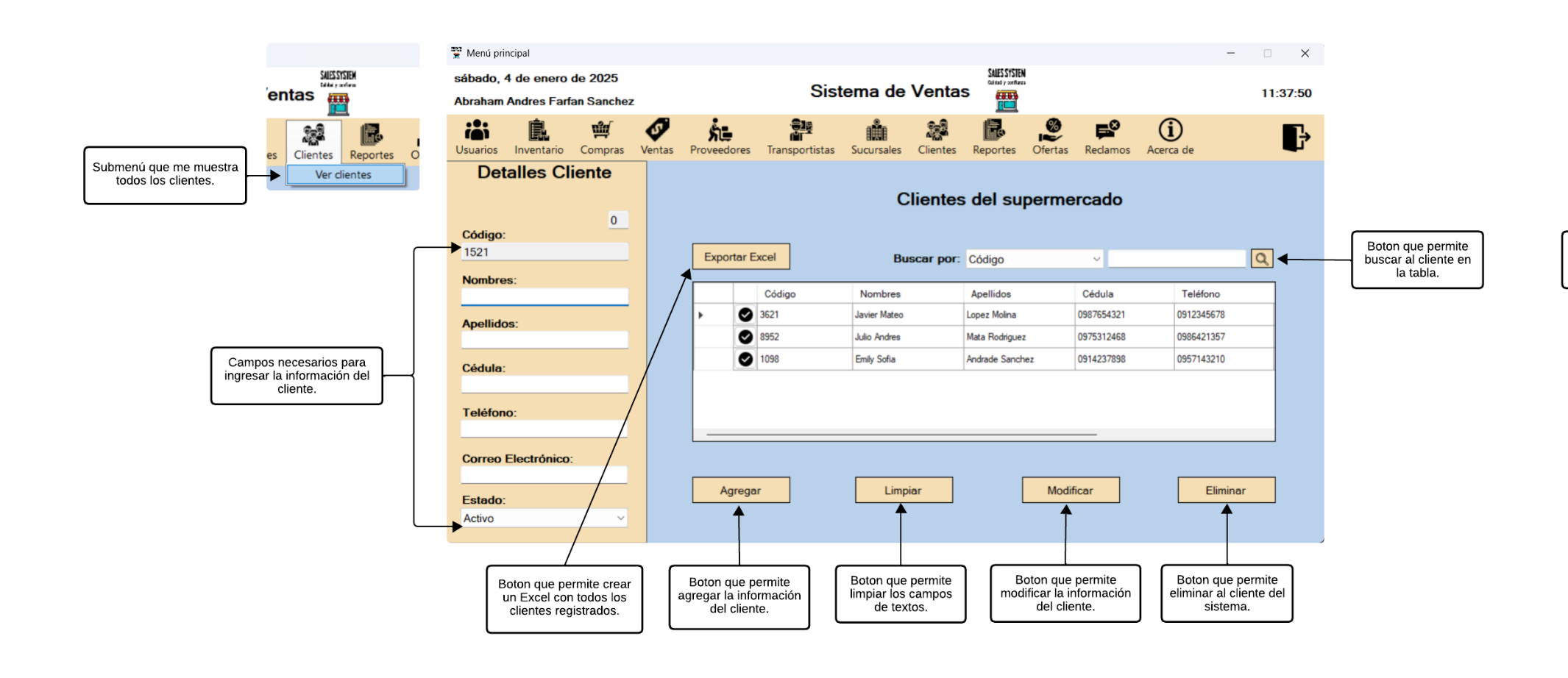
**Funcionalidades:**

1. Listar transportistas existentes en una tabla interactiva
2. Agregar nuevos transportistas al sistema
3. Modificar información de los transportistas existentes
4. Eliminar transportistas del sistema
5. Filtrar transportistas por diferentes criterios
6. Generar códigos automáticos para los transportistas
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Colorización Visual: Se aplica colorización condicional a las celdas de estado en la tabla de los transportistas.
4. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de texto.
5. Automatización: Se genera un código automático para nuevos transportistas.
6. Flexibilidad: El sistema permite agregar, modificar y eliminar múltiples transportistas.
7. Seguridad: Se solicita confirmación antes de eliminar un transportista.
8. Información Adicional: Se muestra información relevante sobre el transportista mientras se busca en la tabla.
9. Flexibilidad de Filtros: Se permite filtrar la lista de transportistas por diferentes columnas.
10. Manejo de Selección: Se implementa un botón de selección para ver detalles de un transportista específico.

#### Diseño de la ventana Cliente



**Descripción General:** Esta pantalla permite gestionar los clientes del sistema, permitiendo agregar, modificar y eliminar clientes. Ofrece una interfaz para visualizar, filtrar y editar información de clientes existentes.

**Elementos Clave:**

* Tabla para mostrar lista de los clientes
* Campos de texto para ingresar/editar detalles del cliente
* ComboBox para seleccionar el estado del cliente
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

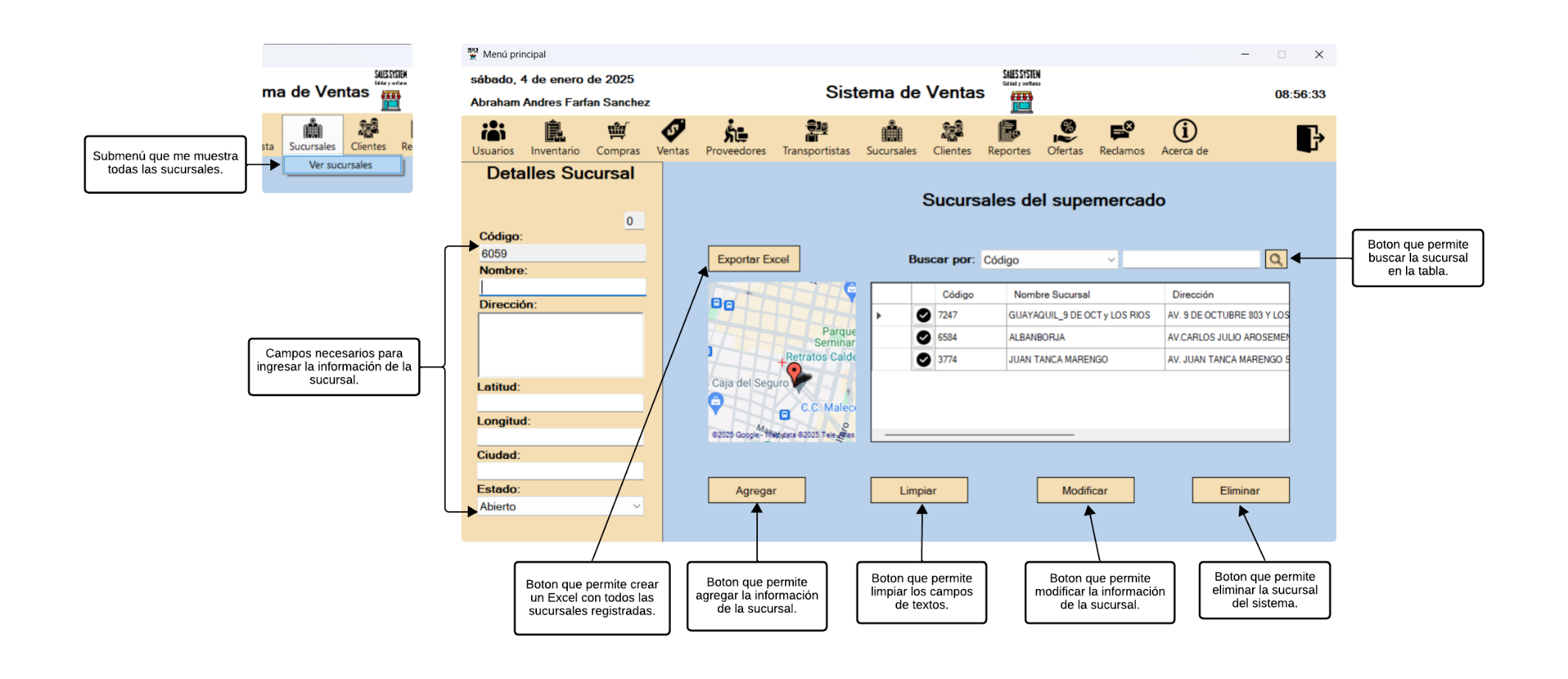
**Funcionalidades:**

1. Listar clientes existentes en una tabla interactiva
2. Agregar nuevos clientes al sistema
3. Modificar información de los clientes existentes
4. Eliminar clientes del sistema
5. Filtrar clientes por diferentes criterios
6. Generar códigos automáticos para los transportistas
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Colorización Visual: Se aplica colorización condicional a las celdas de estado en la tabla de los clientes.
4. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de texto.
5. Automatización: Se genera un código automático para nuevos clientes.
6. Flexibilidad: El sistema permite agregar, modificar y eliminar múltiples clientes.
7. Seguridad: Se solicita confirmación antes de eliminar un cliente.
8. Información Adicional: Se muestra información relevante sobre el cliente mientras se busca en la tabla.
9. Flexibilidad de Filtros: Se permite filtrar la lista de clientes por diferentes columnas.
10. Manejo de Selección: Se implementa un botón de selección para ver detalles de un cliente específico.

#### Diseño de la ventana Sucursal



**Descripción General:** Esta pantalla permite gestionar las sucursales del sistema, permitiendo agregar, modificar y eliminar sucursales. Ofrece una interfaz para visualizar, filtrar y editar información de sucursales existentes.

**Elementos Clave:**

* Tabla para mostrar lista de las sucursales
* Mapa utilizando GMap.NET para visualización geográfica.
* Campos de texto para ingresar/editar detalles de la sucursal
* ComboBox para seleccionar el estado de la sucursal
* Botones para acciones CRUD (Crear, Leer, Actualizar, Eliminar)
* Campo de búsqueda y filtro
* Opción para generar códigos automáticos

**Funcionalidades:**

1. Listar sucursales existentes en una tabla interactiva
2. Agregar nuevas sucursales al sistema
3. Modificar información de las sucursales existentes
4. Eliminar sucursales del sistema
5. Filtrar sucursales por diferentes criterios
6. Generar códigos automáticos para las sucursales
7. Validación de campos obligatorios antes de guardar cambios

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Colorización Visual: Se aplica colorización condicional a las celdas de estado en la tabla de los clientes.
4. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de texto.
5. Automatización: Se genera un código automático para nuevas sucursales.
6. Flexibilidad: El sistema permite agregar, modificar y eliminar múltiples sucursales.
7. Seguridad: Se solicita confirmación antes de eliminar una sucursal.
8. Información Adicional: Se muestra información relevante sobre la sucursal mientras se busca en la tabla.
9. Flexibilidad de Filtros: Se permite filtrar la lista de sucursales por diferentes columnas.
10. Manejo de Selección: Se implementa un botón de selección para ver detalles de una sucursal específica.

### Pantalla Transaccionales

#### Diseño de la ventana Compra

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Descripción General:** Esta pantalla contiene un formulario para gestionar compras en un sistema de inventario. Esta forma permite a los usuarios realizar compras, buscar productos y proveedores, y registrar detalles de la transacción.

**Elementos Clave**

1. ComboBox para seleccionar el tipo de documento (Boleta o Factura).
2. Campos para buscar y mostrar información del proveedor.
3. Campos para buscar y mostrar información del producto y sus precios.
4. Campos para buscar y mostrar información del transportista.
5. Tabla para mostrar los detalles de la compra.
6. Control numérico para ingresar la cantidad de productos.

**Funcionalidades**

1. Buscar y seleccionar proveedores, productos y transportistas.
2. Agregar productos a la tabla de compras.
3. Calcular el total de la compra.
4. Registrar la compra con detalles en la base de datos.
5. Eliminar productos de la tabla de compras.
6. Validar entradas de datos (formato de números y validación de campos obligatorios).
7. Generar número correlativo automático para el documento de compra.
8. Copiar el número de compra al portapapeles.
9. Sumar el stock de los productos comprados.
10. Limpiar los campos de texto.

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Búsqueda Asistida: Se utiliza una ventana modal para buscar proveedores, productos y transportistas, mejorando la experiencia del usuario.
4. Colorización Visual: Se aplica coloración condicional a los campos para indicar estados (por ejemplo, cambio de fondo para campos con errores).
5. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de precio, permitiendo solo dígitos y coma.
6. Automatización: Se genera un número correlativo automático para el documento de compra.
7. Flexibilidad: El sistema permite agregar múltiples productos a la compra y calcular el total automáticamente a medida que se añaden o eliminan productos.
8. Seguridad: Se verifica la existencia del producto antes de agregarlo a la compra.
9. Personalización de Documentos: Permite elegir entre diferentes tipos de documentos (Boleta o Factura).

#### Diseño de la ventana Detalle Compra

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Descripción General:** Esta pantalla contiene un formulario para visualizar los detalles de una compra específica en un sistema de gestión de inventario. Esta forma permite a los usuarios buscar y visualizar información detallada de una compra, incluyendo los productos adquiridos, proveedores, transportistas y otros datos relevantes.

**Elementos Clave**

1. Campo para ingresar el número de código de la compra.
2. Botón para buscar la compra correspondiente al código ingresado.
3. Campos para mostrar información general de la compra como: Fecha compra, Tipo de documento, Nombre completo del usuario, cédula del proveedor, nombres del proveedor, cédula del transportista, nombres del transportista.
4. Tabla para mostrar los detalles de los productos comprados.
5. Campo para mostrar el monto total de la compra.
6. Botón para limpiar todos los campos.
7. Botón para generar un reporte PDF de la compra.

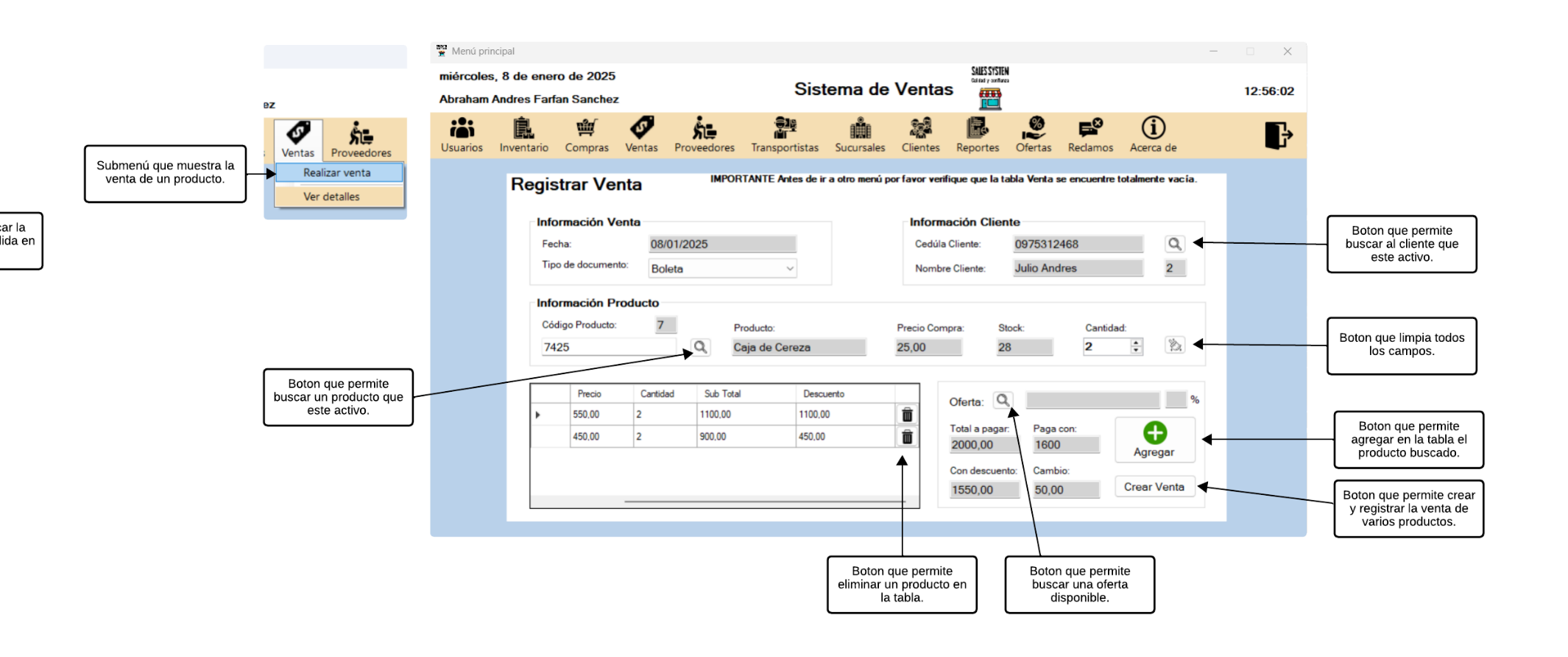
**Funcionalidades**

1. Buscar y visualizar detalles de una compra específica utilizando su código.
2. Mostrar información general de la compra, incluyendo fecha, tipo de documento, usuario que realizó la compra, proveedor y transportista.
3. Visualizar los detalles de los productos comprados, incluyendo nombre, precio, cantidad y subtotal.
4. Mostrar el monto total de la compra.
5. Limpiar todos los campos para buscar una nueva compra.
6. Generar un reporte PDF de la compra, incluyendo información del negocio y detalles de la transacción.
7. Validar la entrada del código de compra para asegurar que solo se ingresen números.

**Consideraciones de Diseño**

1. Interfaz Intuitiva: La interfaz está diseñada para ser fácil de usar, con campos descriptivos y botones claros para cada función principal.
2. Visualización Detallada: Se proporciona una vista completa de la compra, incluyendo información general y detalles de los productos.
3. Flexibilidad de Búsqueda: Permite buscar una compra específica utilizando su código único.
4. Validación de Entradas: Se implementa una validación para asegurar que el código de compra solo contenga números.
5. Capacidad de Limpieza: Incluye una función para limpiar todos los campos, facilitando la búsqueda de nuevas compras.
6. Uso de Recursos Embebidos: Utiliza recursos embebidos (como plantillas HTML) para generar el reporte PDF, lo que mejora la consistencia y facilidad de mantenimiento.
7. Generación de Reportes: Ofrece la capacidad de generar un reporte PDF profesional, incluyendo el logo del negocio y formato personalizado.

#### Diseño de la ventana Venta



**Descripción General:** Esta pantalla contiene un formulario para realizar ventas en un sistema de gestión de inventario. Esta forma permite a los usuarios registrar ventas, incluyendo la selección de clientes, productos, ofertas, y el proceso de pago. También maneja la generación de documentos de venta y el cálculo de totales y cambios.

**Elementos Clave**

1. ComboBox para seleccionar el tipo de documento (Boleta o Factura).
2. Campos para buscar y mostrar información del proveedor.
3. Campos para buscar y mostrar información del producto y sus precios.
4. Campos para buscar y mostrar información del transportista.
5. Tabla para mostrar los detalles de la compra.
6. Control numérico para ingresar la cantidad de productos.

**Funcionalidades**

1. Buscar y seleccionar clientes para la venta.
2. Buscar y seleccionar productos para agregar a la venta.
3. Aplicar ofertas a los productos seleccionados.
4. Agregar productos a la venta, calculando subtotales y descuentos.
5. Eliminar productos de la tabla de ventas.
6. Calcular el total de la venta y el monto a pagar.
7. Manejar el proceso de pago y calcular el cambio.
8. Registrar la venta completa en el sistema.
9. Generar un número de documento único para la venta.
10. Permitir copiar el número de venta al portapapeles.
11. Validar entradas de datos para asegurar la integridad de la información.
12. Restar el stock de los productos vendidos.
13. Limpiar los campos de texto.

**Consideraciones de Diseño**

1. Interfaz Usable: La interfaz está diseñada para ser intuitiva, con campos descriptivos y botones claros para cada función principal.
2. Validación de Entradas: Se implementan controles de entrada para garantizar que los campos obligatorios sean llenados correctamente.
3. Búsqueda Asistida: Se utiliza una ventana modal para buscar clientes, productos y ofertas, mejorando la experiencia del usuario.
4. Colorización Visual: Se aplica coloración condicional a los campos para indicar estados (por ejemplo, cambio de fondo para campos con errores).
5. Personalización de Teclado: Se limita el input de caracteres especiales en los campos de precio, permitiendo solo dígitos y coma.
6. Automatización: Se genera un número correlativo automático para el documento de venta.
7. Flexibilidad: El sistema permite agregar múltiples productos a la venta y calcular el total automáticamente a medida que se añaden o eliminan productos.
8. Personalización de Documentos: Permite elegir entre diferentes tipos de documentos (Boleta o Factura).
9. Manejo de Pago Completo: Incluye cálculo de cambio y manejo de diferentes formas de pago.

#### Diseño de la ventana Detalle Venta

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Descripción General:** Esta pantalla contiene un formulario para visualizar los detalles de una venta específica en un sistema de gestión de inventario. Esta forma permite a los usuarios buscar y visualizar información detallada de una venta, incluyendo los productos adquiridos, clientes, ofertas y otros datos relevantes.

**Elementos Clave**

1. Campo para ingresar el número de código de la venta.
2. Botón para buscar la venta correspondiente al código ingresado.
3. Campos para mostrar información general de la venta como: Fecha compra, Tipo de documento, Nombre completo del usuario, cédula del cliente, nombres del cliente, código del transportista, nombres del transportista.
4. Tabla para mostrar los detalles de los productos vendidos.
5. Campo para mostrar el monto total de la venta, descuento, monto pago y monto cambio.
6. Botón para limpiar todos los campos.
7. Botón para generar un reporte PDF de la venta.

**Funcionalidades**

1. Buscar y visualizar detalles de una compra específica utilizando su código.
2. Mostrar información general de la compra, incluyendo fecha, tipo de documento, usuario que realizó la compra y cliente.
3. Visualizar los detalles de los productos vendidos, incluyendo nombre, precio, cantidad y subtotal.
4. Mostrar el monto total de la compra.
5. Limpiar todos los campos para buscar una nueva compra.
6. Generar un reporte PDF de la compra, incluyendo información del negocio y detalles de la transacción.
7. Validar la entrada del código de compra para asegurar que solo se ingresen números.

**Consideraciones de Diseño**

1. Interfaz Intuitiva: La interfaz está diseñada para ser fácil de usar, con campos descriptivos y botones claros para cada función principal.
2. Visualización Detallada: Se proporciona una vista completa de la venta, incluyendo información general y detalles de los productos.
3. Flexibilidad de Búsqueda: Permite buscar una venta específica utilizando su código único.
4. Validación de Entradas: Se implementa una validación para asegurar que el código de venta solo contenga números.
5. Capacidad de Limpieza: Incluye una función para limpiar todos los campos, facilitando la búsqueda de nuevas ventas.
6. Uso de Recursos Embebidos: Utiliza recursos embebidos (como plantillas HTML) para generar el reporte PDF, lo que mejora la consistencia y facilidad de mantenimiento.
7. Generación de Reportes: Ofrece la capacidad de generar un reporte PDF profesional, incluyendo el logo del negocio y formato personalizado.

### Pantalla de Configuración

#### Diseño de la ventana Información Negocio

**Descripción General:** La pantalla permite gestionar la información del negocio en un sistema de gestión empresarial. Esta forma permite a los administradores ver los detalles actuales del negocio y realizar modificaciones cuando sea necesario.

**Elementos Clave**

1. Campos para mostrar y editar la información del negocio.
2. Botón para cargar una nueva imagen del logo del negocio.
3. Botón para guardar los cambios realizados en la información del negocio.

**Funcionalidades**

1. Mostrar la información actual del negocio al cargar la ventana.
2. Visualizar el logo del negocio.
3. Cargar una nueva imagen para el logo del negocio.
4. Editar los campos de información del negocio (nombre, teléfono, RUC, dirección, correo electrónico).
5. Guardar los cambios realizados en la información del negocio.
6. Conversión de bytes a imagen para mostrar y manipular el logo del negocio.

**Consideraciones de Diseño**

1. Interfaz Intuitiva: La pantalla está diseñada para ser fácil de usar, con botones claros para cada acción principal y campos etiquetados para la entrada de datos.
2. Validación de Datos: Se implementan controles para asegurar que los campos obligatorios sean llenados antes de permitir la creación o modificación de un producto.

#### Diseño de la ventana Información Programa

Interfaz de usuario gráfica

Descripción generada automáticamente

**Descripción General:** Esta pantalla muestra información sobre el programa actual y proporciona un enlace a los proyectos relacionados en GitHub. Esta forma es típicamente utilizada para dar créditos, mostrar licencias o proporcionar información adicional sobre la aplicación.

**Elementos Clave**

1. Enlace que contiene el enlace la documentación del proyecto de GitHub.

**Funcionalidades**

1. Mostrar información general sobre el programa.
2. Proporcionar un enlace ciclable a los proyectos de GitHub.
3. Abrir el enlace en el navegador predeterminado del sistema cuando se hace clic en él.
4. Manejar posibles errores al intentar abrir el enlace.

**Consideraciones de Diseño**

1. Minimalismo: La clase tiene un propósito específico y limitado, enfocándose principalmente en mostrar información y proporcionar un enlace.
2. anejo de Enlaces: Implementa la funcionalidad para abrir URLs externas directamente desde la aplicación.
3. Enfoque en la Usabilidad: Al hacer que el enlace sea ciclable, se facilita la navegación para los usuarios interesados en explorar más sobre el desarrollo del software.

### Pantalla de Informe y Analítica

#### Diseño de la ventana Graficas Compras



**Descripción General:** Esta pantalla muestra una gráfica de compras y permite exportarla a un archivo PDF. Esta forma es parte de un sistema de gestión de inventario y compras, proporcionando visualizaciones de datos importantes para análisis y reportes.

**Elementos Clave**

1. Control de gráfica para mostrar los datos de compras.
2. Botón para cargar los datos de compras y generar la gráfica.
3. Botón para exportar la gráfica a un archivo PDF.

**Funcionalidades**

1. Cargar datos de compras desde la base de datos y generar una gráfica.
2. Mostrar la gráfica de compras en la interfaz de usuario.
3. Exportar la gráfica a un archivo PDF personalizado.
4. Incluir información del negocio (logotipo, nombre, contacto) en el PDF generado.
5. Permitir al usuario elegir la ubicación de guardado del archivo PDF.
6. Manejar errores y mostrar mensajes de confirmación o error según corresponda.

**Consideraciones de Diseño**

1. Visualización de Datos: Utiliza una gráfica para representar los datos de compras de manera intuitiva y fácil de analizar.
2. Exportación Flexible: Permite exportar la gráfica a PDF, un formato ampliamente compatible y versátil para compartir y archivar.
3. Manejo de Archivos: Implementa la captura de la gráfica como imagen y su posterior inclusión en el PDF.
4. Interacción con el Sistema de Archivos: Utiliza el explorado de archivos para permitir al usuario elegir dónde guardar el archivo PDF.
5. Estructura Organizada: Divide el contenido del PDF en secciones claras (encabezado, título, gráfica) para mejorar la legibilidad.
6. Flexibilidad en la Configuración: Permite ajustar aspectos como tamaños de fuente, colores y alineaciones en el PDF generado.
7. Feedback al Usuario: Muestra mensajes de confirmación o error dependiendo del resultado de las operaciones, mejorando la experiencia del usuario.
8. Estilo Consistente: Aplica un estilo coherente en todo el PDF, incluyendo fuentes y colores, para mantener una apariencia profesional.

#### Diseño de la ventana Graficas Ventas

Gráfico

Descripción generada automáticamente

**Descripción General:** Esta pantalla muestra una gráfica de ventas y permite exportarla a un archivo PDF. Esta forma es parte de un sistema de gestión de inventario y ventas, proporcionando visualizaciones de datos importantes para análisis y reportes.

**Elementos Clave**

1. Control de gráfica para mostrar los datos de ventas.
2. Botón para cargar los datos de ventas y generar la gráfica.
3. Botón para exportar la gráfica a un archivo PDF.

**Funcionalidades**

1. Cargar datos de compras desde la base de datos y generar una gráfica.
2. Mostrar la gráfica de ventas en la interfaz de usuario.
3. Exportar la gráfica a un archivo PDF personalizado.
4. Incluir información del negocio (logotipo, nombre, contacto) en el PDF generado.
5. Permitir al usuario elegir la ubicación de guardado del archivo PDF.
6. Manejar errores y mostrar mensajes de confirmación o error según corresponda.

**Consideraciones de Diseño**

1. Visualización de Datos: Utiliza una gráfica para representar los datos de ventas de manera intuitiva y fácil de analizar.
2. Exportación Flexible: Permite exportar la gráfica a PDF, un formato ampliamente compatible y versátil para compartir y archivar.
3. Manejo de Archivos: Implementa la captura de la gráfica como imagen y su posterior inclusión en el PDF.
4. Interacción con el Sistema de Archivos: Utiliza el explorador de archivos para permitir al usuario elegir dónde guardar el archivo PDF.
5. Estructura Organizada: Divide el contenido del PDF en secciones claras (encabezado, título, gráfica) para mejorar la legibilidad.
6. Flexibilidad en la Configuración: Permite ajustar aspectos como tamaños de fuente, colores y alineaciones en el PDF generado.
7. Feedback al Usuario: Muestra mensajes de confirmación o error dependiendo del resultado de las operaciones, mejorando la experiencia del usuario.
8. Estilo Consistente: Aplica un estilo coherente en todo el PDF, incluyendo fuentes y colores, para mantener una apariencia profesional.

### Pantalla Modales

#### Diseño de la ventana Modal Productos



**Descripción General:** Esta pantalla modal muestra una lista de productos y permite su búsqueda y selección. Esta forma es típicamente utilizada en sistemas de gestión de inventario para ayudar a los usuarios a encontrar y seleccionar productos rápidamente.

**Elementos Clave**

1. Una tabla para mostrar la lista de productos.
2. ComboBox para seleccionar la columna por la cual filtrar los resultados.
3. Campo para ingresar el texto de búsqueda.
4. Botón para iniciar la búsqueda de productos.

**Funcionalidades**

1. Mostrar una lista completa de productos existentes en la base de datos.
2. Permitir la búsqueda de productos por diferentes criterios (código, nombre, categoría, etc.).
3. Filtrar la lista de productos en tiempo real según el criterio de búsqueda seleccionado.
4. Mostrar información detallada de cada producto en la tabla.
5. Permitir la selección de un producto haciendo doble clic en él.
6. Formatear visualmente la columna de estado de los productos (activo/no activo).

**Consideraciones de Diseño**

1. Flexibilidad en la Búsqueda: Permite al usuario elegir por qué columna desea filtrar los resultados, aumentando la utilidad de la búsqueda.
2. Interfaz Intuitiva: Utiliza componentes estándar como DataGridView, ComboBox y TextBox para crear una interfaz familiar para los usuarios.
3. Visualización Mejorada: Formatea la columna de estado usando colores para una rápida identificación visual.
4. Coloreado Condicionado: Aplica colores significativos (verde para activo, rojo para no activo) para una rápida comprensión visual del estado de los productos.
5. Inicialización Automática: Carga la lista de productos y configura el ComboBox al cargar la ventana, asegurando que esté listo para usar inmediatamente.

#### Diseño de la ventana Modal Proveedores

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Descripción General:** Esta pantalla modal muestra una lista de proveedores y permite su búsqueda y selección. Esta forma es típicamente utilizada en sistemas de gestión de inventario para ayudar a los usuarios a encontrar y seleccionar proveedores rápidamente.

**Elementos Clave**

1. Una tabla para mostrar la lista de proveedores.
2. ComboBox para seleccionar la columna por la cual filtrar los resultados.
3. Campo para ingresar el texto de búsqueda.
4. Botón para iniciar la búsqueda de proveedores.

**Funcionalidades**

1. Mostrar una lista completa de proveedores existentes en la base de datos.
2. Permitir la búsqueda de proveedores por diferentes criterios (cédula, nombre, etc.).
3. Filtrar la lista de proveedores en tiempo real según el criterio de búsqueda seleccionado.
4. Mostrar información detallada de cada proveedor en la tabla.
5. Permitir la selección de un proveedor haciendo doble clic en él.
6. Formatear visualmente la columna de estado de los proveedores (activo/no activo).

**Consideraciones de Diseño**

1. Flexibilidad en la Búsqueda: Permite al usuario elegir por qué columna desea filtrar los resultados, aumentando la utilidad de la búsqueda.
2. Interfaz Intuitiva: Utiliza componentes estándar como DataGridView, ComboBox y TextBox para crear una interfaz familiar para los usuarios.
3. Visualización Mejorada: Formatea la columna de estado usando colores para una rápida identificación visual.
4. Coloreado Condicionado: Aplica colores significativos (verde para activo, rojo para no activo) para una rápida comprensión visual del estado de los proveedores.
5. Inicialización Automática: Carga la lista de proveedores y configura el ComboBox al cargar la ventana, asegurando que esté listo para usar inmediatamente.

#### Diseño de la ventana Modal Transportistas

Interfaz de usuario gráfica

Descripción generada automáticamente

**Descripción General:** Esta pantalla modal muestra una lista de transportistas y permite su búsqueda y selección. Esta forma es típicamente utilizada en sistemas de gestión de inventario para ayudar a los usuarios a encontrar y seleccionar transportistas rápidamente.

**Elementos Clave**

1. Una tabla para mostrar la lista de transportistas.
2. ComboBox para seleccionar la columna por la cual filtrar los resultados.
3. Campo para ingresar el texto de búsqueda.
4. Botón para iniciar la búsqueda de transportistas.

**Funcionalidades**

1. Mostrar una lista completa de transportistas existentes en la base de datos.
2. Permitir la búsqueda de transportistas por diferentes criterios (cédula, nombre, etc.).
3. Filtrar la lista de transportistas en tiempo real según el criterio de búsqueda seleccionado.
4. Mostrar información detallada de cada transportista en la tabla.
5. Permitir la selección de un transportista haciendo doble clic en él.
6. Formatear visualmente la columna de estado de los transportistas (activo/no activo).

**Consideraciones de Diseño**

1. Flexibilidad en la Búsqueda: Permite al usuario elegir por qué columna desea filtrar los resultados, aumentando la utilidad de la búsqueda.
2. Interfaz Intuitiva: Utiliza componentes estándar como DataGridView, ComboBox y TextBox para crear una interfaz familiar para los usuarios.
3. Visualización Mejorada: Formatea la columna de estado usando colores para una rápida identificación visual.
4. Coloreado Condicionado: Aplica colores significativos (verde para activo, rojo para no activo) para una rápida comprensión visual del estado de los transportistas.
5. Inicialización Automática: Carga la lista de transportistas y configura el ComboBox al cargar la ventana, asegurando que esté listo para usar inmediatamente.

#### Diseño de la ventana Modal Clientes

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

**Descripción General:** Esta pantalla modal muestra una lista de clientes y permite su búsqueda y selección. Esta forma es típicamente utilizada en sistemas de gestión de inventario para ayudar a los usuarios a encontrar y seleccionar clientes rápidamente.

**Elementos Clave**

1. Una tabla para mostrar la lista de clientes.
2. ComboBox para seleccionar la columna por la cual filtrar los resultados.
3. Campo para ingresar el texto de búsqueda.
4. Botón para iniciar la búsqueda de clientes.

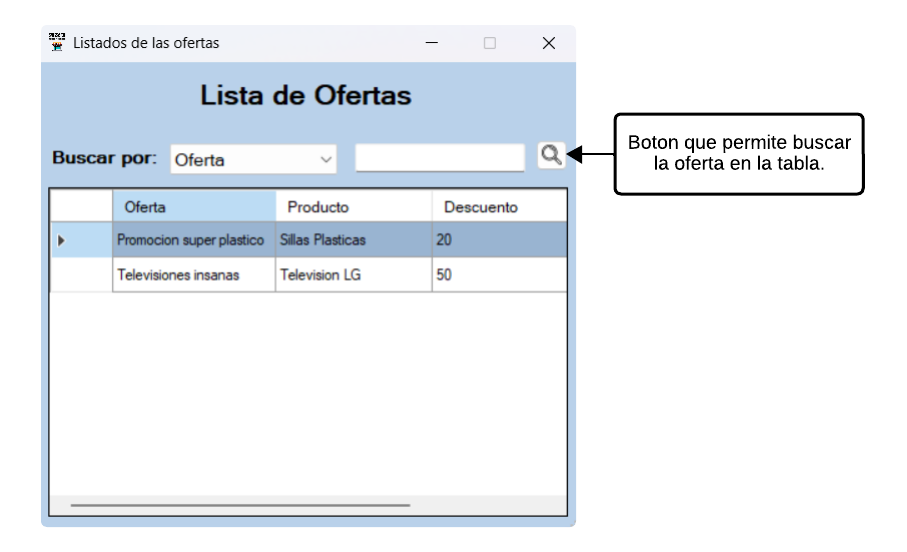
**Funcionalidades**

1. Mostrar una lista completa de clientes existentes en la base de datos.
2. Permitir la búsqueda de clientes por diferentes criterios (cédula, nombre, etc.).
3. Filtrar la lista de clientes en tiempo real según el criterio de búsqueda seleccionado.
4. Mostrar información detallada de cada cliente en la tabla.
5. Permitir la selección de un cliente haciendo doble clic en él.
6. Formatear visualmente la columna de estado de los clientes (activo/no activo).

**Consideraciones de Diseño**

1. Flexibilidad en la Búsqueda: Permite al usuario elegir por qué columna desea filtrar los resultados, aumentando la utilidad de la búsqueda.
2. Interfaz Intuitiva: Utiliza componentes estándar como DataGridView, ComboBox y TextBox para crear una interfaz familiar para los usuarios.
3. Visualización Mejorada: Formatea la columna de estado usando colores para una rápida identificación visual.
4. Coloreado Condicionado: Aplica colores significativos (verde para activo, rojo para no activo) para una rápida comprensión visual del estado de los clientes.
5. Inicialización Automática: Carga la lista de clientes y configura el ComboBox al cargar la ventana, asegurando que esté listo para usar inmediatamente.

#### Diseño de la ventana Modal Ofertas



**Descripción General:** Esta pantalla modal muestra una lista de ofertas y permite su búsqueda y selección. Esta forma es típicamente utilizada en sistemas de gestión de inventario para ayudar a los usuarios a encontrar y seleccionar ofertas rápidamente.

**Elementos Clave**

1. Una tabla para mostrar la lista de ofertas.
2. ComboBox para seleccionar la columna por la cual filtrar los resultados.
3. Campo para ingresar el texto de búsqueda.
4. Botón para iniciar la búsqueda de clientes.

**Funcionalidades**

1. Mostrar una lista completa de ofertas existentes en la base de datos.
2. Permitir la búsqueda de ofertas por diferentes criterios (nombre, producto, descuento, etc.).
3. Filtrar la lista de ofertas en tiempo real según el criterio de búsqueda seleccionado.
4. Mostrar información detallada de cada oferta en la tabla.
5. Permitir la selección de una oferta haciendo doble clic en él.
6. Formatear visualmente la columna de estado de las ofertas (activo/no activo).

**Consideraciones de Diseño**

1. Flexibilidad en la Búsqueda: Permite al usuario elegir por qué columna desea filtrar los resultados, aumentando la utilidad de la búsqueda.
2. Interfaz Intuitiva: Utiliza componentes estándar como DataGridView, ComboBox y TextBox para crear una interfaz familiar para los usuarios.
3. Visualización Mejorada: Formatea la columna de estado usando colores para una rápida identificación visual.
4. Coloreado Condicionado: Aplica colores significativos (verde para activo, rojo para no activo) para una rápida comprensión visual del estado de las ofertas.
5. Inicialización Automática: Carga la lista de ofertas y configura el ComboBox al cargar la ventana, asegurando que esté listo para usar inmediatamente.