

Título Principal: MANUAL TÉCNICO PROGRAMA SAIA.



Manual Técnico para Uso del Programa de Administración de Agro Servicios SAIA

Autores: Diego Alejandro Vasquez Valle 20220668

Josué Carlos Ayala Reyes 20240001

Christopher Alexander Morales Quijano 20240207

Felipe Sebastián Pocasangre Clímaco 20240046

Fecha de entrega: 23/09/2024

Instituto Técnico Ricaldone

- **Índice de Contenidos.**

PORTADA.....	1
ÍNDICE.....	2
INTRODUCCIÓN.....	3
DIAGRAMA DE ACTIVIDADES	4
DIAGRAMA DE CASOS DE USO.....	6
DIAGRAMA DE CLASES.....	7
DIAGRAMA ENTIDAD RELACIÓN.....	10
DICCIONARIO DE DATOS.....	11
CODIFICACIÓN DE ERRORES.....	16
ANEXOS.....	28

- **Introducción:**

El presente manual busca proporcionar todos los recursos necesarios para manejar y controlar el programa de administración de Agro Servicios SAIA, en él se encuentra información como tablas de administración de productos, ventas y usuarios, procesamiento de información dentro del sistema, manejo de usuarios empleados y administradores, manejo de tablas dentro de la base de datos, el como navegar por la interfaz del programa eficientemente, diagramas ilustrativos explicados en texto y las posibilidades y limitaciones en el uso del programa administrativo.

- **Diagramas.**

1. Diagrama de Actividades.

¿Cómo Entrar al Programa?:

En caso de *Primer Uso* ingrese un *Nombre de Usuario* y una *Contraseña* para continuar. Para ingresar al programa necesita un *Nombre de Usuario* y *Contraseña* válidos, en caso de no recordar su *Nombre de Usuario* o *Contraseña* contamos con varios métodos de recuperación de ambos, por ejemplo: mediante *Correo Electrónico* e *Intervención Directa del Administrador*. En caso de la *Recuperación por medio de Correo Electrónico* necesita ingresar su *Correo* o *Nombre de Usuario* con Correo enlazado, se le mandará un mensaje al Correo Respectivo con su nueva *Contraseña* y *Nombre de Usuario*. En cambio, si elije la *Recuperación de Contraseña o Usuario por medio de Intervención del Administrador* deberá escribir su problema al correo del administrador y este restaurará su *Usuario* o *Contraseña* si lo encuentra adecuado, una vez haya ingresado sus datos exitosamente se le presentará la *Configuración Guardada del Usuario* y habrá entrado al programa.

¿Cómo manejar la Tabla de Productos?

Para ingresar a la tala de Productos el Usuario debe clickear el botón “Productos” en el menú principal. Al ingresar a la *Al Tabla de Productos* el *Usuario* puede realizar tres acciones: *Salir de la Tabla de Productos*, *Ver las Marcas* y *Ver los Productos*. Para salir de la *Tabla de Productos* solo debe cerrar la pestaña de *Tabla de Productos*, esto se hace presionando la X en la esquina superior derecha. Para ver los productos el *Usuario* puede verlos en la cuadrícula de abajo donde se muestran el ID del Producto, el Nombre, el ID de Marca, cantidad en Stock, breve descripción del

Producto y el Código. Para agregar un Producto se tiene que clickear en el botón que dice “Agregar Producto”, una vez presionado se abrirá otro formulario pidiendo los detalles del producto que se desea agregar, tales como: Nombre del Producto, Código, Cantidad, Costo, Marca, una Descripción y una Imagen, luego solo se presiona el botón “Agregar” y si los datos son válidos se ingresa en la cuadrícula de la *Tabla de Productos*. Para ver las Marcas disponibles solo se debe presionar el botón con el texto “Agregar Marca”, luego de presionado se abrirá un formulario con una nueva vista de cuadrícula con el Id de la Marcas, y los nombres de las Marcas ya ingresadas, para ingresar una nueva Marca solo se necesita ingresar su nombre y aparecerá en la cuadrícula de la Tabla de Marcas y será elegible para asignarla a un nuevo producto. Para eliminar o modificar una Marca se debe de entrar a la Tabla de Marcas presionando el botón “Agregar Marca” y seleccionando clic derecho a la tabla que se desea eliminar o actualizar, una vez hecho esto solo se selecciona la opción que se desea hacer y se confirma dando “si” en la ventana emergente.

¿Cómo Manejar la Tabla de Proveedores?:

Para ingresar a la tabla de *Proveedores* debes clickear el botón llamado “*Proveedores*” en el menú principal. Una vez adentro se mostrará la tabla de *Proveedores* con datos tales como: Id del Proveedor, Nombre, DUI, Teléfono, Correo e Id de la Marca. Arriba de la tabla de *Proveedores* se encuentra un botón titulado “Añadir Proveedores” presionando dicho botón se abrirá una ventana emergente en la que se pueden ingresar los datos de nuevos proveedores, estos datos son: Nombre completo del nuevo proveedor, DUI, número celular, correo electrónico y empresa, para añadir el proveedor ingresado solo se debe presionar el botón “Agregar Proveedor” en caso que todos los datos sean válidos y no interfieran con ningún otro proveedor ya existente se mostrará el nuevo proveedor en la tabla de *Proveedores* después de confirmar su selección.

2. Diagrama de Caso de Uso.

¿Cómo usar el programa según distintos niveles de acceso?:

En el caso del Administrador se pueden realizar todas las acciones sin límite alguno, únicamente cohibido por la correcta administración de un negocio de Agro Servicios, pueden contratar nuevos empleados y recuperar información para otros niveles de usuario. El empleado tiene menores libertades que el Administrador, no puede despedir otros empleados, ni observar los registros de contraseñas y usuarios, pueden cancelar compras e inspeccionar los proveedores, marcas, productos y registros de ventas con información restringida. El usuario únicamente puede crear una cuenta de usuario y observar el inventario para encargar compras que deberán ser confirmadas por un empleado, también puede utilizar el motor de búsqueda incluido en el sistema de inventarios.

3. Diagrama de Clases.

¿Cómo funciona el programa?

En la tabla de *Productos* se ubican los Getter y Setters de Id de productos, Nombre, id de Marca, precio, stock (o cantidad), imagen y descripción, esta información se puede utilizar para identificar en el buscador mediante específicos en el *Data Grid View* o consultando directamente la base de datos, este cuenta con las clases “Nuevo Producto” que inicia la secuencia para agregar un nuevo producto, “Eliminar Producto” que comienza el proceso para eliminar un producto, “Editar Producto” que empieza el procedimiento para actualizar un producto, “Información Producto” que muestra los productos en el *Data Grid View*, “Agregar Imagen” que enseña la imagen asignada de cada producto en el Data Grid View y “Buscar Producto” que compara Id y otros datos con el inventario para encontrar el producto que se busca. La tabla de *Proveedores* cuenta igualmente con sus respectivos Getters y Setters como: Nombre del Proveedor, DUI, id de Proveedor (este se conecta con la tabla productos), teléfono, correo y el id de Marca, cuenta con las clases “Agregar Proveedor” que empieza el proceso para añadir un nuevo proveedor, “Actualizar Registro” que abre una pestaña para actualizar un proveedor ya existente y “Eliminar Proveedor” que elimina de la base de datos el proveedor seleccionado. La tabla de Marcas cuenta con dos Getters y Setters: Id de Marca y Nombre de Marca estos se relacionan directamente con los productos y aparecen en el *Data Grid View* en la Vista, este cuenta con las clases “Añadir Marca” que empieza el curso de agregar una marca, “Eliminar Marca” elimina una marca específica y “Actualizar Marca” que comienza la ventana para modificar una marca. La tabla de Clientes cuenta con cinco Getters y Setters estos son: Id de Cliente (este se conecta directamente con la tabla de productos), Nombre de Cliente, teléfono, correo y dirección, sus clases son “Registrar Cliente” que ingresa un nuevo cliente a la base de datos e “Información Cliente” que muestra y actualiza la vista de la tabla de Clientes. La tabla de empleados con siete Getters y Setters como: id de Empleado, Nombre del Empleado, Fecha de

nacimiento, teléfono, correo, DUI y dirección del empleado, sus clases son: “Nuevo Registro” que registra un nuevo empleado, “Eliminar Empleado” que elimina el empleado deseado, “Actualizar Empleado” que permite actualizar los datos de un empleado ya existente, “Rest Empleado” que desactiva por un tiempo indeterminado a un empleado (solo para administradores), “información Empleado” muestra los datos relevantes de todos los empleados o de uno en específico, “Agregar Imagen” que permite añadir una imagen a la vista del empleado, aparte el empleado puede crear Usuarios. La tabla de Usuarios tiene los Getters y Setters Usuario, Contraseña, Intentos de Login, id categoría (que obtiene de la tabla de categorías) y Foto de perfil, cuenta con dos clases “Crear Usuario” que crea un usuario nuevo y “Asignar Usuario” otorga su categoría como: el id Categoría y el Nombre de Categoría; ambos solo pueden ejecutarse por medio de un empleado. El Cliente selecciona lo que quiere comprar y el empleado se encarga de cancelar su compra, la tabla de Ventas requiere los Getters y Setters: id de Venta, fecha de venta, cantidad de producto y total a pagar, cuenta con dos clases “Nueva Venta” que registra la venta y la envía a la tabla de detalle Venta y “Vuelto” que regresa el excedente del precio total de los productos adquiridos. La tabla de Detalle de Venta cuenta con los Getters y Setters respectivos de: id de Historial, fecha de venta, monto y total a pagar, sus clases son: “Guardar Venta” que actualiza la base de datos con las informaciones necesarias a la tabla de ventas semanales, “Eliminar Detalle de Venta” que elimina el registro de una compra realizada anteriormente e “información Detalle de Venta” que refresca la información en el registro de la tabla de Detalle de Venta, “Detalle de Venta Devolución” que manda la información a la tabla de Devoluciones. La tabla de Ventas Semanales es un registro con Getters y Setters para: id Venta Semanal, fecha, monto y total, con las clases “Calcular Venta” que registra la información de la venta obtenida de la tabla de Detalle de Venta, “Calcular Cantidad” que muestra la cantidad de dinero obtenido en la semana e “Información de Ventas Semanales” que muestra los registro mediante un Data Grid View en la tabla de Ventas Semanales. La tabla de Devoluciones cuenta con los Getters y Setters de: id de Devolución, fecha de Devolución, monto de Devolución, cantidad de Devolución, id del

Cliente, id del Empleado e id del Detalle de Venta, tiene dos clases: “Devolución” que confirma y realiza la devolución del producto seleccionado e “información de Devolución” que muestra mediante un Data Grid View las devoluciones realizadas o pendientes por realizar.

4. Diagrama de Entidad Relación.

Las categorías cuentan con un nombre y un id de categoría estas se heredan al usuario que cuenta con intentos limitados de acceso al programa (3), una foto asociada a su perfil, una id de categoría anteriormente mencionada, una fecha de nacimiento, un usuario, un nombre y una contraseña, el empleado es también un usuario así que cuenta con un usuario, una fecha de nacimiento, un nombre todos heredados de la tabla anterior, una id de empleado, un número de teléfono asignado, un correo electrónico mail, una dirección requerida y un DUI. Las marcas cuentan con un nombre de marca y una id de marca que se heredan al proveedor. El proveedor cuenta con un DUI, un número de teléfono, un id de proveedor, un nombre de empresa, un correo electrónico mail, una marca asignada y un nombre de representante, los productos heredan el id de una marca y un id de proveedor, cuentan con un id de producto único, una cantidad ajustable de stock o inventario, un precio igualmente ajustable, un nombre común, una descripción breve informativa, un código asociado a su potencial venta y una imagen del producto. El detalle de venta hereda la información del producto, este cuenta con una cantidad de productos vendidos, un id de venta, un id de detalle de venta, una fecha de adquisición de producto, id del producto, id de la marca, id del proveedor, cantidad restante de producto específico en inventario o stock y precio necesario para cancelar la compra, aparte de una opción de devolución con almacenando la fecha de devolución, cantidad de productos devueltos, id del cliente que realizó la devolución, id del empleado que facilitó la devolución, monto descontado por devolución y el todos los detalles de la venta antes del producto devuelto .El producto se transfiere al detalle de venta, el empleado provee la aprobación de venta para un cliente que realiza la compra mediante un detalle de venta específico de su elección con el cual puede devolver el producto antes de alcanzar la fecha estipulada para la validez de devolución o conservarlo.

- **Diccionario de Datos.**

- ***Data_de_datos_Agro Tablas:***

- **Tabla Categorías:**

idCategoria *INT* *identity(1,1)* *PRIMARY KEY,* *--Código para identificar rol*
Nombre *varchar(15)* *Not null* *--El nombre del rol*

- **Tabla Usuarios:**

Usuario *varchar(50)* *PRIMARY KEY,* *--Nombre del usuario*
Contraseña *varchar(100)* *not null,* *--Contraseña del usuario*
IntentosUsuario *INT,* *--Intentos que tiene para ingresar el usuario y contraseña*
idCategoria *INT* *Foreign key References Categorías(idCategoria)* *Not null,*
 --Llave foránea de la tabla de Categorías, es el nombre del rol; es obligatorio
picprofile *VARBINARY(MAX)* *--Almacena una imagen*

- **Tabla Empleados:**

idEmpleado *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar el empleado*
Nombre *varchar (65)* *Not null,* *--Nombre del Empleado*
FechaDeNacimiento *DateTime CHECK(FechaDeNacimiento >= 18)* *not null,* *--Fecha de nacimiento del Empleado*
Telefono *varchar(15)* *Not null,* *--Número de Teléfono del Empleado*
Correo *varchar(75)* *Not null,* *--Correo Electrónico del Empleado*
DUI *varchar(15)* *UNIQUE* *Not null,* *--DUI del Empleado, es obligatorio*
Direccion *varchar(150)* *Not null,* *--Dirección del Empleado, es obligatoria*
Usuario *varchar(50)* *Foreign key References Usuarios(Usuario)* *--Llave foránea de la tabla de Usuarios, es el nombre de usuario*

- **Tabla Marcas:**

idMarca *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar la Marca*
NombreMarca *varchar (15)* *UNIQUE* *--Nombre de la Marca*

- **Tabla Proveedores:**

idProveedor INT IDENTITY(1,1) PRIMARY KEY, --Código para identificar al Proveedor
Nombre varchar(65) UNIQUE Not null, --Nombre del Proveedor, es obligatorio
DUI varchar(15) UNIQUE Not null, --DUI del Proveedor, es obligatorio
Teléfono varchar(15) Not null, --Número de Teléfono del Proveedor, es obligatorio
Correo varchar(75) Not null, --Correo Electrónico del Proveedor, es obligatorio
idMarca INT Foreign key (idMarca) References Marcas(idMarca) Not null
 --Llave foránea de la tabla de Marca, es el nombre de la Marca; es obligatoria

• Tabla Productos:

idProducto INT IDENTITY(1,1) PRIMARY KEY, --Código para identificar el Producto
Nombre varchar(80) UNIQUE Not null, --Nombre del Producto
idMarca INT Foreign key (idMarca) References Marcas(idMarca) Not null, --Llave foránea de la tabla de Marca, es el nombre de la Marca; es obligatorio
Precio DECIMAL(10,2) Not null, --Precio del Producto, es obligatorio
Stock INT Not null, --Cantidad en inventario, es obligatorio
Descripcion varchar (150) Not null, --Breve descripción del Producto, es obligatoria
Codigo INT UNIQUE Not null, --Código único de cada producto, es obligatorio
imgNombre VARBINARY(MAX), --Imagen ilustrativa del Producto

• Tabla Clientes:

idCliente INT IDENTITY(1,1) PRIMARY KEY, --Código para identificar al Cliente
Nombre varchar(65) Not null, --Nombre del Cliente, es obligatorio
Telefono varchar(15) Not null, --Número de Teléfono del Cliente, es obligatorio
Correo varchar(75) Not null, --Correo Electrónico del Cliente, es obligatorio
Direccion varchar(150) Not null, --Dirección del Cliente, es obligatorio
DUI varchar(15) UNIQUE Not null, --DUI del Cliente, es obligatorio

- **Tabla Ventas:**

idVenta *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar la venta*

idCliente *INT* *Foreign key (idCliente) References Clientes(idCliente)* *Not null,*
--Llave foránea de la tabla de Clientes, es el cliente que realiza la venta; es obligatorio

idEmpleado *INT* *Foreign key (idEmpleado) References Empleados(idEmpleado)*
Not null, *--Llave foránea de la tabla de Empleados, es el empleado que facilita la venta; es obligatorio*

FechaVenta *DATETIME* *Not null,* *--Fecha de la venta, es obligatoria*

MontoTotal *DECIMAL(10,2)* *Not null* *--Costo de la compra, es obligatorio*

- **Tabla Detalle de Venta:**

idDetalleVenta *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar el detalle de venta*

idEmpleado *INT* *Foreign key (idEmpleado) References Empleados(idEmpleado)* *Not null,*
--Llave foránea de la tabla de Empleados, es el empleado que atendió la venta; es obligatorio

idCliente *INT* *Foreign key (idCliente) References Clientes(idCliente)* *Not null,*
--Llave foránea de la tabla de Clientes, es el cliente que realizó la compra; es obligatorio

idProducto *INT* *Foreign key (idProducto) References Productos(idProducto)* *Not null,*
--Llave foránea de la tabla de Productos, es el producto que se vendió; es obligatorio

Fecha *DATE* *Not null,* *--Fecha de la venta, es obligatorio*

Cantidad *INT* *Not null,* *--Cantidad de Productos*

Monto_Total *DECIMAL* *(10,2* *Not null* *--Costo de la compra, es obligatorio*

- **Tabla Cierre de Ventas:**

idCierreVenta *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar la venta cerrada*

idDetalleVenta *INT* *Foreign key (idDetalleventa) References DetalleVenta(idDetalleVenta),* *--Llave foránea de la tabla Detalle de Venta, es la información de la venta; es obligatoria*

Fecha *DATE* *Not null,* *--Fecha del cierre de la venta, es obligatorio*

Cantidad *INT* *Not null,* *--Cantidad de cierre de ventas, es obligatorio*

MontoTotal *DECIMAL(10,2)* *Not null* *--Ganancias del cierre de la venta*

• **Tabla Cierre Semanal:**

idCierreSemanal *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar el cierre semanal*

idCierreVenta *INT* *Foreign key (idCierreVenta) References CierreVentas(idCierreVenta),* *--Llave foránea de la tabla de Cierre de Venta, son las ventas finalizadas; es obligatorio*

Fecha *DATE* *Not null,* *--Fecha de la semana*

CantidadSemanal *INT* *Not null,* *--Total de ventas semanales*

MontoSemanal *DECIMAL(10,2)* *Not null* *--Cantidad de dinero ganado en la semana*

• **Tabla Devoluciones:**

idDevolucion *INT* *IDENTITY(1,1)* *PRIMARY KEY,* *--Código para identificar la devolución*

idVenta *INT,* *--Código para identificar la Venta*

idProducto *INT,* *--Código para identificar el Producto*

idCliente *INT,* *--Código para identificar el Cliente*

FechaDevolucion *DATETIME,* *--Fecha de la devolución*

Cantidad *INT,* *--Cantidad de productos que se devuelven*

MontoDevolucion *DECIMAL(10, 2),* *--Reembolso requerido*

Motivo *varchar(255),* *--Justificación de retorno del producto*

CONSTRAINT *FK_Venta* *Foreign key (idVenta) References Ventas(idVenta),* *--Constraint-Llave foránea de la tabla de Ventas, es la venta que se cancela; es obligatoria*

CONSTRAINT *FK_Producto* *Foreign key (idProducto) References Productos(idProducto),* *--Constraint-Llave foránea de la tabla de Productos, es el producto que se devuelve; es obligatorio*

CONSTRAINT *FK_Cliente* *Foreign key (idCliente) References Clientes(idCliente)*
--Constraint-Llave foránea de la tabla de Clientes, es el cliente que hace la devolución; es obligatorio

• **Tabla Preguntas de Seguridad:**

PreguntaID *INT* *PRIMARY KEY* *IDENTITY(1,1),* *--Código para identificar la Pregunta de Seguridad*

Pregunta *nvarchar(255) Not null* *--La pregunta de seguridad en texto*

• **Tabla Respuestas de Seguridad:**

RespuestaID *INT* *PRIMARY KEY* *IDENTITY(1,1),* *--Código para identificar la Respuesta a una Pregunta de Seguridad*

Usuario *varchar (50)* *Foreign key (Usuario) References Usuarios(Usuario),* *--Llave foránea de la tabla de Usuarios, es a que usuario le pertenece una respuesta de seguridad específica, es obligatorio*

PreguntaID *INT* *Foreign key (PreguntaID) References PreguntasSeguridad(PreguntaID),* *--Llave foránea de la tabla de Preguntas de Seguridad, es la pregunta a la que corresponde la respuesta, es obligatoria*

RespuestaCifrada *nvarchar(255) Not Null,* *--La respuesta a una Pregunta de Seguridad específica*

• Codificación de Errores.

• Errores posibles al Crear Productos:

- Error de llenado de campos.

```
MessageBoxP(Color.Yellow, Color.Orange, "Error", "Todos los campos son obligatorios",
Properties.Resources.MensajeWarning);

return;
```

- Final del Error.

No se llenaron todos los campos.

Solución: Compruebe que llenó correctamente todos los campos necesarios e intente de nuevo.

- Más de 80 Caracteres en nombre.

```
if (!ValidarNombre(ObjCreateProducto1.txtNombreProducto.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "El nombre del producto tiene más de 80 caracteres",
Properties.Resources.MensajeWarning);

    return;
}
```

- Final del Error.

El nombre del producto tenía más de 80 caracteres.

Solución: Revise que el nombre del producto no exceda el límite de caracteres e intente nuevamente.

- Se ingresó un valor incorrecto en código.

```
if (!ValidarNumero(ObjCreateProducto1.txtCodigo.Text) ||
    !ValidarNumero(ObjCreateProducto1.txtCantidad.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "La cantidad y el código del producto solo deben de tener
números enteros", Properties.Resources.MensajeWarning);

    return;
}
```

- Final del Error.

No se ingresó el valor adecuado.

Solución: Valide que la cantidad del producto contenga solo números enteros e intente de nuevo.

- Más de 14 números en el código.

```
if (!CantidadCodigo(ObjCreateProducto1.txtCodigo.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "El código del producto tiene más de 14 números",
        Properties.Resources.MensajeWarning);

    return;
}
```

Fin del Error.

Se ingresaron más de 14 números en el código de producto.

Solución: Verifique que el código del producto no exceda los 14 dígitos e intente de nuevo.

- Más de 500 productos.

```
if (!CantidadStock(ObjCreateProducto1.txtCantidad.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "La cantidad del producto no puede exceder los 500",
        Properties.Resources.MensajeWarning);

    return;
}
```

▪ *Fin del Error.*

Se ingresaron más de quinientos productos.

Solución: Verifique que no se haya ingresado una cantidad de productos mayor a las 500 unidades e intente de nuevo.

- Error al poner precios.

```
if (!ValidarDecimales(ObjCreateProducto1.txtPrecio.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "El precio del producto solo debe de tener numeros decimales
    y no exeder los 1000.00", Properties.Resources.MensajeWarning);

    return;
}
```

▪ *Fin del Error.*

Solo pueden ingresarse números decimales.

Solución: Valide que el precio del producto solo contenga decimales e intente de nuevo.

- Más de 100 caracteres en la descripción.

```
if (!ValidarDescripcion(ObjCreateProducto1.txtDescripcion.Text))
```

```
{
```

```
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "La descripción del producto tiene más de 100 caracteres",
    Properties.Resources.MensajeWarning);
```

```
    return;
```

```
}
```

- Fin del Error:

Se ingresaron más de 100 caracteres en la descripción de un producto e intente de nuevo.

Solución: Verifique que la descripción de su producto no exceda los 100 caracteres.

- Producto no fue registrado.

```
{
```

```
    MandarValoresAlerta(Color.Red, Color.DarkRed, "Error", "Verifique que el producto no se esté duplicando o
    asociando a otro registro", Properties.Resources.ErrorIcono);
```

```
    VistaLogin backForm = new VistaLogin();
```

```
}
```

- Fin del Error:

El producto ingresado no fue registrado.

Solución: Valide que el producto no esté duplicado o con valores incorrectos e intente de nuevo.

• Errores posibles al crear una Marca:

- Error de llenado de campos.

```
if (ObjCreateMarca.txtNombreMarca.Text == null)
```

```
{
```

```
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Todos los campos son obligatorios",
    Properties.Resources.MensajeWarning);
```

```
    return;
```

```
}
```

- Fin del Error:

No se llenaron todos los campos.

Solución: Compruebe que llenó correctamente todos los campos necesarios e intente de nuevo.

- Más de 15 Caracteres en nombre.

```

if (!ValidarNombre(ObjCreateMarca.txtNombreMarca.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Hay más de 15 caracteres en el nombre de la marca",
        Properties.Resources.MensajeWarning);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 15 caracteres en el nombre.

Solución: Valide que el nombre del producto no exceda 15 caracteres e intente de nuevo.

- **Marca no fue registrada.**

```

{
    MessageBoxP(Color.Red, Color.DarkRed, "Error", "Verifique que los datos no estén duplicados",
        Properties.Resources.ErrorIcono);
}

```

▪ *Fin del Error.*

La marca ingresada no fue registrada.

Solución: Valide que la marca no esté duplicada o con valores incorrectos e intente de nuevo.

- **Error al eliminar una Marca.**

```

catch (Exception ex)
{
    MessageBox.Show("No se pudo eliminar la marca. \n\n" +
        "Es posible que esté asociada a otros registros, como productos. Verifique esto antes de intentarlo de nuevo.\n" +
        "Error: " + ex.Message,
        "Error al Eliminar", MessageBoxButtons.OK, MessageBoxIcon.Warning);
}

```

▪ *Final del Error.*

No se pudo eliminar la **Marca** deseada.

Solución: Verifique que haya ingresado un valor valido o vuelva a intentarlo / si el problema persiste elimine el registro y vuelva a ingresar uno nuevo.

• **Errores posibles al crear Cuentas:**

- Error de llenado de campos

```

{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Faltan campos que llenar",
Properties.Resources.MensajeWarning);

    return;
}

```

▪ *Fin del Error.*

No se llenaron todos los campos.

Solución: Compruebe que llenó correctamente todos los campos necesarios e intente de nuevo.

- Más de 50 Caracteres en el Nombre de Usuario.

```

if (!ValidarUsuario(ObjUsers.txtNewUser.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Hay más de 50 caracteres en el usuario",
Properties.Resources.Report2);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 50 caracteres para el nombre de usuario.

Solución: Verifique que no se hayan ingresado más de 50 caracteres para la sección de usuario e intente de nuevo.

- Números en el Nombre.

```

if (!ValidarLetra(ObjUsers.txtNewFirstName.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Números en el nombre del empleado",
Properties.Resources.Report2);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron números en el nombre del empleado.

Solución: Valide que en el nombre del empleado solo se hayan ingresado letras, nada de números e intente de nuevo.

- Más de 65 Caracteres en Nombre.

```

if (!ValidarNombre(ObjUsers.txtNewFirstName.Text))

```

```

{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Hay más de 65 caracteres en el nombre del empleado",
    Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 65 caracteres en el nombre del empleado.

Solución: Valide que el nombre del empleado no exceda 65 caracteres e intente de nuevo.

- **Más de 15 Caracteres en la contraseña.**

```

if (!ValidarContraseña(ObjUsers.txtNewPassword.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Hay más de 15 caracteres en la contraseña",
    Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 15 caracteres en la contraseña del empleado.

Solución: Valide que la contraseña del empleado no exceda 15 caracteres e intente de nuevo.

- **Teléfono Salvadoreño.**

```

if (!ValidarTelefono(ObjUsers.txtNewPhone.Text))
{
    MessageBoxP(Color.Yellow, Color.DarkRed, "Error", "El teléfono debe de ser de El Salvador",
    Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Se ingreso un número de teléfono invalido.

Solución: Valide el formato del número de teléfono e intente de nuevo.

- **Falta de @ en Correo.**

```

if (!ValidarCorreo(ObjUsers.txtNewCorreo.Text))
{

```

```

    MessageBoxP(Color.Yellow, Color.DarkRed, "Error", "Falta el @ o el dominio del correo",
Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Faltó ingresar @ en la dirección del correo.

Solución: Valide el formato del correo e intente de nuevo.

▪ **Más de 75 Caracteres en el Correo.**

```

if (!ValidarCorreoCantidad(ObjUsers.txtNewCorreo.Text))
{
    MessageBoxP(Color.Yellow, Color.DarkRed, "Error", "Hay más de 75 caracteres en el correo",
Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 75 caracteres en el correo.

Solución: Verifique que no se hayan ingresado más de 75 caracteres en el apartado del correo e intente de nuevo.

▪ **Más de 100 Caracteres en Dirección.**

```

if (!ValidarDireccion(ObjUsers.txtNewDireccion.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "La dirección no debe exceder los 100 caracteres",
Properties.Resources.Report2);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 100 caracteres en el apartado de dirección.

Solución: Valide que la dirección del empleado no exceda los 100 caracteres e intente de nuevo.

▪ **Más de 8 Caracteres en DUI.**

```

if (!ValidarDUI(ObjUsers.maskDui.Text))
{

```

```

    MessageBoxP(Color.Yellow, Color.DarkRed, "Error", "El DUI debe de tener 8 caracteres",
Properties.Resources.Delete);

    return;
}

```

▪ *Fin del Error.*

Se ingresaron más de 8 caracteres en el apartado de DUI.

Solución: Valide el formato del DUI e intente de nuevo.

- Eliminar un Usuario.

```

{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "No se pudo eliminar al usuario",
Properties.Resources.MensajeWarning);
}

```

▪ *Fin del Error.*

No se eliminó el usuario.

Solución: Verifique que haya ingresado un valor valido o vuelva a intentarlo / si el problema persiste elimine el registro y vuelva a ingresar uno nuevo.

• Errores posibles al crear Clientes:

- Error de Llenado de Campos.

```

if (string.IsNullOrEmpty(ObjCreateCliente.txtNombreCliente.Text) ||
    string.IsNullOrEmpty(ObjCreateCliente.maskDui.Text))
{
    MessageBoxP(Color.Yellow, Color.Orange, "Error", "Todos los campos son obligatorios",
Properties.Resources.MensajeWarning);

    return;
}

```

▪ *Fin del Error.*

No se llenaron todos los campos.

Solución: Valide que los campos no estén vacíos e intente de nuevo.

- Números en el Nombre del Cliente.

```

if (!ValidarLetra(ObjCreateCliente.txtNombreClienteg.Text))
{

```

```

    MessageBoxP(Color.Yellow, Color.Orange, "Error", "No se pueden ingresar números en el nombre",
Properties.Resources.MensajeWarning);

```

```

    return;

```

```

}

```

▪ Fin del Error.

Se ingresaron números en el nombre del cliente.

Solución: Valide el formato del nombre solo si se ingresó uno e intente de nuevo.

- Más de 65 Caracteres en Nombre.

```

if (!ValidarNombre(ObjCreateCliente.txtNombreCliente.Text))

```

```

{

```

```

    // ValidarDUI ya muestra un mensaje de error si el formato es incorrecto.

```

```

    MessageBox.Show("El nombre del cliente no debe exceder los 65 caracteres.",

```

```

        "Error de validación",

```

```

        MessageBoxButtons.OK,

```

```

        MessageBoxIcon.Error);

```

```

    return;

```

```

}

```

▪ Fin del Error.

Se ingresaron más de 65 caracteres en el nombre del cliente.

Solución: Valide que el nombre del cliente no exceda los 65 caracteres e intente de nuevo.

- Más de 9 dígitos en DUI.

```

if (!ValidarDUI(ObjCreateCliente.maskDui.Text))

```

```

{

```

```

    MessageBox.Show("El DUI del cliente debe contener exactamente 9 dígitos",

```

```

        "Error de validación",

```

```

        MessageBoxButtons.OK,

```

```

        MessageBoxIcon.Error);

```

```

    return;

```

```

}

```

▪ Fin del Error.

Se ingresaron más de 9 dígitos en el DUI.

Solución: Valide el formato del DUI e intente de nuevo.

- Más de 8 dígitos.

```
if (!string.IsNullOrEmpty(ObjCreateCliente.txtTelefonoCliente.Text) && !ValidarTelefono())
{
    MessageBox.Show("El número de teléfono debe tener 8 dígitos y comenzar con 2, 6, o 7 (válido para El Salvador).",
        "Error de validación",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

    return;
}
```

- Fin del Error.

No se ingresaron los dígitos adecuados.

Solución: Valide el formato del teléfono solo si se ingresó uno e intente de nuevo.

- Falta @ en Correo.

```
if (!string.IsNullOrEmpty(ObjCreateCliente.txtCorreoCliente.Text))
{
    if (!ValidarCorreo(ObjCreateCliente.txtCorreoCliente.Text.Trim()))
    {
        MessageBox.Show("Formato de correo inválido. Verifique que contiene '@' y un dominio válido.",
            "Error de validación",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);

        return;
    }
}
```

- Fin del Error.

No se ingresó el @ en el apartado del correo.

Solución: Valide el formato del correo solo si se ingresó uno e intente de nuevo.

- Más de 75 Caracteres en el Correo

```
if (!ValidarCorreoCantidad(ObjCreateCliente.txtCorreoCliente.Text.Trim()))
```

```

{
    MessageBox.Show("El correo del proveedor no debe tener más de 75 caracteres",
        "Error de validación",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

    return;
}

```

▪ Fin del Error.

Se ingresaron más de 75 caracteres en el apartado del correo.

Solución: Verifique que el correo no exceda los 75 caracteres e intente de nuevo.

▪ Más de 150 Caracteres en Dirección.

```

if (!string.IsNullOrEmpty(ObjCreateCliente.txtDireccionCliente.Text) && !ValidarDireccion())
{
    // ValidarCantidadTelefono ya muestra un mensaje de error si el formato es incorrecto.

    MessageBox.Show("En la dirección del cliente no se pueden ingresar más de 150 caracteres", "Error de
validación", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return;
}

```

▪ Fin del Error.

Se ingresaron más de 150 caracteres en el apartado de dirección del cliente.

Solución: Valide la dirección del cliente solo si se ingresó uno e intente de nuevo.

▪ Error de Registro.

```

{
    MessageBox.Show("Los datos no pudieron ser registrados, verifique que los datos no estén duplicados.",
        "Proceso interrumpido",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}

```

▪ Fin del Error.

No se pudo ingresar los datos.

Solución: Verifique que los datos no estén duplicados e intente de nuevo.

- **Errores posibles en Preguntas de Seguridad:**

- **Contraseña Invalida.**

```
if (string.IsNullOrEmpty(objrest.txtRestPass.Text))
{
    MessageBox.Show("Introduzca una contraseña valida.",
        "Error de validación",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

    return;
}
```

- *Fin del Error.*

Se ingresó una contraseña incorrecta.

Solución: Compruebe que la contraseña ingresada sea la correcta e intente de nuevo.

- **Error con la Nueva Contraseña.**

```
{
    MessageBox.Show("La contraseña no se ha podido actualizar.",
        "Proceso interrumpido",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
```

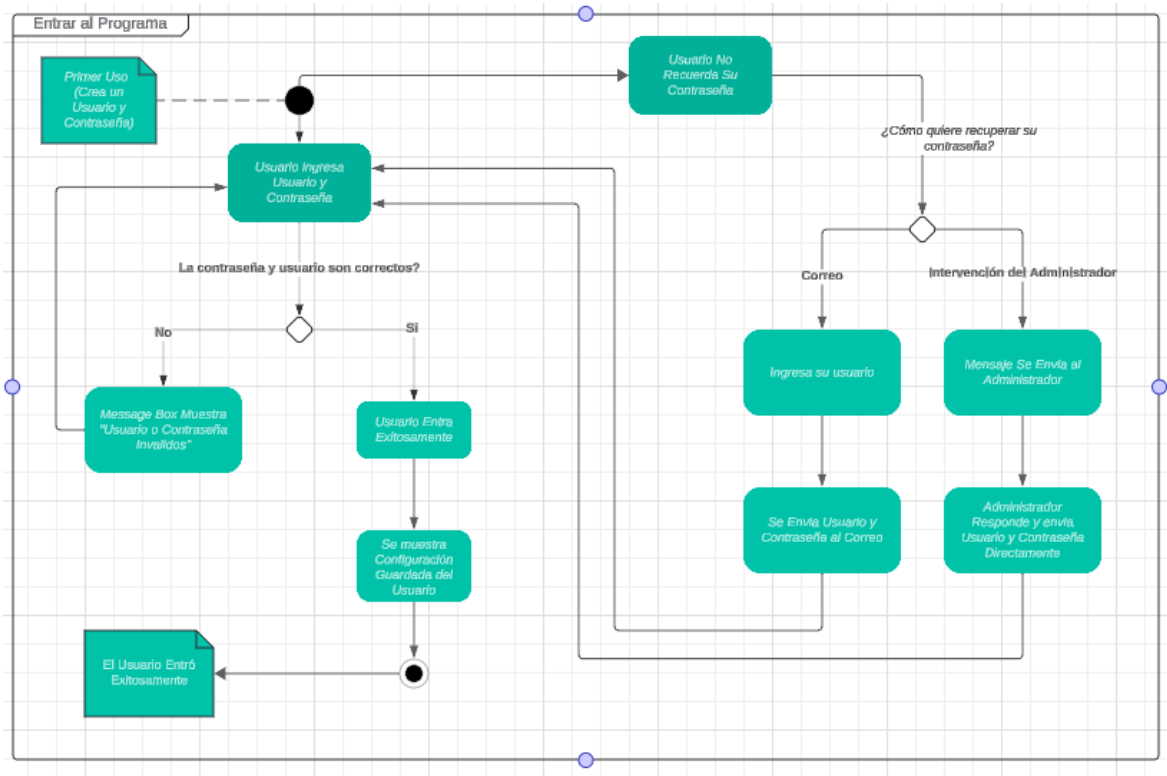
- *Fin del Error.*

No se actualizó la contraseña.

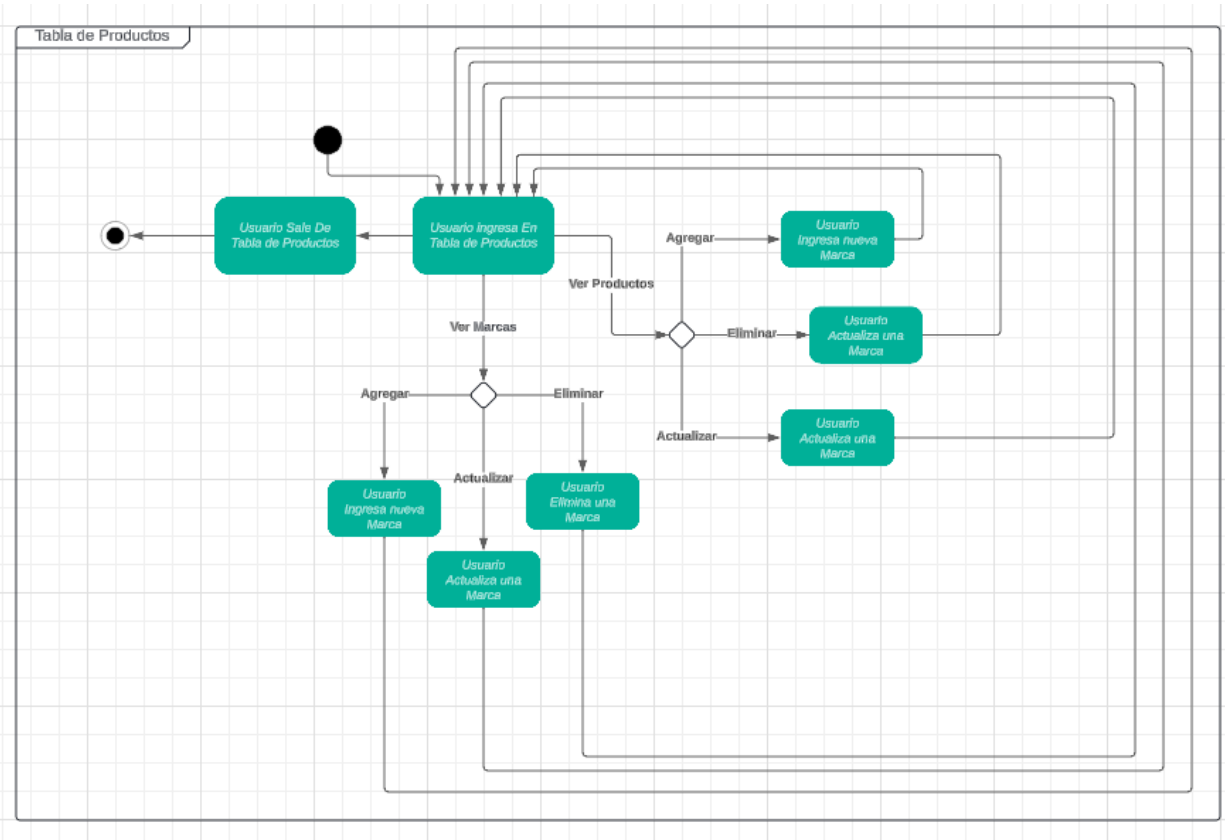
Solución: Verifique los datos han sido ingresados correctamente e intente de nuevo.

- Anexos:

Diagramas de Actividades; Entrar al Programa.



Diagramas de Actividades; Tabla de Productos.



Diagramas de Actividades; Tabla de Proveedores.

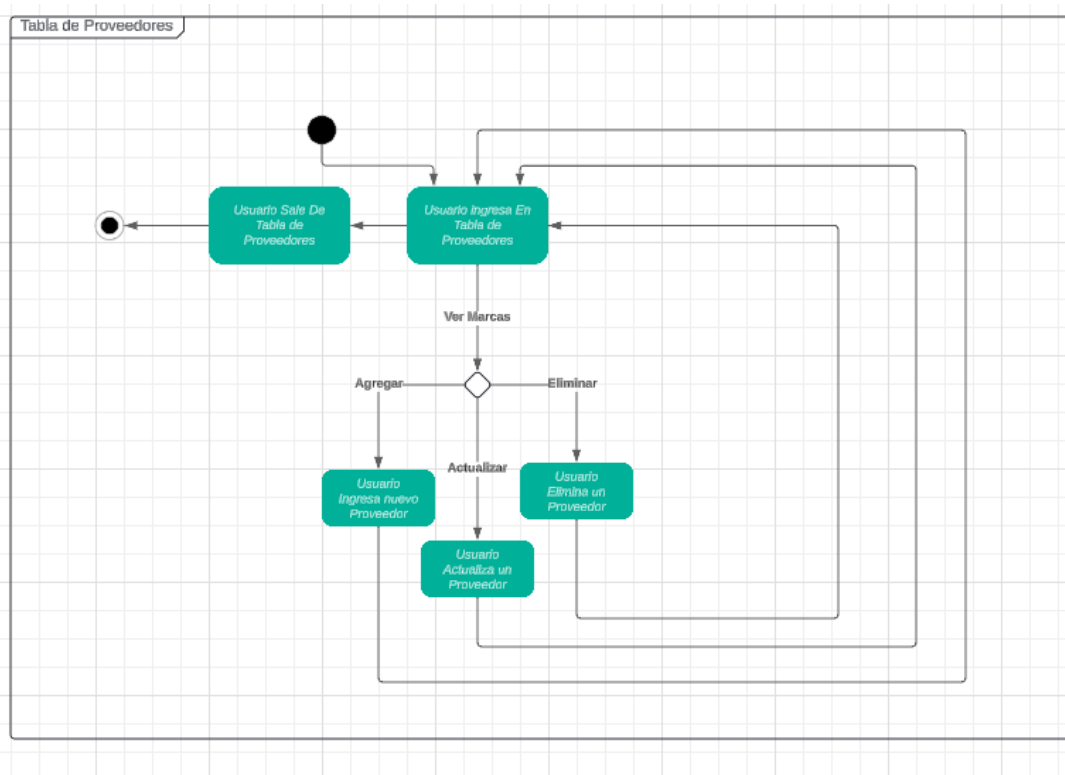


Diagrama de Casos de Uso; Programa de administración.

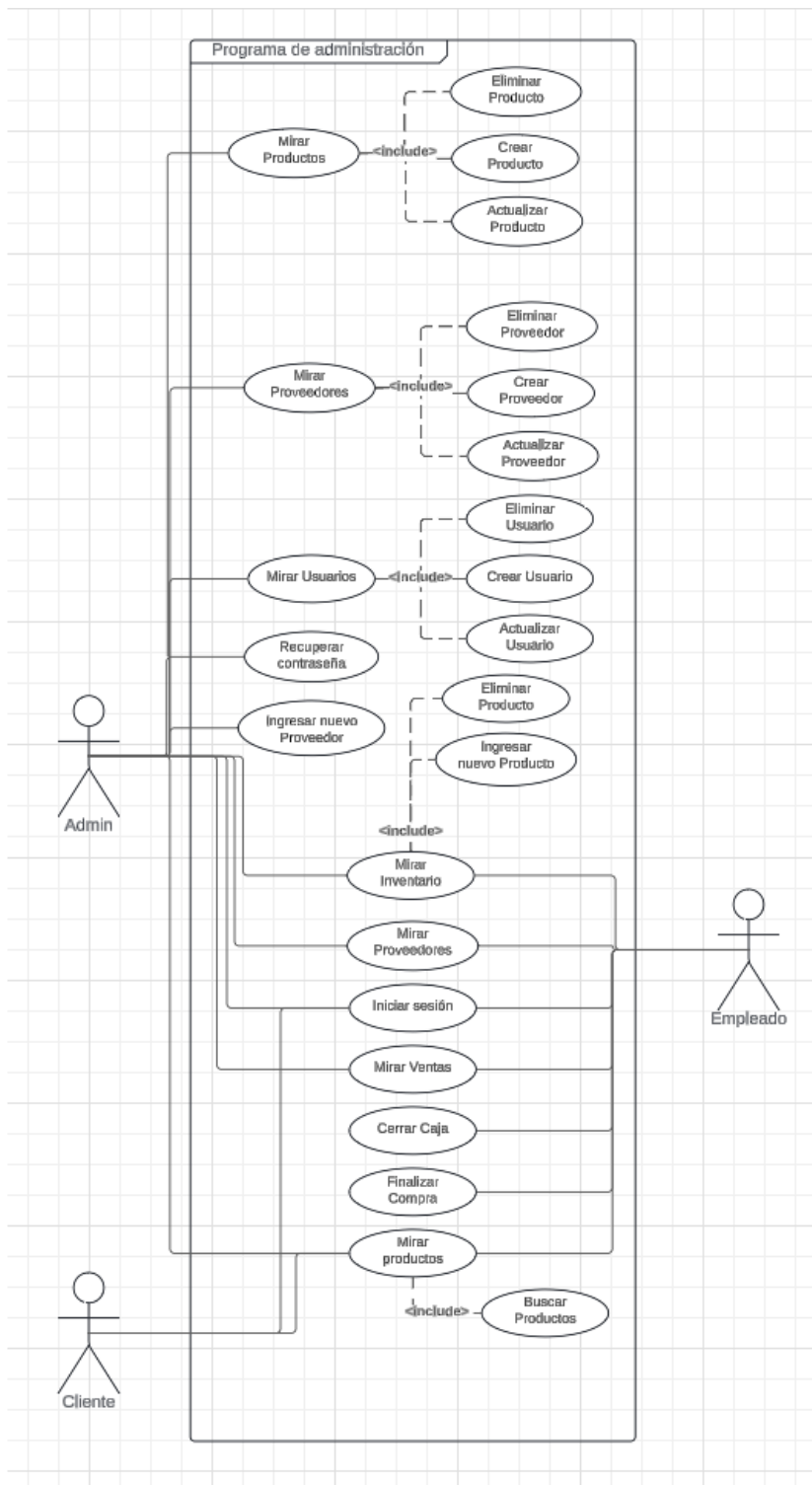


Diagrama de Clases; Tabla de Proveedores.

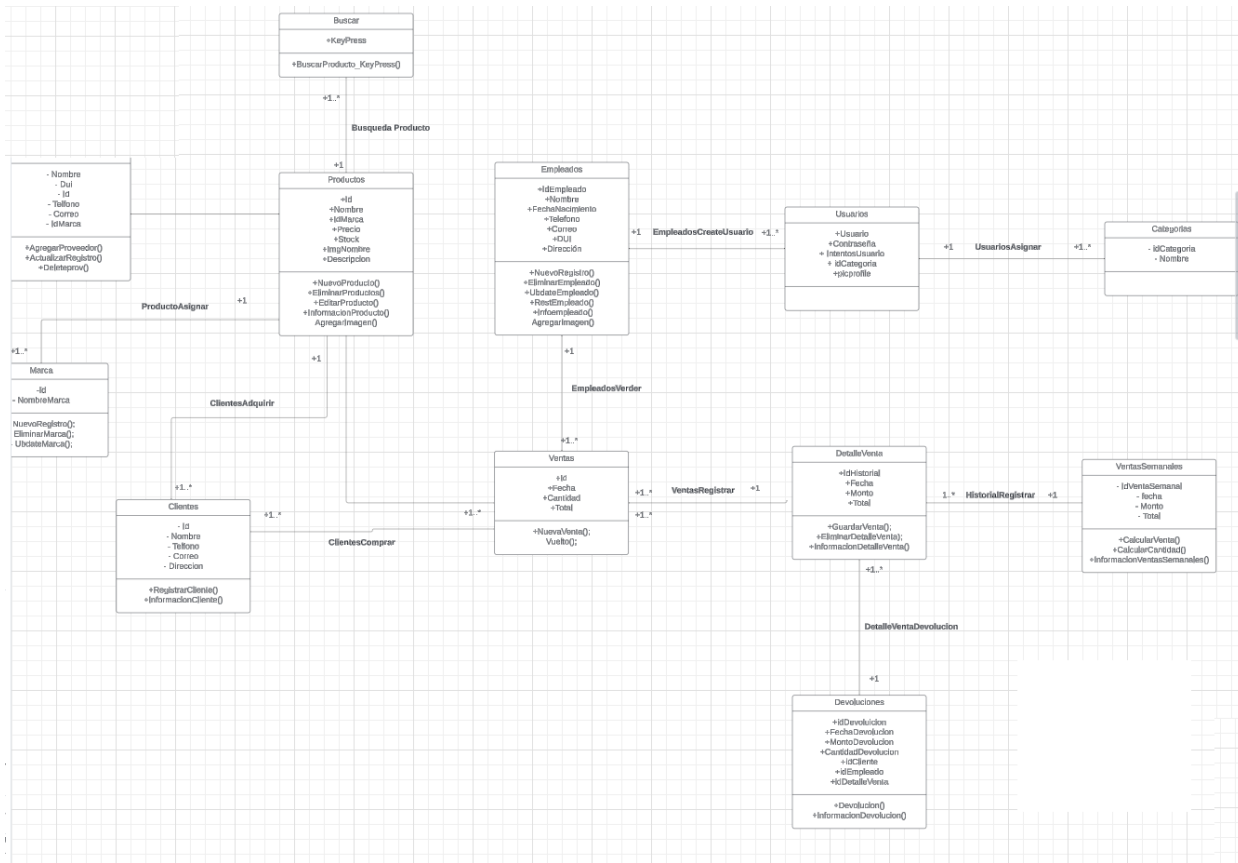
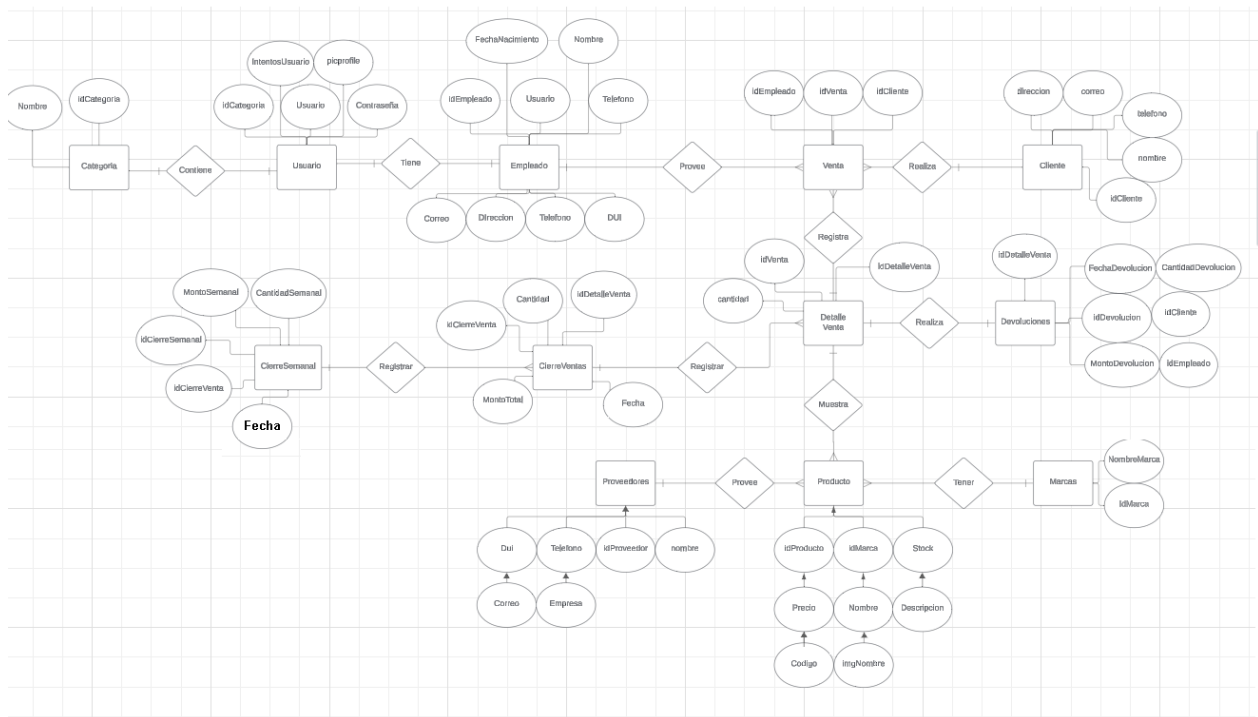


Diagrama Entidad Relación:



Modelo de Dominio

