

UNIVERSIDAD NACIONAL DE INGENIERIA INSTITUTO DE ESTUDIOS SUPERIORES

TRABAJO DE CURSO BASE DE DATOS I

TEMA: Prototipo de Aplicación de Inventario y Facturación para la Surtidora de Productos Varios ALE.

GRUPO:2M2-S

PROFESOR: ING. Gabriel Ignacio Toruño

INTEGRANTES:

- Cesar José Cerda
- Marlon Chamorro Tijerino
- Diego Romero Mena

FECHA: 28 de noviembre de 2023

Índice

INTRODUCCION	4
ANTECEDENTES	5
JUSTIFICACION	6
OBJETIVOS	7
I. Entorno Organizacional	8
Giro de Negocio	8
Misión	9
Visión	9
Valores	9
Estructura Organizacional	10
Procesos a automatizar	11
II. Modelo Conceptual	13
III. Modelo Lógico	14
IV. Modelo Físico	15
Script de creación de Base de Datos y tablas	15
Script de Vistas y Procedimientos almacenados	18
V. Diseño De La Aplicación	32
Explicación de las Tecnologías utilizadas	32
Capturas de Pantalla del Proceso de Venta y Facturación	32
VI. Conclusiones Y Recomendaciones	38
VIII. Anexos	39
Diccionario de Datos	39
Resto de Capturas	42

INTRODUCCION

En el presente se abordará la realización de un prototipo de sistema de venta e inventario para una Surtidora de productos varios, ubicada en el departamento de Granada.

Para esto se dará a conocer su entorno organizacional, tomando en cuenta su giro de negocio, misión, visión y valores de esta pequeña entidad.

Conocer cuáles serían las áreas en las que impactara la aplicación y los procesos manuales a mejorar.

Para realizar la aplicación se tendrá que aplicar la metodología adecuada para el desarrollo de una base de datos que permita controlar los procesos de esta entidad, con esta metodología se podrá llegar a obtener el diagrama entidad relación y el modelo relacional final de la base de datos y realizar sus normalizaciones de ser necesario, has llegar al modelo relacional final para el modelado físico de la base de datos y la aplicación

ANTECEDENTES

Antes de la automatización, las empresas llevaban registros contables y de inventario de forma manual utilizando libros de contabilidad y hojas de cálculo escritas a mano. Esta forma de registro era propensa a errores humanos y resultaba en una gestión lenta y tediosa de las facturas y el inventario.

A finales del siglo XIX, se inventaron las primeras máquinas registradoras para agilizar las transacciones en negocios minoristas y restaurantes.

Estas máquinas permitían a los comerciantes calcular automáticamente el importe de la venta y registrarla, pero no gestionaban inventario.

Con la aparición de las computadoras en la década de 1950, se comenzaron a desarrollar sistemas de facturación electrónica que permitían a las empresas generar facturas y realizar un seguimiento más eficiente de las transacciones.

Estos sistemas iniciales eran costosos y requerían equipos informáticos grandes.

A medida que las computadoras se volvieron más accesibles y económicas en la década de 1960, las empresas comenzaron a desarrollar sistemas de gestión de inventario electrónicos.

Estos sistemas permitían llevar un seguimiento en tiempo real de las existencias, agilizando las operaciones y reduciendo el riesgo de agotar productos.

A lo largo de las décadas de 1970 y 1980, se desarrollaron sistemas más avanzados que podían integrar la gestión de inventario y la facturación en una única plataforma.

Estos sistemas permitían a las empresas automatizar gran parte de sus procesos contables y comerciales.

Con la expansión de la tecnología de la información, los sistemas de facturación e inventario se volvieron más accesibles y asequibles para las pequeñas y medianas empresas.

JUSTIFICACION

Las empresas pueden optimizar sus procesos internos automatizando la gestión de inventario y la facturación, la mayor eficiencia del flujo de trabajo es el resultado de menores costos operativos y la eliminación del error humano.

A través de una facturación precisa y un mantenimiento de registros adecuado, los sistemas de facturación ayudan a las empresas a cumplir con las regulaciones fiscales y contables. Esto es crucial para evitar sanciones y problemas legales.

Al automatizar el control de inventario y la facturación, se reduce la carga de trabajo administrativa y se ahorra tiempo y recursos. Esto libera a los empleados para que puedan concentrarse en trabajos más estratégicos.

Las empresas pueden tomar mejores decisiones sobre compras, producción y estrategias comerciales gracias a la información en tiempo real de los sistemas sobre ventas e inventario disponible. Al prevenir errores en las facturas y garantizar la disponibilidad del producto, la precisión y la eficiencia en la facturación y la gestión del inventario conducen a una experiencia más gratificante para el cliente.

Los beneficios de utilizar sistemas de inventario y facturación son, en resumen, una mayor satisfacción del cliente, una disminución de costos y errores y el cumplimiento de las leyes fiscales.

Estos sistemas tienen un impacto positivo en una mayor eficiencia operativa, una gestión más eficaz de existencias y recursos y mejores decisiones comerciales.

OBJETIVOS

General: Implementar un prototipo de aplicación para la gestión de control de inventario, y ventas con facturación para la Surtidora Ale de productos varios ubicada en el departamento Granada.

Específicos: - Desarrollar un prototipo funcional de la aplicación.

- Realizar el modelo conceptual de Entidad-Relación haciendo uso de la metodología aprendida durante la asignatura de base de Datos

- Modelar el diagrama Relacional tomando en cuenta el Diagrama Entidad Relación, el cual servirá como estructura de nuestra Base De Datos para la aplicación del Negocio.

- Desarrollar el prototipo de Aplicación de escritorio utilizando las tecnologías C# en su editor de Visual Studio 2022, y el gestor de Base De Datos SQL Server 2019.

- Incorporar funciones de facturación, registro de ventas y compras de los Productos.

- Realizar pruebas de usabilidad del prototipo.

I. Entorno Organizacional

Giro de Negocio

Selección del Producto: Para empezar el negocio empieza por la selección de sus productos en este caso los tipos de productos que desea vender ya que es una Surtidora de productos varios se seleccionan productos para uso del hogar, alimenticios y aseo personal vendidos al por menor.

Adquisición de los productos: El negocio se encarga de comprar los productos a sus proveedores en este caso son distribuidoras más grandes ubicadas en los mercados del departamento de managua o bien puede ser a distribuidores ambulantes o vendedores de las diferentes marcas o categorías de los productos que este negocio ofrece.

Almacenamiento de los productos: Cuentan con un espacio adecuado y optimo para el almacén de sus productos. Teniendo un control de inventario a mano.

Distribución: Los productos son vendidos al por menor a clientes diversos del departamento o locación donde se encuentra el negocio. Llegando el cliente al local para obtener los productos que estos desean.

Marketing: Desarrolla estrategias de marketing para promocionar sus productos mediante sorteos de canastas básicas en temporadas celebres. Utiliza un medio de publicidad por radio y en el establecimiento mediante afiches de sus productos.

Gestión de Clientes: Ofrece un buen servicio al cliente y establece relaciones sólidas con sus compradores. La satisfacción del cliente es fundamental para el éxito a largo plazo del negocio.

Gestión Financiera: Lleva un control financiero riguroso de sus compras y ventas para asegurarse de que el negocio sea rentable. Esto implica la gestión de ingresos y costos.

Cumplimiento Legal y Regulatorio: Cumple con todas las regulaciones y requisitos legales para que el negocio pueda operar, cumpliendo así con licencias comerciales, permisos y cumplimientos de normativas.

Misión

En Surtidora Ale, nuestra misión es brindar a nuestros clientes productos de calidad que ofrezcan una amplia gama de productos que satisfagan sus necesidades diarias. Estamos comprometidos con la excelencia en el servicio al cliente, la eficiencia de la distribución y la construcción de relaciones sólidas con nuestros proveedores y clientes.

Visión

Ser reconocidos como la surtidora líder de diversos productos en nuestro departamento y barrio de ubicación, brindando soluciones versátiles y confiables para la venta de productos esenciales a nuestros clientes. Nos esforzamos por ampliar nuestro alcance y mantener nuestra posición sostenible en el mercado mediante la promoción de la innovación y la responsabilidad social en nuestras operaciones.

Valores

Calidad: Nos esforzamos por vender productos de calidad que cumplan incluso con los más altos estándares.

Servicio al cliente: Nuestros clientes son nuestra primera prioridad y trabajamos incansablemente para superar sus expectativas y brindar un servicio excepcional.

Integridad: Actuamos de forma ética y transparente en todas nuestras actividades y transacciones comerciales.

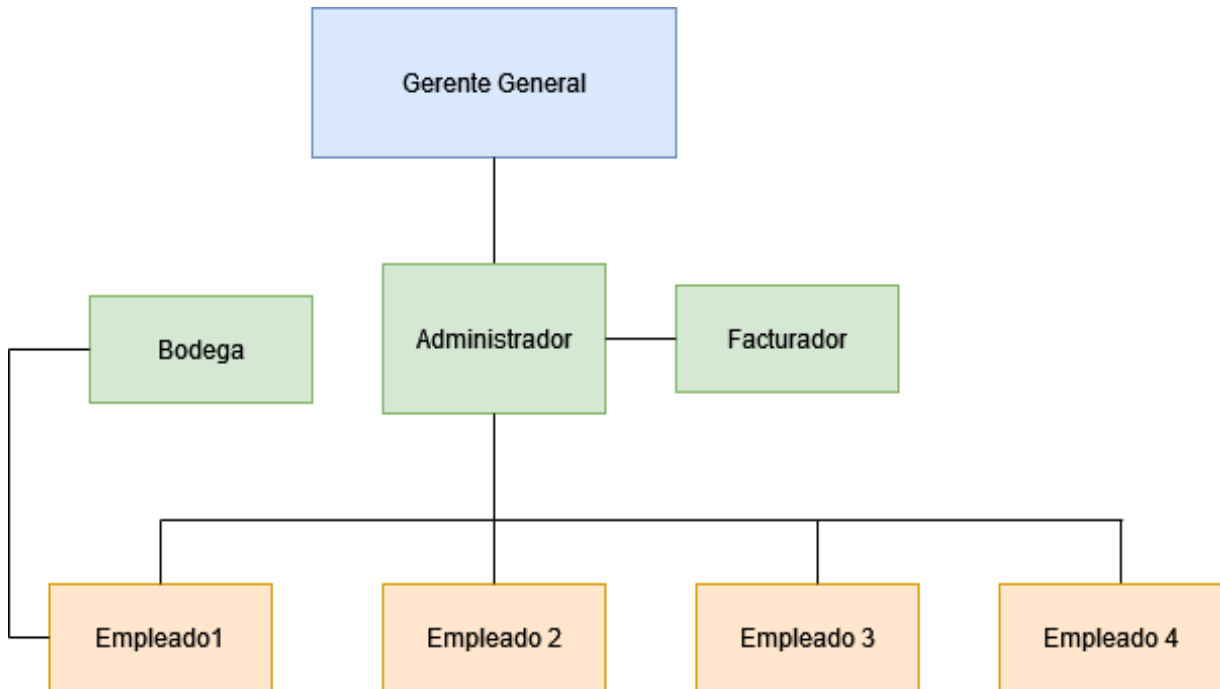
Colaboración: Promovemos un ambiente de trabajo en equipo, colaboración y comunicación abierta.

Innovación: Buscamos constantemente formas de mejorar y adaptarnos a las necesidades cambiantes del mercado.

Responsabilidad social: Somos conscientes de nuestro impacto en la comunidad y el medio ambiente y nos esforzamos por ser ciudadanos responsables.

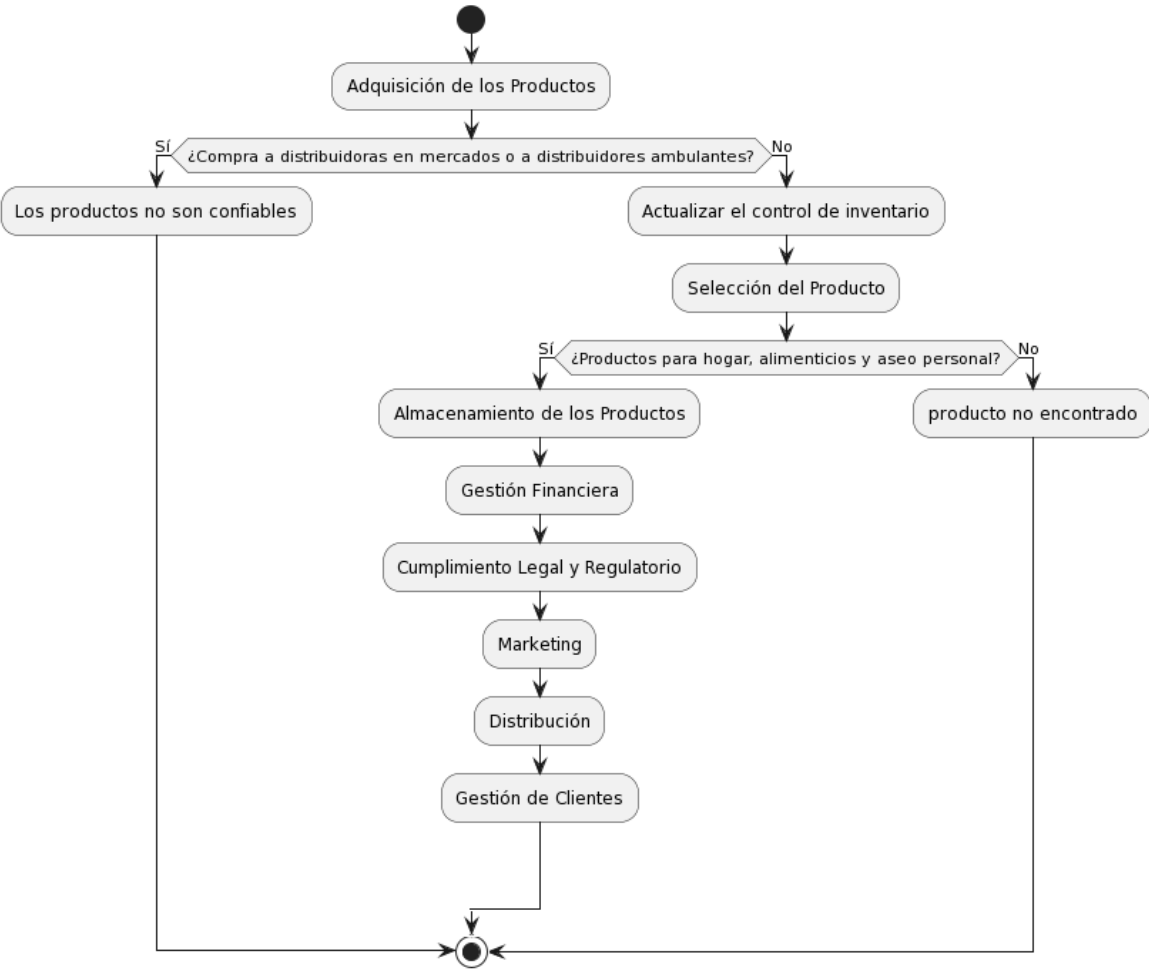
Eficiencia: Gestionamos nuestros recursos de manera efectiva para asegurar la rentabilidad y sostenibilidad del negocio a largo plazo.

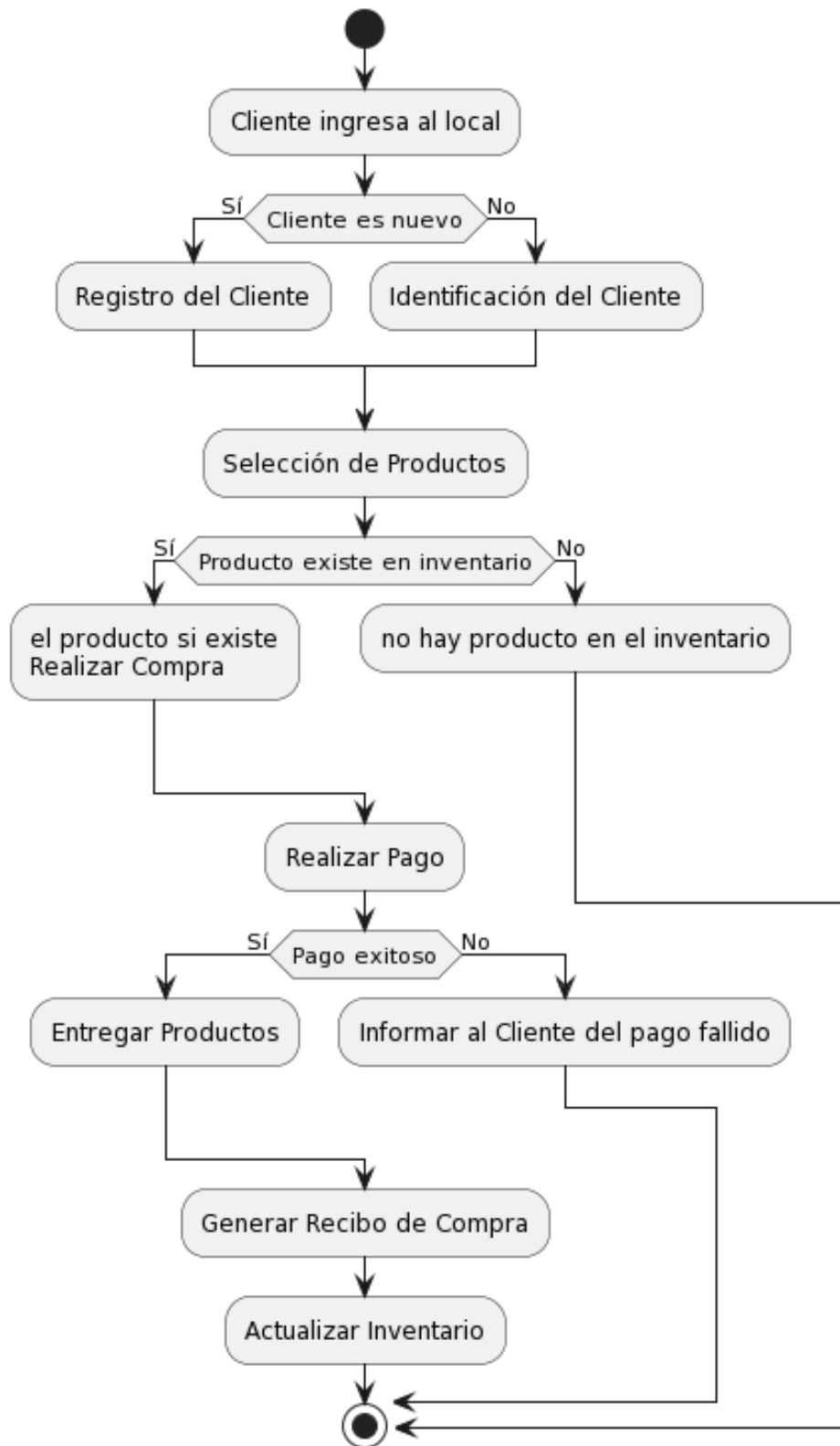
Estructura Organizacional



Dentro de las áreas que impactara el prototipo de aplicación tenemos el área de administración del negocio esto con el control del inventario con sus entradas con compras a diferentes proveedores y salidas con ventas a diversos clientes , así también el área de facturación con brindar el detalle comprobante para el cliente cuando uno de los empleados realice el despacho de los distintos productos que este llevara, teniendo un control de las ventas y compras realizadas por el negocio en cierto periodo y el detalle a quien fue realizado.

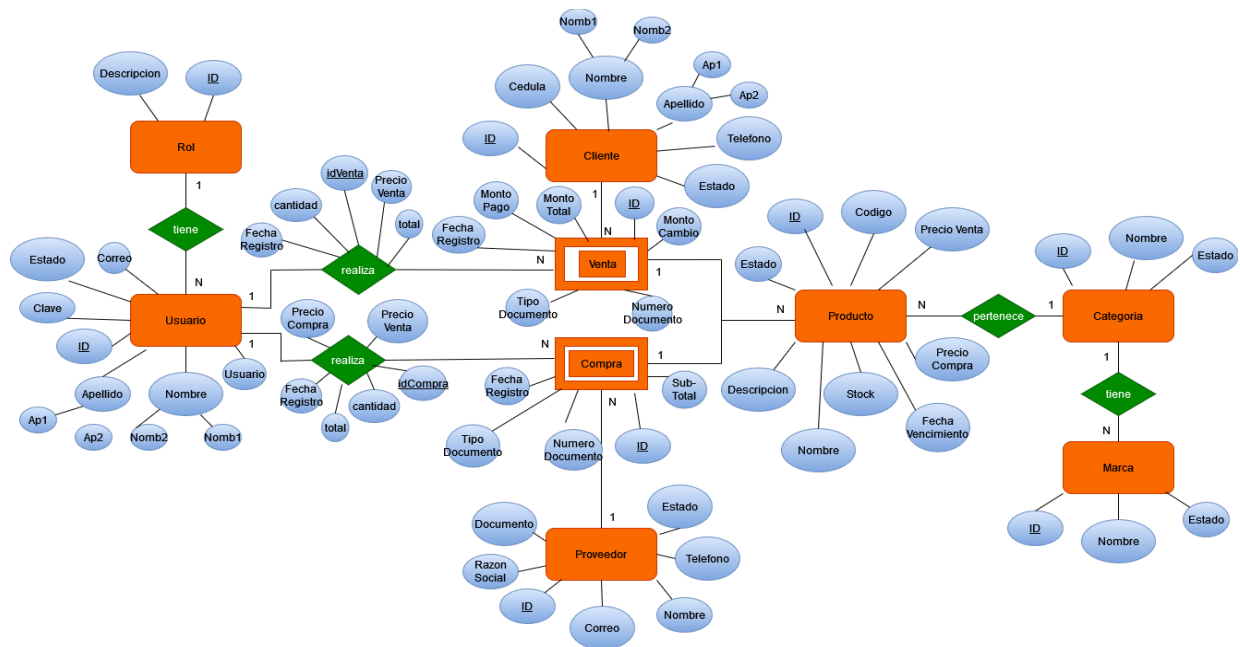
Procesos a automatizar





II. Modelo Conceptual

Se aplicó el modelado conceptual de manera abstracta para el diseño de la base de datos de dicho negocio, de esta manera identificando las entidades débiles y fuertes y las relaciones existentes entre ellas, la cardinalidad de las relaciones y los diferentes atributos que presenta cada entidad todo esto en base a los requerimientos del negocio, para poder implementar el prototipo de sistema de ventas, cabe destacar que la implementación de este diagrama se realizó utilizando la simbología adecuada para la realización de este.

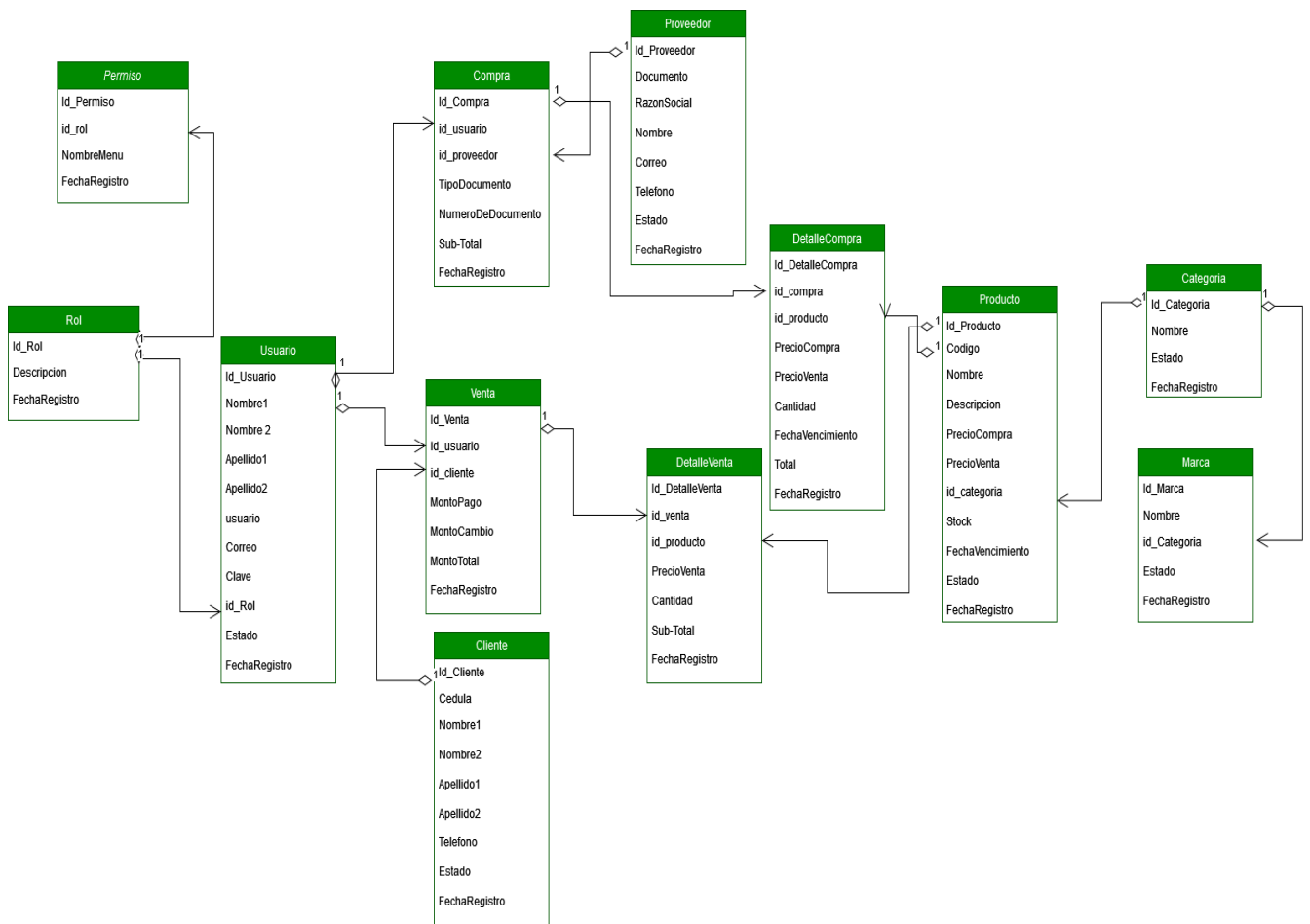


III.Modelo Lógico

En este modelo relacional se representan las entidades a manera de tablas, y las relaciones entre estas se establecen con una llave primaria y una llave foránea a la cual una entidad dependerá o estará relacionada con otra, también cada atributo de la entidad se representa a manera de columna dentro de las tablas

En este Diagrama Relacional podemos ver los procesos que realizara la Surtidora en su sistema asignando un rol de usuario el cual a su vez dependiendo de su rol realizara los procesos de compra y venta de los productos a los clientes, también se clasifica el catálogo de los productos a vender en categoría y cada categoría tiene una marca definida.

Como parte del proceso de abastecimiento se detallará la compra de los productos a manera de dos tablas Compra y Detalle de Compra y el Proceso de Venta a los clientes como salida de igual manera a detalle con la Tabla Ventas y DetalleDeVenta esto para tener un registro exacto de estos procesos y de esta manera se pueda realizar la reporteria necesaria para el negocio.



IV. Modelo Físico

Script de creación de Base de Datos y tablas

```
Create DataBase Sistema_SurtidoraAlejandra
```

```
use Sistema_SurtidoraAlejandra  
go
```

```
-----TABLA ROL-----
```

```
Create Table Rol(  
id_Rol int identity(1,1) primary key,  
Descripcion nvarchar(50),  
FechaRegistro datetime default getdate()  
)  
go
```

```
---TABLA PERMISO---
```

```
Create Table Permiso(  
id_Permiso int identity(1,1) primary key,  
id_rol int references Rol(id_Rol),  
NombreMenu nvarchar(70),  
FechaRegistro datetime default getdate()  
)  
go
```

```
-----TABLA PROVEEDOR-----
```

```
Create Table Proveedor(  
id_Proveedor int identity(1,1) primary key,  
Documento nvarchar(50),  
RazonSocial nvarchar(50),  
Nombre nvarchar(50),  
Correo nvarchar(20),  
Teléfono nvarchar(8),  
Estado bit,  
FechaRegistro datetime default getdate()  
)  
go
```

```
-----TABLA CLIENTE-----
```

```
Create Table Cliente(  
id_Cliente int identity(1,1) primary key,  
Cedula nvarchar(15),  
Nombre1 nvarchar(10),  
Nombre2 nvarchar(10),  
Apellido1 nvarchar(10),  
Apellido2 nvarchar(10),  
Telefono nvarchar(8),  
Estado bit,  
FechaRegistro datetime default getdate()  
)  
go
```

```
-----TABLA USUARIO-----
```

```
Create Table Usuario(  
id_Usuario int identity(1,1) primary key,  
Nombre1 nvarchar(10),  
Nombre2 nvarchar(10),  
Apellido1 nvarchar(10),
```

```

Apellido2 nvarchar(10),
usuario nvarchar(10),
Correo nvarchar(10),
Clave nvarchar(max),
id_rol int references Rol(id_Rol),
Estado bit,
FechaRegistro datetime default getdate()
)
go

----TABLA CATEGORIA----
Create Table Categoria(
id_Categoria int identity(1,1)primary key,
Nombre nvarchar(20),
Estado bit,
FechaRegistro datetime default getdate()
)
go

---TABLA MARCA---
Create Table Marca(
id_Marca int identity(1,1) primary key,
Nombre nvarchar(20),
id_categoria int references Categoria(id_Categoria),
Estado bit,
FechaRegistro datetime default getdate()
)

----TABLA PRODUCTO----
Create Table Producto(
id_Producto int identity(1,1)primary key,
Codigo nvarchar(20),
Nombre_Producto nvarchar(20),
Descripcion nvarchar(20),
PrecioCompra decimal(18,2) default 0,
PrecioVenta decimal(18,2) default 0,
id_categoria int references Categoria(id_Categoria),
id_marca int references Marca(id_Marca),
Stock int not null default 0,
FechaVencimiento date default '2000/01/01',
Estado bit,
FechaRegistro datetime default getdate()
)
go

----TABLA COMPRA---
Create Table Compra(
id_Compra int identity(1,1) primary key,
id_usuario int references Usuario(id_Usuario),
id_proveedor int references Proveedor(id_Proveedor),
TipoDocumento nvarchar(20),
NumeroDocumento nvarchar(20),
Sub_Total decimal(18,2),
FechaRegistro datetime default getdate()
)
go

---TABLA DETALLE DE COMPRA---

```



```

Create Table Detalle_Compra(
id_DetalleCompra int identity(1,1) primary key,
id_compra int references Compra(id_Compra),
id_producto int references Producto(id_Producto),
PrecioCompra decimal(18,2) default 0,
PrecioVenta decimal(18,2) default 0,
FechaVencimiento date default '2000/01/01',
Cantidad int not null,
Total decimal (18,2),
FechaRegistro datetime default getdate()
)
go

```

```

----TABLA VENTA----
Create Table Venta(
id_Venta int identity(1,1)primary key,
id_usuario int references Usuario(id_Usuario),
TipoDocumento nvarchar(20),
NumeroDocumento nvarchar(20),
id_cliente int references Cliente(id_Cliente),
MontoPago decimal(18,2),
MontoCambio decimal(18,2),
MontoTotal decimal(18,2),
FechaRegistro datetime default getdate()
)
go

```

```

---TABLA DETALLE DE VENTA----
Create Table Detalle_Venta(
id_DetalleVenta int identity(1,1) primary key,
id_venta int references Venta(id_Venta),
id_producto int references Producto(id_Producto),
PrecioVenta decimal(18,2),
Cantidad int not null,
Sub_Total decimal(18,2),
FechaRegistro datetime default getdate()
)
go

```

```

--TABLA DE DATOS DEL NEGOCIO--
Create Table Negocio(
id_Negocio int primary key,
Nombre varchar(80) not null,
RUC varchar(80)not null,
Direccion varchar(80)not null,
Logo varbinary(max) null
)

```

Script de Vistas y Procedimientos almacenados

---Procedimiento de Ingreso de Proveedor---

```
Create or alter Procedure Ingresar_Proveedor(
@Documento nVarchar(30),
@RazonSocial nVarchar(30),
@Nombre nvarchar(30),
@Correo nvarchar(30),
@Telefono nvarchar(9),
@Estado bit,
@Resultado int output,
@Mensaje varchar(500) output
)as
begin
    set @Resultado=0
    Declare @IDPERSONA INT
    if not exists(Select * from Proveedor where Documento=@Documento)
    begin
        insert into
Proveedor(Documento,RazonSocial,Nombre,Correo,Telefono,Estado)values(@Documento,@Raz
onSocial,@Nombre,@Correo,@Telefono,@Estado)
        set @Resultado=SCOPE_IDENTITY()
    end
    else
        set @Mensaje='El Numero de Documento ya existe'
end
go
```

----Procedimiento Para Editar Proveedor---

```
Create or alter Procedure Editar_Proveedor(
@id_Proveedor int,
@Documento nVarchar(30),
@RazonSocial nVarchar(30),
@Nombre nvarchar(30),
@Correo nvarchar(30),
@Telefono nvarchar(9),
@Estado bit,
@Resultado int output,
@Mensaje varchar(500) output
)
as
begin
    set @Resultado=1
    declare @IDPERSONA INT
    if not exists(Select * from Proveedor where Documento=@Documento and
id_Proveedor != @id_Proveedor)
    begin
        update Proveedor set
Documento=@Documento,
RazonSocial=@RazonSocial,
Nombre=@Nombre,
Correo=@Correo,
Telefono=@Telefono,
Estado=@Estado
where id_Proveedor=@id_Proveedor
    end
    else
        begin
```

```

        set @Resultado=0
        set @Mensaje='El numero de Documento ya existe'
    end
end
go

---Procedimeinto Para Eliminar Proveedor--
Create Procedure Eliminar_Proveedor(
    @id_Proveedor int,
    @Resultado bit output,
    @Mensaje varchar(500) output
)
as
begin
    set @Resultado=1
    if not exists(select * from Proveedor
    inner join Compra on Proveedor.id_Proveedor=Compra.id_Proveedor
    where Proveedor.id_Proveedor=@id_Proveedor)
    begin
        delete top(1) from Proveedor where id_Proveedor=@id_Proveedor
    end
    else
    begin
        set @Resultado=0
        set @Mensaje='El Proveedor esta relacionado asociado a una compra'
    end
end
go

--Inserciones para acessos--
--Los roles del sistema--
insert into Rol(Descripcion)values('Administrador')
insert into Rol(Descripcion)values('Bodeguero')
insert into Rol(Descripcion)values('Facturador')
select * from Rol
delete from Rol
use Sistema_SurtidoraAlejandra
go
select * from Permiso
---Permisos del sistema---
---Rol de Administrador---
insert into
Permiso(id_rol,NombreMenu)values(10,'mantenedorToolStripMenuItem'),(10,'proveedoresT
oolStripMenuItem'),(10,'clientesToolStripMenuItem'),(10,'comprasToolStripMenuItem'),
(10,'ventasToolStripMenuItem'),(10,'reportesToolStripMenuItem'),(10,'usuariosToolStr
ipMenuItem')
---Rol de Facturador---
insert into
Permiso(id_rol,NombreMenu)values(12,'clientesToolStripMenuItem'),(12,'ventasToolStri
pMenuItem'),(12,'reportesToolStripMenuItem')
---Rol de Bodeguero--
insert into
Permiso(id_rol,NombreMenu)values(11,'mantenedorToolStripMenuItem'),(11,'proveedoresT
oolStripMenuItem'),(11,'comprasToolStripMenuItem')
go

```

```

--Procedimiento para Ingresar Usuarios--
create or alter procedure Insertar_Usuarios(
@Nombre1 nvarchar(max),
@Nombre2 nvarchar(max),
@Apellido1 nvarchar(max),
@Apellido2 nvarchar(max),
@Usuario nvarchar(max),
@Correo nvarchar(max),
@clave nvarchar(max),
@id_Rol int,
@Estado bit,
@idUsuarioResultado int output,
@Mensaje varchar(500) output
)
as
begin
    set @idUsuarioResultado=0
    set @Mensaje=''
    if not exists(select * from Usuario where Correo=@Correo)
    begin
        insert into
        Usuario(Nombre1,Nombre2,Apellido1,Apellido2,usuario,Correo,Clave,id_rol,Estado)

        values(@Nombre1,@Nombre2,@Apellido1,@Apellido2,@Usuario,@Correo,@clave,@id_Rol,@Esta
do)

        set @idUsuarioResultado=SCOPE_IDENTITY()
    end
    else
        set @Mensaje='No se puede repetir el mismo correo para mas de un Usuario'

end
go

--Procedimiento para Editar Usuario---
create or alter procedure Editar_Usuarios(
@idUsuario int,
@Nombre1 nvarchar(max),
@Nombre2 nvarchar(max),
@Apellido1 nvarchar(max),
@Apellido2 nvarchar(max),
@Usuario varchar(max),
@Correo nvarchar(max),
@Clave varchar(max),
@id_Rol int,
@Estado bit,
@Respuesta bit output,
@Mensaje varchar(500) output
)
as
begin
    set @Respuesta=0
    set @Mensaje=''
    if not exists(select * from Usuario where @Correo=@Correo and
id_Usuario!=@idUsuario)
    begin

```

```

        update Usuario set
            Nombre1=@Nombre1,
            Nombre2=@Nombre2,
            Apellido1=@Apellido1,
            Apellido2=@Apellido2,
            usuario=@Usuario,
            clave=@Clave,
            id_Rol=@id_Rol,
            Estado=@Estado
        where id_Usuario=@idUsuario
    set @Respuesta=1
    end
else
    set @Mensaje='No se puede repetir el mismo Correo para mas de un Usuario'

end
go
--Procedimiento para Eliminar Usuario---
create procedure Eliminar_Usuarios(
    @idUsuario int,
    @Respuesta bit output,
    @Mensaje varchar(500) output
)
as
begin
    set @Respuesta=0
    set @Mensaje=''
    declare @pasoreglas bit=1
    if exists(select * from Compra c
        inner join Usuario u on u.id_Usuario=c.id_usuario
        where u.id_Usuario=@idUsuario)
    begin
        set @pasoreglas=0
        set @Respuesta=0
        set @Mensaje=@Mensaje+'No se puede eliminar el Usuario porque esta asociado
a una compra\n'
    end
    if exists(Select * from Venta v
        inner join Usuario u on u.id_Usuario=v.id_usuario
        where u.id_Usuario=@idUsuario)
    begin
        set @pasoreglas=0
        set @Respuesta=0
        set @Mensaje=@Mensaje+'No se puede eliminar el Usuario porque esta asociado
a una venta\n'
    end
    if(@pasoreglas=1)
    begin
        delete from Usuario where id_Usuario=@idUsuario
        set @Respuesta=1
    end
end
end

```

```

---Procedimiento para registrar un cliente
create or alter procedure Registrar_Cliente
@Cedula nvarchar(15),
@Nombre1 nvarchar(max),
@Nombre2 nvarchar(max),
@Apellido1 nvarchar(max),
@Apellido2 nvarchar(max),
@Telefono nvarchar(8),
@Estado bit,
@Resultado int output,
@Mensaje varchar(500) output
as
begin
set @Resultado=0
declare @IDPERSONA int
if not exists(select * from Cliente Where Cedula=@Cedula)
begin
insert into Cliente(Cedula,Nombre1,Nombre2,Apellido1,Apellido2,Telefono,Estado)
values (@Cedula,@Nombre1,@Nombre2,@Apellido1,@Apellido2,@Telefono,@Estado)
set @Resultado=SCOPE_IDENTITY()
end
else
set @Mensaje='No se Puede Repetir el Numero de Cedula de un Cliente'
end
go

```

```

---Procedimiento para editar un cliente---
Create or alter Procedure Editar_Cliente
@idCliente int,
@Cedula nvarchar(15),
@Nombre1 nvarchar(max),
@Nombre2 nvarchar(max),
@Apellido1 nvarchar(max),
@Apellido2 nvarchar(max),
@Telefono nvarchar(8),
@Estado bit,
@Resultado int output,
@Mensaje varchar(500) output
as
begin
set @Resultado=1
declare @IDPERSONA int
if not exists(select * from Cliente where Cedula=@Cedula and id_Cliente!=@idCliente)
begin
update Cliente set
Cedula=@Cedula,
Nombre1=@Nombre1,
Nombre2=@Nombre2,
Apellido1=@Apellido1,
Apellido2=@Apellido2,
Telefono=@Telefono,
Estado=@Estado
where id_Cliente=@idCliente
end
else
begin
set @Resultado=0

```

```

set @Mensaje='El numero de Cedula ya existe'
end
end
go

---Procedimiento para eliminar un cliente----
Create Procedure Eliminar_Cliente(
    @idCliente int,
    @Resultado bit output,
    @Mensaje varchar(500) output
)
as
begin
    set @Resultado=1
    if not exists(select * from Cliente
    inner join Venta on Cliente.id_Cliente=Venta.id_cliente
    where Cliente.id_Cliente=@idCliente)
    begin
        delete top(1) from Cliente where id_Cliente=@idCliente
    end
    else
    begin
        set @Resultado=0
        set @Mensaje='El Cliente esta relacionado asociado a una venta'
    end
end
go

----Procedimientos almacenado para categorias--
alter table Categoria
alter column Nombre nvarchar(max)
go

Create or alter Procedure Ingresar_Categoria
    @Nombre nvarchar(max),
    @Estado bit,
    @Resultado int output,
    @Mensaje varchar(5005) output
as
begin
    set @Resultado=0
    if not exists(Select * from Categoria where Nombre=@Nombre)
    begin
        insert into Categoria(Nombre,Estado)values(@Nombre,@Estado)
        set @Resultado=SCOPE_IDENTITY()
    end
    else
    set @Mensaje='No se puede repetir el nombre de la categoria'
end
go

Create or alter procedure Editar_Categoria
    @idCategoria int,
    @Nombre nvarchar(max),
    @Estado bit,
    @Resultado bit output,
    @Mensaje varchar(500) output
as

```

```

begin
set @Resultado=1
if not exists(Select * from Categoria where Nombre=@Nombre and
id_Categoria!=@idCategoria)
update Categoria set
Nombre=@Nombre,
Estado=@Estado
where id_Categoria=@idCategoria
else
begin
set @Resultado=0
set @Mensaje='No se puede repetir el nombre de la categoria'
end
end
go

create Procedure Eliminar_Categoria
@idCategoria int,
@Resultado bit output,
@Mensaje varchar(500)output
as
begin
set
@Resultado=1
if not exists(Select * from Categoria
inner join Producto on Categoria.id_Categoria=Producto.id_categoria
where Categoria.id_Categoria=@idCategoria)
begin
delete top(1) from Categoria where id_Categoria=@idCategoria
end
else
begin
set @Resultado=0
set @Mensaje='La categoria esta relacionada a un producto'
end
end
go

```

```

--Procedimientos almacenados para Marca--
Create Procedure Ingresar_Marca
@Nombre nvarchar(20),
@idCateg int,
@Estado bit,
@Respuesta bit output,
@Mensaje varchar(500) output
as
begin
set @Respuesta=0
set @Mensaje=''
if not exists(select * from Marca where Nombre=@Nombre)
begin
insert into Marca(Nombre,id_categoria,Estado)
values(@Nombre,@idCateg,@Estado)

```



```

        set @Respuesta=SCOPE_IDENTITY()
    end
else
    set @Mensaje='No se puede repetir el mismo nombre para una marca'

end
go

Create or alter Procedure Editar_Marca
@idMarca int,
@Nombre nvarchar(max),
@idCateg int,
@Estado bit,
@Resultado bit output,
@Mensaje varchar(500) output
as
begin
set @Resultado=1
if not exists(Select * from Marca where Nombre=@Nombre and id_Marca!=@idMarca)
update Marca set
Nombre=@Nombre,
Estado=@Estado,
id_categoria=@idCateg
where id_Marca=@idMarca
else
begin
set @Resultado=0
set @Mensaje='No se puede repetir el nombre de la marca'
end
end
go

Create or alter Procedure Eliminar_Marca
@idMarca int,
@Resultado bit output,
@Mensaje varchar(500)output
as
begin
set
@Resultado=1
if not exists(Select * from Marca
inner join Producto on Marca.id_Marca=Producto.id_marca
where Marca.id_Marca=@idMarca)
begin
delete top(1) from Marca where id_Marca=@idMarca
end
else
begin
set @Resultado=0
set @Mensaje='La Marca No se puede eliminar porque esta asociada a un producto'
end
end
go

alter table Producto
alter Column Descripcion nvarchar(max)
go

```

---Procedimientos almacenados para productos---

```
-----  
select * from Producto  
go  
create or alter procedure Ingresar_Productos  
@Codigo nvarchar(max),  
@NombreProducto nvarchar(max),  
@idCategoria int,  
@idMarca int,  
@Descripcion nvarchar(max),  
@Estado bit,  
@Resultado bit output,  
@Mensaje varchar(500) output  
as  
begin  
set @Resultado=0  
if not exists(select * from Producto where Codigo=@Codigo)  
begin  
insert into  
Producto(Codigo,Nombre_Producto,id_categoria,id_marca,Descripcion,Estado)  
values(@Codigo,@NombreProducto,@idCategoria,@idMarca,@Descripcion,@Estado)  
set @Resultado=SCOPE_IDENTITY()  
end  
else  
set @Mensaje='Ya existe un producto con el mismo Codigo'  
end
```

go

```
-----  
create or alter procedure Editar_Producto  
@idProducto int,  
@Codigo nvarchar(max),  
@Nombre nvarchar(max),  
@idCategoria int,  
@idMarca int,  
@Descripcion nvarchar(max),  
@Estado bit,  
@Resultado bit output,  
@Mensaje varchar(500) output  
as  
begin  
set @Resultado=1  
if not exists(Select * from Producto where Codigo=@Codigo and  
id_Producto!=@idProducto)  
update Producto set  
Codigo=@Codigo,  
Nombre_Producto=@Nombre,  
Descripcion=@Descripcion,  
id_categoria=@idCategoria,  
id_marca=@idMarca  
where id_Producto=@idProducto  
else  
begin  
set @Resultado=0  
set @Mensaje='Ya existe un producto con el mismo codigo'  
end  
end  
go
```

```

-----
Create Procedure Eliminar_Producto
@idProducto int,
@Respuesta bit output,
@Mensaje varchar(500)output
as
begin
set @Respuesta=0
set @Mensaje=''
declare @pasoreglas bit=1
if exists (Select * from Detalle_Compra dc
inner join Producto p on p.id_Producto=dc.id_producto
where p.id_Producto=@idProducto)
begin
set @pasoreglas=0
set @Respuesta=0
set @Mensaje=@Mensaje+'No se puede eliminar el producto porque esta relacionado a
una compra\n'
end
if exists (Select * from Detalle_Venta dv
inner join Producto p on p.id_Producto=dv.id_producto
where p.id_Producto=@idProducto)
begin
set @pasoreglas=0
set @Respuesta=0
set @Mensaje=@Mensaje+'No se puede eliminar el prodcuto porque esta asociado a una
Venta\n'
end
if(@pasoreglas=1)
begin
delete from Producto where id_Producto=@idProducto
set @Respuesta=1
end
end
-----
go

```

---Procesos de abastecimiento de productos en a travez de las compras ---
 --esta es una tabla temporal como un tipo de parametro para procedimiento
 almacenado----

```

create type [dbo].[eDetalleCompr]as table(
[id_Producto]int null,
[PrecioCompra]decimal(10,2) null,
[PrecioVenta]decimal(10,2)null,
[FechaVencimiento]date,
[Cantidad]int null,
[Total]decimal(10,2)null
)
go

```

---Procediiento para resgstrar una compra y a la vez su detalle como lista de
 parametros

--cabe recalacar que estamos usando la funcion transacition por medio de
 transacciones ----

```

create or alter procedure RegistrarCompraa(
@idusuario int,

```

```

@idproveed int,
@TipoDeDocumento nvarchar(max),
@NumeroDocumento nvarchar(max),
@MontoTotal decimal(10,2),
@DetalleCompra [eDetalleCompr] readonly,
@Resultado bit output,
@Mensaje varchar(500) output
)
as
begin
    begin try
        declare @id_Compra int=0
        set @Resultado=1
        set @Mensaje=''

        begin transaction registro
            insert into
Compra(id_usuario,id_Proveedor,TipoDocumento,NumeroDocumento,Sub_Total)

                values(@idusuario,@idproveed,@TipoDeDocumento,@NumeroDocumento,@MontoTotal)

                set @id_Compra= SCOPE_IDENTITY()

            insert into
Detalle_Compra(id_compra,id_producto,PrecioCompra,PrecioVenta,FechaVencimiento,Canti
dad,Total)

                select
@id_Compra,id_Producto,PrecioCompra,PrecioVenta,FechaVencimiento,Cantidad,Total from
@DetalleCompra

                update p set p.Stock=p.Stock+dc.Cantidad,
                p.PrecioCompra=dc.PrecioCompra,
                p.PrecioVenta=dc.PrecioVenta,
                p.FechaVencimiento=dc.FechaVencimiento
                from Producto p
                inner join @DetalleCompra dc on
dc.id_Producto=p.id_Producto
            commit transaction registro
        end try
        begin catch
            set @Resultado=0
            set @Mensaje=ERROR_MESSAGE()
            rollback transaction registro
        end catch
    end
go

----Para el correlativo de la Compra---
select count(*)+1 from Compra
----Proceso para Obtener el detalle de la compra----
select Nombre_Producto,Detalle_Compra.PrecioCompra,Cantidad,Total
from Detalle_Compra
inner join Producto on Detalle_Compra.id_producto=Producto.id_Producto
where id_compra=2
----Proceso para obtener compra
Select c.id_Compra,
u.Nombre1,
pr.Documento,pr.RazonSocial,

```

```

c.TipoDocumento,c.NumeroDocumento,c.Sub_Total,convert(char(10),c.FechaRegistro,103)[
FechaCreacion]
from Compra c
inner join Usuario u on u.id_Usuario=c.id_usuario
inner join Proveedor pr on pr.id_Proveedor=c.id_Proveedor
where c.NumeroDocumento='0002'

---Proceso para obtener venta de venta---
select v.id_Venta,
u.Nombre1,c.Cedula,c.Nombre1 as NombreCliente,
v.TipoDocumento,v.NumeroDocumento,
v.MontoPago,v.MontoCambio,
v.MontoTotal,convert(char(10),v.FechaRegistro,103)[FechaCreacion]
from Venta v
inner join Usuario u on u.id_Usuario=v.id_usuario
inner join Cliente c on c.id_Cliente=v.id_cliente
where NumeroDocumento='0001'
---proceso para obtener el detalle de venta---
Select p.Nombre_Producto,dv.PrecioVenta,dv.Cantidad,dv.Sub_Total
from Detalle_Venta dv
inner Join Producto p on p.id_Producto=dv.id_producto
where dv.id_venta=1

```

----Proceso de Ventas con tabla temporal todo esto para registrar la venta---

```

create type [dbo].[eDetalleVenta]as table(
[id_producto]int null,
[PrecioVenta]decimal(10,2)null,
[Cantidad]int null,
[Sub_Total]decimal(10,2)null
)
go

CREATE OR ALTER PROCEDURE RegistrarVenta(
    @idusuario INT,
    @TipoDeDocumento NVARCHAR(MAX),
    @NumeroDocumento NVARCHAR(MAX),
    @idcliente INT,
    @MontoPago DECIMAL(10, 2),
    @Montocambio DECIMAL(10, 2),
    @MontoTotal DECIMAL(10, 2),
    @DetalleVenta [eDetalleVenta] READONLY,
    @Resultado BIT OUTPUT,
    @Mensaje VARCHAR(500) OUTPUT
)
AS
BEGIN
    BEGIN TRY
        DECLARE @idVenta INT = 0
        SET @Resultado = 1
        SET @Mensaje = ''

        -- Validar la fecha de vencimiento
        IF not EXISTS (
            SELECT 1

```

```

        FROM @DetalleVenta dv
        JOIN Producto p ON dv.id_producto = p.id_Producto
        WHERE p.FechaVencimiento < GETDATE()
    )
    BEGIN TRANSACTION registro
    INSERT INTO Venta (id_usuario, TipoDocumento, NumeroDocumento, id_cliente,
MontoPago, Montocambio, MontoTotal)
    VALUES (@idusuario, @TipoDeDocumento, @NumeroDocumento, @idcliente,
@MontoPago, @Montocambio, @MontoTotal)

    SET @idVenta = SCOPE_IDENTITY()

    INSERT INTO Detalle_Venta (id_venta, id_producto, PrecioVenta,
Cantidad, Sub_Total)
    SELECT @idVenta, id_producto, PrecioVenta, Cantidad, Sub_Total FROM
@DetalleVenta

    COMMIT TRANSACTION registro
END TRY
BEGIN CATCH
    SET @Resultado = 0
    SET @Mensaje = 'No se puede realizar la venta. La fecha de vencimiento de
al menos un producto es mayor a la fecha actual.'
    ROLLBACK TRANSACTION registro
END CATCH
END

go

CREATE or alter VIEW VistaVentaDetalle AS
SELECT
    C.NumeroDocumento AS NumeroDocumento,
    C.TipoDocumento AS TipoDeDocumento,
    p.Cedula as Cedula,
    P.Nombre1 AS NombreCliente,
    U.Nombre1 AS NombreDelFacturador,
    PC.Nombre_Producto AS NombreProducto,
    DC.PrecioVenta as Precio,
    DC.Cantidad as Cantidad,
    C.MontoTotal AS MontoTotal,
    C.MontoPago as MontoPago,
    C.MontoCambio as Cambio
FROM
    Venta C
JOIN
    Usuario U ON C.id_usuario = U.id_Usuario
JOIN
    Cliente P ON C.id_cliente = P.id_Cliente
JOIN
    Detalle_Venta DC ON C.id_Venta = DC.id_venta
JOIN
    Producto PC ON DC.id_producto = PC.id_Producto
WHERE
    C.NumeroDocumento =NumeroDocumento
go

```

```

----Procedimiento para generar el reporte de Compra
create or alter procedure ReporteCompras(
@fechaCreacion varchar(10),
@fechaFin varchar(10),
@idproveedor int
)
as
begin
set dateformat dmy;
select
convert(char(10),c.FechaRegistro,103)[FechaCreacion],c.TipoDocumento,c.NumeroDocumen
to,c.Sub_Total,
u.Nombre1[NombreUsuario],
pr.Documento[DocumentoProveedor],pr.RazonSocial,
p.Codigo[CodigoProducto],p.Nombre_Producto[NombreProducto],mc.Nombre[Marca],ca.Nombr
e[Categoria],p.FechaVencimiento[FechaVencimiento],dc.PrecioCompra,dc.PrecioVenta,dc.
Cantidad,dc.Total[Total]
from Compra c
inner join Usuario u on u.id_usuario=c.id_usuario
inner join Proveedor pr on pr.id_Proveedor=c.id_proveedor
inner join Detalle_Compra dc on dc.id_compra=c.id_Compra
inner join Producto p on p.id_Producto=dc.id_producto
inner join Categoria ca on ca.id_Categoria=p.id_categoria
inner join Marca mc on p.id_marca=mc.id_Marca
where convert(date,c.FechaRegistro) between @fechaCreacion and @fechaFin
and pr.id_Proveedor= if(@idproveedor=0,pr.id_Proveedor,@idproveedor)
end

```

```

---Procedimiento almacenado para generar el reporte de Venta---
create or alter procedure ReporteVentas(
@fechaCreacion varchar(10),
@fechaFin varchar(10),
@idcliente int
)
as
begin
set dateformat dmy;
select
convert(char(10),v.FechaRegistro,103)[FechaCreacion],v.TipoDocumento,v.NumeroDocumen
to,v.MontoTotal,
u.Nombre1[NombreUsuario],
cl.Cedula[Cedula],cl.Nombre1[NombreCliente],cl.Apellido1[ApellidoCliente],
p.Codigo[CodigoProducto],p.Nombre_Producto[NombreProducto],mc.Nombre[Marca],ca.Nombr
e[Categoria],p.FechaVencimiento[FechaVencimiento],dv.PrecioVenta,dv.Cantidad,dv.Sub_
Total[Total]
from Venta v
inner join Usuario u on u.id_usuario=v.id_usuario
inner join Cliente cl on cl.id_Cliente=v.id_cliente
inner join Detalle_Venta dv on dv.id_venta=v.id_venta
inner join Producto p on p.id_Producto=dv.id_producto
inner join Categoria ca on ca.id_Categoria=p.id_categoria
inner join Marca mc on p.id_marca=mc.id_Marca
where convert(date,v.FechaRegistro) between @fechaCreacion and @fechaFin
and cl.id_Cliente= if(@idcliente=0,cl.id_Cliente,@idcliente)
end

```

V. Diseño De La Aplicación

Explicación de las Tecnologías utilizadas

Las tecnologías utilizadas en el desarrollo de esta aplicación son como gestor de base de Datos SQL Server en su versión Managment Studio 2019 y el lenguaje de Programación C# con el editor de Código Visual Studio en su versión 2022, así también utilizando la conexión a la base de datos con la tecnología ADO.net, La tecnología de desarrollo de la aplicación es. NetFramework 4.8.2

La arquitectura de desarrollo es la arquitectura en capas através de bibliotecas de clases de .NetFramework en la cual estarán contenida la Capa Datos, Capa Entidad, Capa Negocio y la Capa Presentación, Utilizando la inserción de Datos a través de procedimientos almacenados y realizando ciertas validaciones a la Base De Datos, también en el apartado de la reporteria utilizando librerías y extensiones como ReportViewer, DocumentFormat, itextSharp, SpreadsheetLigth para reportes de detalles de Factura generando también Pdf y Excel para la reporteria de Inventario, Compras y ventas por fechas, as también utilizando plantillas HTML para la estructura de facturas de detalles

Capturas de Pantalla del Proceso de Venta y Facturación

SISTEMA DE VENTAS

Mantenedor Proveedores Clientes Compras Ventas Reportes Usuarios Mercedes

domingo, 26 de noviembre de 2023 21:14:05

REGISTRAR VENTA

Información Compra

Fecha: 26/11/2023 Documento: Boleta

Información Del Cliente

Cedula: [] Nombre: [] Apellido: []

Información De Producto

Codigo Producto	Producto	Precio	Stock	Cantidad
[]	[]	[]	[]	1

Table:

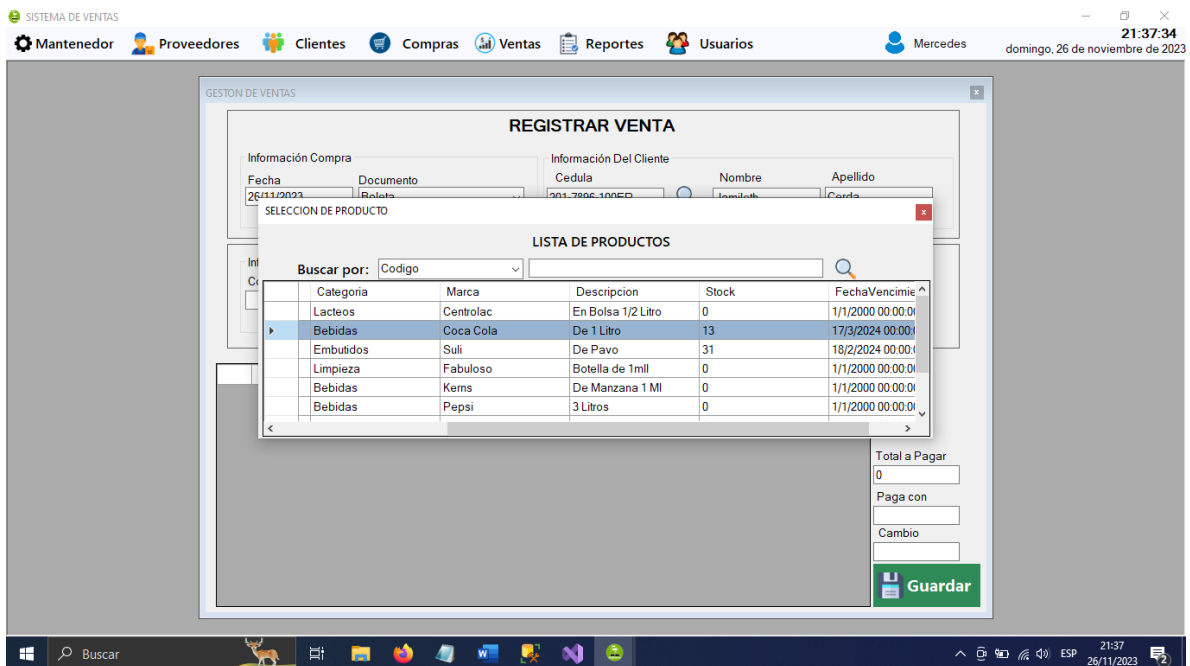
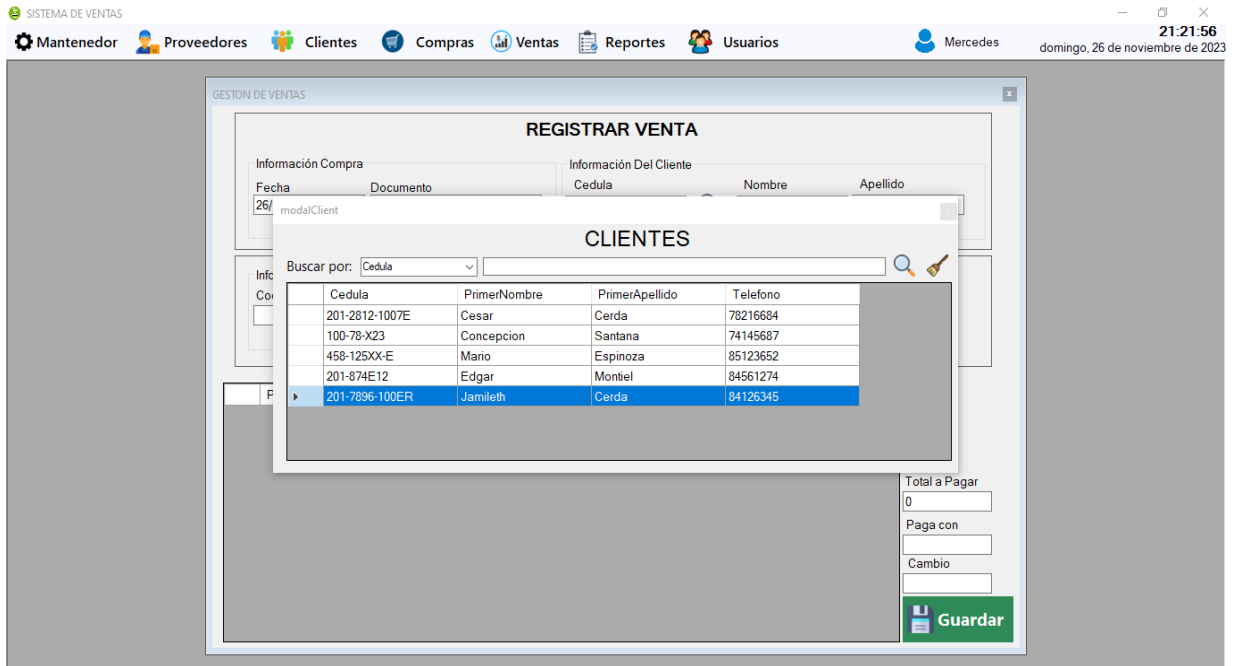
Producto	Precio	Cantidad	SubTotal
----------	--------	----------	----------

Total a Pagar: 0

Paga con: []

Cambio: []

Guardar



GESTION DE VENTAS

REGISTRAR VENTA

Información Compra

Fecha
26/11/2023

Documento
Boleta

Información Del Cliente

Cedula
201-7896-100ER

Nombre
Jamileth

Apellido
Cerde

Información De Producto

Codigo Producto

Producto

Precio

Stock

Cantidad
1

Producto	Precio	Cantidad	SubTotal
Gaseosa	40.00	2	80.00

Total a Pagar
80.00

Paga con

Cambio

Guardar

GESTION DE VENTAS

REGISTRAR VENTA

Información Compra

Fecha
26/11/2023

Documento
Boleta

Información Del Cliente

Cedula
201-7896-100ER

Nombre
Jamileth

Apellido
Cerde

Información De Producto

Codigo Producto

Producto

Precio

Stock

Cantidad
1

Producto	Precio	Cantidad	SubTotal
Gaseosa	40.00	2	80.00

Total a Pagar
80.00

Paga con
100

Cambio
20.00

Guardar

SISTEMA DE VENTAS

Mantenedor Proveedores Clientes Compras Ventas Reportes Usuarios Mercedes

domingo, 26 de noviembre de 2023 21:15:38

REGISTRAR VENTA

Información Compra

Fecha: 26/11/2023 Documento: Boleta

Información Del Cliente

Cedula: 201-7896-100ER Nombre: Jamileth Apellido: Cerda

Información De Producto

Codigo Producto: Producto: Cantidad: 1

Producto: Gaseosa Precio: 40.00 Cantidad: 2

Numero de Venta generado: 0007

¿Desea trasladar al portapapeles?

Sí No

Total a Pagar: 80.00

Paga con: 100

Cambio: 20.00

Guardar

SISTEMA DE VENTAS

Mantenedor Proveedores Clientes Compras Ventas Reportes Usuarios Mercedes

domingo, 26 de noviembre de 2023 21:16:27

DETALLE DE VENTA

Detalle De Venta

Numero de Documento: 0007

Información De Venta

Fecha: 26/11/2023 Tipo De Documento: Boleta Usuario: Mercedes

Información Del Cliente

Numero De Cedula: 201-7896-100ER Nombre Cliente: Jamileth

Producto	Precio	Cantidad	SubTotal
Gaseosa	40.00	2	80.00

Monto Total: 80.00

Monto Pago: 100.00

Monto Cambio: 20.00

Factura

Numero de Documento 0007

Ventas Reportes Usuarios Mercedes 21:40:30 domingo, 26 de noviembre de 2023

SURTIDORA ALEJANDRA
J0310000203288
Oficinas E.Chamorro 3C al Norte

N°Documento 0007
Tipo De Documento Boleta

Datos del Cliente
N°Cedula 201-7896-100ER
Nombre Jamileth

Datos del Facturador
Nombre

Numero de Documento 0007

De Documento Usuario
eta Mercedes

Nombre Cliente
Jamileth

Precio	Cantidad	SubTotal
40.00	2	80.00

Monto Total 80.00
Monto Pago 100.00
Monto Cambio 20.00

Buscar

Factura

Numero de Documento 0007

Ventas Reportes Usuarios Mercedes 21:40:45 domingo, 26 de noviembre de 2023

Datos del Facturador
Nombre

Detalle De la Compra

Nombre Producto	Precio	Cantidad
Gaseosa	40.00	2
TOTAL		80.00
PAGO		100.00
CAMBIO		20.00

¡GRACIAS POR COMPRAR!

Numero de Documento 0007

De Documento Usuario
eta Mercedes

Nombre Cliente
Jamileth

Precio	Cantidad	SubTotal
40.00	2	80.00

Monto Total 80.00
Monto Pago 100.00
Monto Cambio 20.00

Buscar

DETALLE DE VENTA

Numero de Documento 0007

Información de Venta

Fecha 26/11/2023 Tipo De Documento Boleta Usuario Mercedes

Información Del Cliente

Numero De Cedula 201-7896-100ER

Producto

Gaseosa

SubTotal 80.00

Monto Total 80.00 Monto Pago 100.00 Monto Cambio 20.00

Reporte de Factura Generado

Aceptar

Web Page Design - FacturaDeVenta

file:///C:/Users/Cesar Cerda/Desktop/FacturaDeVenta_11erce26e59.pdf

1 de 1

110%

SURTIDORA ALEJANDRA

RUC: J0310000203288
Dirección: Oficinas E. Chamorro 3 Cuadras al Norte

BOLETA

0007

Cedula Cliente: 201-7896-100ER
Nombre Cliente: Jamileth
Fecha Registro: 26/11/2023
Usuario Registro: Mercedes

Producto	Precio	Cantidad	Sub Total
Gaseosa	40.00	2	80.00

Monto Total	Pago Con	Cambio
80.00	100.00	20.00

VI. Conclusiones Y Recomendaciones

En conclusión con respecto a todo lo planteado en el proyecto, se fundamentó y se demostró que el proyecto fue un prototipo de aplicación para la gestión de control de inventario y ventas con facturación para la surtidora "Ale" de productos varios ubicado en el departamento de Granada. Para ello, se desarrolló lo siguiente, Se desarrolló un prototipo funcional de la aplicación, que permitió validar el diseño y la funcionalidad de la misma. Se realizó el modelo conceptual de entidad-relación, que define la estructura de los datos que se almacenarán en la base de datos. Se modeló el diagrama relacional, que representa la estructura física de la base de datos. Se desarrolló el prototipo de aplicación de escritorio, que utiliza las tecnologías C#, Visual Studio 2022 y SQL Server 2019. Se incorporaron funciones de facturación, registro de ventas y compras de los productos. El prototipo desarrollado cumple con los objetivos planteados, y proporciona a la surtidora "Ale" una herramienta para el control de inventario y ventas, que le permitirá mejorar su eficiencia y productividad.

VIII. Anexos

Diccionario de Datos

Tabla Rol

	Columna	Tipo De Dato	Descripcion
PK	id_Rol	int identity(1,1)	Identificador Para Tabla Rol
	Descripcion	nVarchar(50)	Descripcion O Nombre del Rol
	FechaRegistro	dateTime	Fecha de Registro del Rol

Tabla Permiso

	Columna	Tipo De Dato	Descripcion
PK	id_Permiso	int identity(1,1)	Identificador Para Tabla Permiso
FK	id_rol	int	Referencia a la Tabla Rol
	NombreMenu	nvarchar(70)	Indica el nombre del menu para acceso
	FechaRegistro	dateTime	Fecha de Registro del permiso

Tabla Proveedor

	Columna	Tipo De Dato	Descripcion
PK	id_Proveedor	int identity(1,1)	Identificador para Tabla Proveedor
	Documento	nvarchar(50)	Documeto del Proveedor
	Razon Social	nvarchar(50)	RazonSocial para auditoria
	Nombre	nvarchar(50)	Nombre del Proveedor
	Correo	nvarchar(50)	Correo Para Comunicarse Con el Proveedor
	Telefono	nvarchar(8)	Telefono del proveedor
	Estado	bit	indica si el proveedor esta activo o no
	FechaRegistro	dateTime	Fecha de Registro del Proveedor

Tabla Cliente

	Columna	Tipo De Dato	Descripcion
PK	id_Cliente	int identity(1,1)	Identificador para Tabla Cliente
	Cedula	nvarchar(15)	Indica la cedula o identificador del cliente
	Nombre1	nvarchar(10)	Primer Nombre del Cliente
	Nombre2	nvarchar(10)	Segundo Nombre del Cliente
	Apellido1	nvarchar(10)	Primer Apellido del Cliente
	Apellido2	nvarchar(10)	Segundo Apellido del Cliente
	Telefono	nvarchar(8)	Indica el telefono para contactar al cliente
	Estado	bit	Indica si el cliente esta activo o no
	FechaRegistro	dateTime	Fecha De Registro del Cliente

Tabla Usuario

	Columna	Tipo De Dato	Descripcion
PK	id_Usuario	int identity(1,1)	identificador para la tabla Usuario
	Nombre1	nvarchar(10)	Primer Nombre del Usuario
	Nombre2	nvarchar(10)	Segundo Nombre del Usuario
	Apellido1	nvarchar(10)	Primer Apellido del Usuario
	Apellido2	nvarchar(10)	Segundo Apellido del Usuario
	usuario	nvarchar(10)	nombre de Usuario para acceso
	Correo	nvarchar(10)	indica el correo del usuario
FK	Clave	nvarchar(10)	indica la clave de usuario para acceso
	id_rol	int	Referencia a la tabla rol
	Estado	bit	indica si el usuario esta o no activo
	FechaRegistro	dateTime	Fecha de Registro del Usuario

Tabla Categoria

	Columna	Tipo De Dato	Descripcion
PK	id_Categoria	int identity(1,1)	Identificador de Tabla Categoria
	Nombre	nvarchar(20)	Indica el Nombre de la Categoria
	Estado	bit	indica si la categoria esta activa o no
	FechaRegistro	dateTime	Indica la fecha de Registro de la Categoria

Tabla Marca

	Columna	Tipo De Dato	Descripcion
PK	id_Marca	int identity(1,1)	Identificador de la Tabla Marca
	Nombre	nvarchar(20)	indica el nombre de la marca
FK	id_categoria	int	Referencia a la tabla Categoria
	Estado	bit	indica si la Marca esta activa o no
	FechaRegistro	dateTime	Indica la fecha de Registro de la Marca

Tabla Detalle_Venta

	Columna	Tipo de Dato	Descripcion
PK	id_DetalleVenta	int identity(1,1)	identificador de la tabla DetalleVenta
FK	id_venta	int	hace referencia a la tabla venta
FK	id_Producto	int	hace referencia a la tabla producto
	PrecioVenta	decimal(18,2)	indica el precio de venta del producto
	Cantidad	int not null	indica la cantidad vendida del producto
	Sub_Total	decimal(18,2)	indica el sub total de la venta
	FechaRegistro	dateTime	indica la fecha de registro del detalle de la venta

Tabla Venta

	Columna	Tipo de Dato	Descripcion
PK	id_Venta	int identity(1,1)	identificador de la tabla venta
FK	id_usuario	int	hace referencia a la tabla usuario
	TipoDocumento	nvarchar(20)	indica el tipo de documento para la venta
	NumeroDocumento	nvarchar(20)	Indica el numero de documento de la venta
FK	id_Cliente	int	hace referencia a la tabla cliente
	MontoPago	decimal(18,2)	Indica el monto de pago de la venta
	MontoCambio	decimal(18,2)	Indica el monto de cambio de la venta
	MontoTotal	decimal(18,2)	Indica el monto de total de la venta
	FecahRegistro	dateTime	Indica la fecha de registro de la venta

Tabla Producto

	Columna	Tipo De Dato	Descripcion
PK	id_Producto	int identity(1,1)	Identificador de la Tabla Producto
	Codigo	nvarchar(20)	indica el codigo unico de cada producto
	Nombre_Producto	nvarchar(20)	Indica el nombre del Producto
	Descripcion	nvarchar(20)	descripcion para producto
	PrecioCompra	decimal(18,2)	indica el precio de compra del producto
	PrecioVenta	decimal(18,2)	indica el precio de venta del producto
FK	id_categoria	int	referencia a la tabla Categoria
	Stock	int not null	hace referencia a la cantidad que hay en inventario
	Estado	bit	Indica si el producto esta activo o no
	FechaRegistro	dateTime	FechaDeRegistroDelProducto
	FechaVencimiento	date	Hace Referencia a la fecha de vencimiento del producto

Tabla Compra

	Columna	Tipo De Dato	Descripcion
PK	id_Compra	int identity(1,1)	identificador de la tabla Compra
FK	id_usuario	int	referencia a la tbala Usuario
FK	id_proveedor	int	referncia a la tbala proveedor
	TipoDocumento	nvarchar(20)	indica si el dcoumento es factura o boleta
	NumeroDocumento	nvarchar(20)	indica el numero del documento
	Sub_Total	decimal(18,2)	indica el total de la compra de un producto
	FechaRegistro	dateTime	indica la fecha de registro de la venta

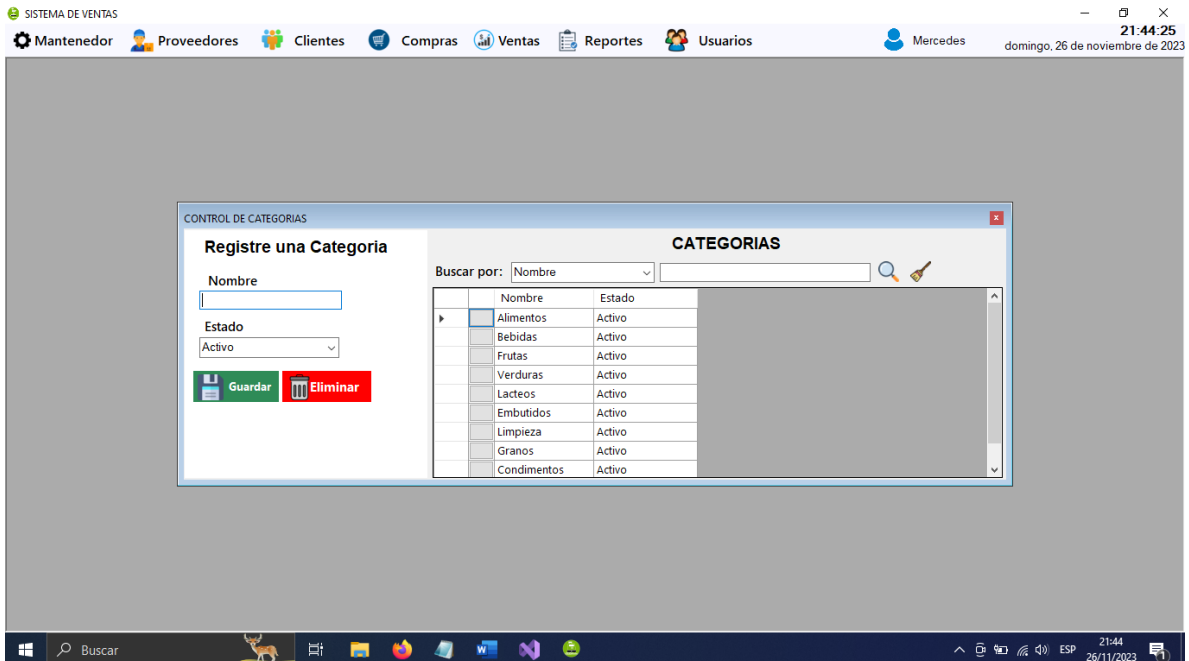
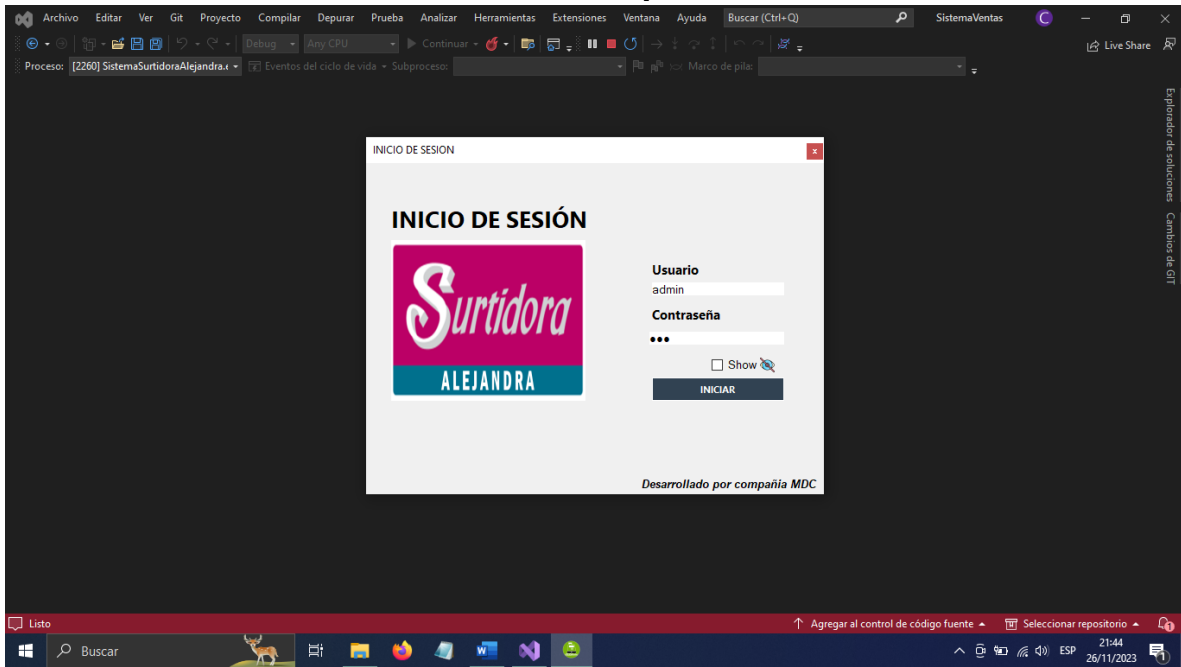
Tabla Detalle_Compra

	Columna	Tipo De Dato	Descripcion
PK	id_DetalleCompra	int identity(1,1)	identificador de la tabla DetalleCompra
FK	id_Compra	int	referencia a la Tabla Compras
FK	id_producto	int	referencia a la tabla producto
	PrecioCompra	decimal(18,2)	Precio de compra del producto
	PrecioVenta	decimal(18,2)	precio de venta del producto
	Cantidad	int	Cantidad de producto comprado
	Total	decimal(18,2)	Indica el total de la Compra de un producto
	FechaRegistro	dateTime	indica la fecha de registro de la compra
	FechaVencimiento	date	indica la fecha de vencimiento del producto

Tabla Negocio

	Columna	Tipo de Dato	Descripcion
PK	id_Negocio	int primary key	identificador de la tabla negocio
	Nombre	nvarchar(80)	Almacena el nombre del Negocio
	RUC	nvarchar(80)	almacena el numero RUC del Negocio
	Direccion	nvarchar(80)	almacena la direccion del Negocio
	Logo	varbinay(max)	almacena la imagen o logo del negocio

Resto de Capturas



CONTROL DE MARCAS

Registre una Marca

Nombre

Categoría

Estado

MARCAS

Buscar por:

	Marca	Categoría	Estado
▶	La Perfecta	Lacteos	Activo
	Coca Cola	Bebidas	Activo
	Pepsi	Bebidas	Activo
	Eskimo	Lacteos	Activo
	Centrolac	Lacteos	Activo
	Parmalat	Lacteos	Activo
	Kerns	Bebidas	Activo
	Suli	Embutidos	Activo
	Kuaker	Granos	Activo
	Fabuloso	Limpieza	Activo
	Yumiey	Golosinas	Activo

CONTROL DE PRODUCTOS

INGRESO DE PRODUCTOS

Producto

Categoría

Marca

Código

Descripción

Estado

PRODUCTOS

Buscar por:

	Codigo	Nombre	Categoría	Marca	Descripción	Stock
▶	LCH-120	Leche	Lacteos	Centrolac	En Bolsa 1/2 Litro	0
	GBCC-3075	Gaseosa	Bebidas	Coca Cola	De 1 Litro	13
	JES-1780	Jamon	Embutidos	Suli	De Pavo	31
	ALF-2114	Asistin	Limpieza	Fabuloso	Botella de 1ml	0
	JBK-922	Jugo	Bebidas	Kerns	De Manzana 1 Mi	0
	GBP-79	Gaseosa	Bebidas	Pepsi	3 Litros	0
	JBDV-3483	Jugo	Bebidas	Del Valle	2 Litros	0
	AGK-767	Avena	Granos	Kuaker	Sin Granular	0
	CLLP-2999	Crema	Lacteos	La Perfecta	2mLi En Bolsa	0


SISTEMA DE VENTAS

Mantenedor Proveedores Clientes Compras Ventas Reportes Usuarios Mercedes

21:45:07
domingo, 26 de noviembre de 2023

DATOS SURTIDORA

LOGO



Subir

Nombre del Negocio

Surtidora Alejandra


RUC

J0310000203288

Dirección

Oficinas E. Chamorro 3 Cuadras

Guardar Cambios



Buscar

SISTEMA DE VENTAS

Mantenedor Proveedores Clientes Compras Ventas Reportes Usuarios Mercedes

21:45:19
domingo, 26 de noviembre de 2023

CONTROL DE PROVEEDORES

Registrar Proveedor

N°Documento

Razón Social

Nombre

Correo

Teléfono

Estado

Activo

Guardar Eliminar

PROVEEDORES

Buscar por: N°Documento

	N°Documento	Razon Social	Nombre	Correo	Telefono
	J5080F45	Proveedor de Bebidas	Coca Cola	cocacola@gmail...	2552-9635
	J6764D34	Proveedor de Dulces	Bocas Diana	bcd@gmail.com	2552-8712
	J123G123	Proveedor de Carnes y ...	SuKarne	sukar@gmail.com	2552-7841
	J56789GA34	Proveedor de Alimentos	Agricorp	agrip@gmail.com	2552-1445

Buscar

SISTEMA DE VENTAS

Mantenedor

Proveedores

Cientes

Compras

Ventas

Reportes

Usuarios

Mercedes

21:45:40

domingo, 26 de noviembre de 2023

CONTROL DE CLIENTES

Registrar Cliente

N°Cedula

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Teléfono

Estado

Activo

Guardar

Eliminar

CLIENTES

Buscar por: N°Cedula

	N°Cedula	PrimerNombre	SegundoNombre	PrimerApellido	SegundoApellido	Te
▶	201-2812-100...	Cesar	Jose	Cerda	Maldonado	782
	100-78-X23	Concepcion	Del Socorro	Santana	Abea	741
	458-125XX-E	Mario	Jose	Espinoza	Lugo	851
	201-874E12	Edgar	Manuel	Montiel	Meneses	845
	201-7896-100...	Jamileth	Del Socorro	Cerda	Abea	841

SISTEMA DE VENTAS

Mantenedor

Proveedores

Cientes

Compras

Ventas

Reportes

Usuarios

Mercedes

21:45:55

domingo, 26 de noviembre de 2023

GESTION DE COMPRA

REGISTRAR COMPRA

Información Compra

Fecha

26/11/2023

Documento

Boleta

Información Del Proveedor

N° Documento

Nombre

Información De Producto

Codigo Producto	Producto	Precio Compra	Precio Venta	Cantidad	Fecha Vencimiento
				1	2023/11/26

Producto	PrecioCompra	Cantidad	SubTotal
----------	--------------	----------	----------

Total a Pagar

0

Guardar

REPORTE DE COMPRAS

Reporte De Compras

Fecha Inicio: 17/11/2023 Fecha Fin: 29/11/2023 Proveedor: Proveedor de Carnes y Embutido

Buscar Por: FechaCreacion

FechaCreacion	TipoDocumento	NumeroDocumento	Sub_Total	NombreUsuario	DocumentoProveedor	RazonSocial
17/11/2023	Boleta	0003	3000.00	Mercedes	J123G123	Proveedor de C.
17/11/2023	Boleta	0005	350.00	Mercedes	J123G123	Proveedor de C.

REPORTE DE VENTAS

Reporte De Ventas

Fecha Inicio: 26/11/2023 Fecha Fin: 29/11/2023 DATOS DEL CLIENTE: Nombre: Jamileth Apellido: Cerda

Buscar Por: FechaCreacion

Generar

FechaCreacion	TipoDocumento	NumeroDocumento	MontoTotal	NombreUsuario	Cedula
26/11/2023	Boleta	0007	80.00	Mercedes	201-7896-100ER

SISTEMA DE VENTAS

Mantenedor

Proveedores

Clientes

Compras

Ventas

Reportes

Usuarios

Mercedes

domingo, 26 de noviembre de 2023

21:47:48

CONTROL DE USUARIOS

Registre un Usuario

Primer Nombre

Segundo Nombre

Primer Apellido

Segundo Apellido

Usuario

Correo

Contraseña

Confirmar Contraseña

Rol

Estado

Guardar

Eliminar

USUARIOS

Buscar por:

PrimerNombre

	PrimerNombre	SegundoNombre	PrimerApellido	SegundoApellido	Usuario	Correo
	Mercedes	Del Socorro	Santana	Aleman	admin	mer@gam
	Maria	Alejandra	Ruiz	Santana	emp	aleruiz@g
	Francisco	Jose	Martinez	Lopez	bodega	franmar@c

Buscar

21:47

26/11/2023

```

    erDiagram
        Permiso ||--o{ Usuario : "tiene"
        Rol ||--o{ Usuario : "tiene"
        Usuario ||--o{ Compra : "realiza"
        Usuario ||--o{ Venta : "realiza"
        Cliente ||--o{ Venta : "realiza"
        Proveedor ||--o{ Compra : "provee"
        Producto ||--o{ Compra : "provee"
        Producto ||--o{ Venta : "provee"
        Categoria ||--o{ Producto : "clasifica"
        Marca ||--o{ Categoria : "clasifica"
        Detalle_Compra ||--o{ Compra : "detalle"
        Detalle_Venta ||--o{ Venta : "detalle"
  
```

Permiso

- Id_Permiso
- Id_Rol
- NombreMenu
- FechaRegistro

Rol

- Id_Rol
- Descripcion
- FechaRegistro

Usuario

- Id_Usuario
- Nombre1
- Nombre2
- Apellido1
- Apellido2
- usuario
- Correo
- Clave
- Id_Rol
- Estado
- FechaRegistro

Cliente

- Id_Cliente
- Cedula
- Nombre1
- Nombre2
- Apellido1
- Apellido2
- Telefono
- Estado
- FechaRegistro

Compra

- Id_Compra
- Id_usuario
- Id_proveedor
- TipoDocumento
- NumeroDocumento
- Sub_Total
- FechaRegistro

Venta

- Id_Venta
- Id_usuario
- TipoDocumento
- NumeroDocumento
- Id_cliente
- MontoPago
- MontoCambio
- MontoTotal
- FechaRegistro

Proveedor

- Id_Proveedor
- Documento
- RazonSocial
- Nombre
- Correo
- Telefono
- Estado
- FechaRegistro

Producto

- Id_Producto
- Codigo
- Nombre_Producto
- Descripcion
- PrecioCompra
- PrecioVenta
- Id_categoria

Categoria

- Id_Categoria
- Nombre
- Estado
- FechaRegistro

Marca

- Id_Marca
- Nombre
- Id_categoria
- Estado
- FechaRegistro

Detalle_Compra

- Id_DetalleCompra
- Id_compra
- Id_producto
- PrecioCompra
- PrecioVenta
- FechaVendimiento
- Cantidad
- Total
- FechaRegistro

Detalle_Venta

- Id_DetalleVenta
- Id_venta
- Id_producto
- PrecioVenta
- Cantidad
- Sub_Total
- FechaRegistro