

***Universidad Nacional Autónoma de Honduras***



***Informe de Proyecto Final: Sistema de Ventas y  
Facturación - Mercadito Darwin***

***Asignatura: Bases de Datos I***

***Catedrático: Ingeniero Constantino Sorto***

***Grupo#2***

***Duvan Lemus — 20161000875***  
***Junior Zambrano — 20161003533***  
***Yonny Lopez — 20141012365***

***I PAC 2022***

## Índice

Índice .....	2
Introducción .....	3
Objetivos .....	4
Objetivo General .....	4
Objetivos Específicos .....	4
Enunciado de Problema a Resolver .....	5
Formato Real de una Factura Válida en la SAR .....	7
Script de la Base de Datos .....	9
Capturas de Pantalla del Sistema de Ventas y Facturación .....	19

## **Introducción**

Los procesos se industrializan cada vez más. Uno de los aspectos clave de la contemporaneidad en que vivimos es la constante y creciente demanda de software que ayuda a preservar información y, sobre todo, agilizar el proceso de generación de dinero para las empresas, desde negocios emprendedores personales hasta grandes multinacionales que requieren el uso de complejas piezas informáticas para procesar datos.

Es en ese momento cuando las bases de datos y su posterior poder de análisis de datos entra en juego. Los procesos de negocios incluyen el procesamiento de datos para distintos objetivos, sin embargo, se puede considerar que la toma de decisiones dentro de la empresa y la proyección de la rentabilidad a largo plazo son los factores esenciales para el uso de software de análisis y almacenamiento de datos.

Para llevar a cabo un ejemplo práctico y funcional, se modela un escenario en el cual una empresa que vende productos de abarrotería en general quiere digitalizar el proceso de venta, compra, distribución y manejo de inventarios, así como asignar roles a los distintos colaboradores que laboran, tomando en cuenta el dinamismo de la empresa con una proyección amplia para manejar más de una sucursal.

## **Objetivos**

### ***Objetivo General***

Explorar la funcionalidad de las bases de datos en el mundo empresarial a través del desarrollo de un sistema de ventas que cumpla con los requisitos para su funcionalidad en un establecimiento comercial de tipo “mercadito”.

### ***Objetivos Específicos***

1. Desarrollar una base de datos relacional (Microsoft SQL Server) con ayuda de los conocimientos adquiridos durante el desarrollo de la asignatura, que permita visualizar la correcta relación entre los componentes de una empresa.
2. Crear una interfaz gráfica con ayuda de un lenguaje de programación (C#) para controlar las funciones de la base datos, aplicando los conocimientos y funciones de DML, DDL, TCL y DCL.
3. Comprender el funcionamiento tributario nacional con la aplicación de la normativa autorizada por el Servicio de Administración de Rentas (SAR), tomando en consideración todos los elementos que debe contener un recibo de compra, boleta o factura en Honduras.

## **Enunciado de Problema a Resolver**

Se ha solicitado la modelar la realidad relativa para una empresa enfocada en el rubro de venta de productos de primera mano, un Mercadito, el cual ha tenido un crecimiento espontaneo en el año 2021 y ha decidido abrir una sucursal mas, el dueño ha tomo la decisión que si tiene éxito proseguiría a su expansión con una nueva sucursal, por lo cual se tiene que tomar en cuenta al momento de diseñar la base de datos.

El Mercadito con el nombre de Mercadito Darwin necesita que disponga de las siguientes áreas, las cuales son obligatorias, estas son especificadas detalladamente con lo que se requiere conocer de cada una de ellas.

Para el área de Recursos Humanos se desea conocer para las personas los datos personales, siendo identificados inequívocamente por su numero de cédula, nombre completo obligatorio, con la excepción del segundo apellido, el cual puede no tener, genero, estado civil, ciudad, ciudad. Para los datos de empleado, Email, teléfonos, cargo, área de trabajo, sucursal a la que pertenece, dirección detallada, tipo de contrato, salario, bonos, beneficios, fecha de contratación, fecha de despido o renuncia, indicando una descripción y si tiene prestaciones por su servicio.

Productos manejo de inventario, para esta área se requiere saber del producto, el id del producto, código de Barra, nombre, descripción, precio, fecha de caducidad, unidad de medida, la categoría a la que pertenece, y el Lote, el cual contiene en el, la fecha de fabricación del producto, ubicación, y quien lo hizo. Se debe mostrar en el inventario cantidad de productos actual (stock) , la cual se obtendrá de la existencia de mes, mas las entradas menos las salidas de producto, y mostrar el producto que ya caduco.

En el caso de la Facturación, se pide para los clientes tener registrado el id, nombre y apellido, primer nombre y apellido obligatorio, RTN, ciudad, fecha de nacimiento, teléfonos, correos Email, categoria. Mostrar en la factura, cliente, los productos que lleva, nombre, precio, cantidad, total parcial. Calculandose al final de la factura valor antes de impuestos, descuentos, y el total de la compra.

Los clientes pueden pagar con efectivo o Tarjeta de crédito/débito. El cliente puede decidir si quiere una sola factura o varias de la compra que realizó.

Se desea que haya una modulación en lo que los usuarios puedan hacer en la base de datos al autenticarse, de manera que haya perfiles creados, los cuales tendrán permisos diferentes mayores o menores para el uso del sistema que será implementado en el Mercadito. De los usuarios, se requiere, el id, un nombre, un tipo de usuario y una contraseña.

En cuanto a cada sucursal se solicita el número de sucursal, ciudad, y dirección en la que está ubicada.

LOGO

RAZON SOCIAL DEL CLIENTE

R.T.N.: 01010101010101

DIRECCION, DIRECCION, DIRECCION,  
DIRECCION, DIRECCION, DIRECCION, DIRECCION,  
TEL.: (504) 0000-1111

Correo electrónico: XXXXXXXXXX@XXXXX.com

C.A.I.: 000000-000000-000000-000000-000000-00

RANGO AUTORIZADO DEL 000-000-00-00000000 AL 000-000-00-00000000

Fecha Límite de Emisión: XX/XX/XXXX

2

FACTURA

No. 000-001-01-0XXXXXXX

1

CLIENTE:

R.T.N.:

CANTIDAD	DESCRIPCION	PRECIO UNITARIO	DESCUENTOS Y REBAJAS OTORGADOS	TOTAL
			5	
		TOTAL		

VALOR EN LETRAS:

LA FACTURA ES BENEFICIO DE TODOS "EXIJALA"

6

N° Correlativo de orden de compra exenta

N° Correlativo de constancia de registro exonerado

N° Identificativo del registro de la SAG

7

IMPORTE EXONERADO L.

IMPORTE EXENTO L.

IMPORTE GRAVADO 15% L.

IMPORTE GRAVADO 18% L.

I.S.V. 15% L.

I.S.V. 18% L.

TOTAL A PAGAR L.

8

ORIGINAL: CLIENTE COPIA: EMISOR

1

CLIENTE:

R.T.N.:

CANTIDAD	DESCRIPCION	PRECIO UNITARIO	DESCUENTOS Y REBAJAS OTORGADOS	TOTAL
			5	
		TOTAL		

VALOR EN LETRAS:

LA FACTURA ES BENEFICIO DE TODOS "EXIJALA"

6

N° Correlativo de orden de compra exenta

N° Correlativo de constancia de registro exonerado

N° Identificativo del registro de la SAG

7

IMPORTE EXONERADO L.

IMPORTE EXENTO L.

IMPORTE GRAVADO 15% L.

IMPORTE GRAVADO 18% L.

I.S.V. 15% L.

I.S.V. 18% L.

TOTAL A PAGAR L.

8

ORIGINAL: CLIENTE COPIA: EMISOR





## Script de la Base de Datos

```
CREATE DATABASE Proyecto_Mercadito;  
USE Proyecto_Mercadito;
```

```
----- Tablas de Sucursal -----  
--Notación Snake Case  
--1  
CREATE TABLE Departamento(id_departamento INTEGER not null IDENTITY(1,1),  
nombre varchar(100),  
CONSTRAINT Departamento_PK PRIMARY  
KEY( id_departamento));  
GO  
CREATE TABLE Ciudad(id_ciudad INTEGER not null IDENTITY(1,1), departamento_id  
INTEGER not null, nombre varchar(100),  
CONSTRAINT Ciudad_PK PRIMARY KEY(id_ciudad),  
CONSTRAINT Departamento_FK FOREIGN  
KEY(departamento_id) REFERENCES Departamento(id_departamento) ON UPDATE  
CASCADE);  
GO  
  
CREATE TABLE Colonia(id_colonia INTEGER not null IDENTITY(1,1), ciudad_id  
INTEGER not null, nombre varchar(100),  
CONSTRAINT Colonia_PK PRIMARY KEY(id_colonia),  
CONSTRAINT Ciudad_FK FOREIGN KEY(ciudad_id)  
REFERENCES ciudad(id_ciudad) ON UPDATE CASCADE);  
GO  
CREATE TABLE Sucursal(id_sucursal INTEGER not null IDENTITY(1,1), colonia_id  
INTEGER not null, rtn varchar(14), nombre_sucursal varchar(100),  
nombre_empresa varchar(100), direccion_detalle  
varchar(200), no_edificio varchar(100),  
CONSTRAINT Sucursal_PK PRIMARY KEY(id_sucursal),  
FOREIGN KEY(colonia_id) REFERENCES  
colonia(id_colonia) ON UPDATE CASCADE);  
GO  
CREATE TABLE Telefono(id_telefono INTEGER not null IDENTITY(1,1),  
numero_telefono varchar(100)  
CONSTRAINT Telefono_PK PRIMARY KEY(id_telefono));  
GO  
CREATE TABLE Email(id_email INTEGER not null IDENTITY(1,1), direccion_email  
varchar(100),  
CONSTRAINT Email_PK PRIMARY KEY(id_email));  
GO  
  
--7  
CREATE TABLE Sucursal_X_Telefono(sucursal_id INTEGER not null, telefono_id  
INTEGER not null,  
CONSTRAINT Sucursal_FK FOREIGN KEY(sucursal_id)  
REFERENCES Sucursal(id_sucursal) ON UPDATE CASCADE,  
CONSTRAINT Telefono_FK FOREIGN KEY(telefono_id)  
REFERENCES Telefono(id_telefono) ON UPDATE CASCADE);  
GO  
CREATE TABLE Sucursal_X_Email( sucursal_id INTEGER not null, email_id INTEGER  
not null,  
FOREIGN KEY(sucursal_id) REFERENCES  
Sucursal(id_sucursal) ON UPDATE CASCADE,  
CONSTRAINT Email_FK FOREIGN KEY(email_id) REFERENCES  
Email(id_email) ON UPDATE CASCADE);  
GO
```

-----Tablas de Recursos Humanos -----

```
CREATE TABLE Estado_Civil(id_estado_civil INTEGER not null IDENTITY(1,1),
nombre_estado varchar(100),
CONSTRAINT Estado_Civil_PK PRIMARY
KEY(id_estado_civil));
GO
CREATE TABLE Persona(id_persona varchar(13) not null, estado_civil_Id INTEGER
not null, ciudad_id INTEGER, p_nombre varchar(100), s_nombre varchar(100),
p_apellido varchar(100), s_apellido varchar(100), genero
varchar(100), fecha_nacimiento DATE,
CONSTRAINT Persona_PK PRIMARY KEY(id_persona),
CONSTRAINT Estado_Civil_FK FOREIGN KEY(estado_civil_Id)
REFERENCES Estado_Civil(id_estado_civil) ON UPDATE CASCADE,
FOREIGN KEY(ciudad_id) REFERENCES Ciudad(id_ciudad) ON
UPDATE CASCADE);
GO
```

-----CAMPO CALCULADO-----

--11

```
CREATE TABLE Persona_X_Telefono(persona_id varchar(13) not null, telefono_id
INTEGER not null,
CONSTRAINT Persona_FK FOREIGN KEY(persona_id)
REFERENCES Persona(id_persona) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY(telefono_id) REFERENCES
Telefono(id_telefono) ON DELETE CASCADE ON UPDATE CASCADE);
GO
CREATE TABLE Persona_X_Email(persona_id varchar(13) not null, email_id INTEGER
not null,
FOREIGN KEY(persona_id) REFERENCES
Persona(id_persona) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY(email_id) REFERENCES Email(id_email) ON
UPDATE CASCADE);
GO
```

```
CREATE TABLE Cargo(id_cargo INTEGER not null IDENTITY(1,1), nombre_cargo
varchar(100),
CONSTRAINT Cargo_PK PRIMARY KEY(id_cargo));
GO
CREATE TABLE Empleado(id_empleado INTEGER not null IDENTITY(1,1), persona_id
varchar(13) not null, cargo_id INTEGER not null, sucursal_id INTEGER not null,
direccion_detallada varchar(200),
CONSTRAINT Empleado_PK PRIMARY KEY(id_empleado),
FOREIGN KEY(persona_id) REFERENCES
Persona(id_persona) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT Cargo_FK FOREIGN KEY(cargo_id) REFERENCES
Cargo(id_cargo) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY(sucursal_id) REFERENCES
Sucursal(id_sucursal));
GO
--15
```

```
CREATE TABLE Estado(id_estado INTEGER not null IDENTITY(1,1), empleado_id
INTEGER not null, estado_descripcion varchar(100), fecha_cambio_estado
SMALLDATETIME,
CONSTRAINT Estado_PK PRIMARY KEY(id_estado),
CONSTRAINT Empleado_FK FOREIGN KEY(empleado_id)
```

```

REFERENCES Empleado(id_empleado) ON DELETE CASCADE ON UPDATE CASCADE);
GO
CREATE TABLE Cliente(id_cliente INTEGER not null IDENTITY(1,1), persona_id
varchar(13) not null, rtn varchar(14),
CONSTRAINT Cliente_PK PRIMARY KEY(id_cliente),
FOREIGN KEY(persona_id) REFERENCES
Persona(id_persona) ON DELETE CASCADE ON UPDATE CASCADE);
GO
CREATE TABLE Contrato(id_contrato INTEGER not null IDENTITY(1,1), empleado_id
INTEGER not null UNIQUE, descripcion_contrato varchar(500) , salario_dia
DECIMAL(8,2), fecha_inicio_contrato DATE, fecha_final_contrato DATE,
CONSTRAINT Contrato_PK PRIMARY KEY(id_contrato),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado) );
GO

-----CAMPO CALCULADO PARA ESTADO DE CONTRATO-----

ALTER TABLE Estado ADD estado_contrato as
CASE
WHEN Contrato.fecha_final_contrato <= GETDATE()
THEN 'Contrato Caducado'
ELSE 'Contrato Vigente'
END;
GO

-----

CREATE TABLE Beneficio(id_beneficio INTEGER not null IDENTITY(1,1), contrato_id
INTEGER not null , nombre_beneficio varchar(100), costo_beneficio DECIMAL(8,2),
CONSTRAINT Beneficio_PK PRIMARY KEY(id_beneficio),
CONSTRAINT Contrato_FK FOREIGN KEY(contrato_id)
REFERENCES Contrato(id_contrato));
GO
CREATE TABLE Contrato_X_Beneficio( contrato_id INTEGER not null , beneficio_id
INTEGER not null,
CONSTRAINT Beneficio_FK FOREIGN KEY(beneficio_id)
REFERENCES Beneficio(id_beneficio),
FOREIGN KEY(contrato_id) REFERENCES
Contrato(id_contrato));
GO
--20
CREATE TABLE Detalle_Pago(id_detalle_pago INTEGER not null IDENTITY(1,1),
empleado_id INTEGER not null, dias_laburados INTEGER not null,
dias_no_laburados INTEGER not null,
dias_descanso INTEGER not null, total_beneficios
INTEGER not null, costo_total_beneficio DECIMAL(8,2), salario_dia DECIMAL(8,2)
CHECK(salario_dia >0), devengado DECIMAL(8,2) CHECK(devengado >0),
deducciones DECIMAL(8,2), total DECIMAL(8,2)
CHECK(total>0),
CONSTRAINT Detalle_Pago_PK PRIMARY
KEY(id_detalle_pago),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado));
GO

CREATE TABLE Bono(id_bono INTEGER not null IDENTITY(1,1), detalle_pago_id
INTEGER not null, nombre_bono varchar(100), cantidad_dinero DECIMAL(8,2)
CHECK(cantidad_dinero>0), descripcion varchar(200),
CONSTRAINT Bono_PK PRIMARY KEY(id_bono),
CONSTRAINT Detalle_Pago_FK FOREIGN
KEY(detalle_pago_id) REFERENCES Detalle_Pago(id_detalle_pago));
GO

```

```

CREATE TABLE Recibo_Pago(id_voucher INTEGER not null IDENTITY(1,1),
detalle_pago_id INTEGER not null, cargo varchar(100), cuenta varchar(100),
fecha_recibo SMALLDATETIME,
total_descripcion varchar(100), concepto
varchar(100), total_bruto DECIMAL(8,2) CHECK(total_bruto>0), total_neto
DECIMAL(8,2) CHECK(total_neto>0),
CONSTRAINT Recibo_Pago_PK PRIMARY KEY(id_voucher),
FOREIGN KEY(detalle_pago_id) REFERENCES
Detalle_Pago(id_detalle_pago));
GO
CREATE TABLE Tipo_Baja(id_tipo_baja INTEGER not null IDENTITY(1,1), nombre_baja
varchar(100),
CONSTRAINT Tipo_Baja_PK PRIMARY KEY(id_tipo_baja));
GO
CREATE TABLE Dar_Baja(id_Baja INTEGER not null IDENTITY(1,1), empleado_Id
INTEGER not null, descripcion varchar(100), tipo_baja_id INTEGER not null,
fecha_baja DATE,
CONSTRAINT Dar_Baja_PK PRIMARY
KEY(id_Baja),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado),
CONSTRAINT Tipo_Baja_FK FOREIGN
KEY(tipo_baja_id) REFERENCES Tipo_Baja(id_tipo_baja));
GO
CREATE TABLE Rol(id_rol INTEGER not null IDENTITY(1,1), nombre_rol varchar(200),
fecha SMALLDATETIME,
CONSTRAINT Rol_PK PRIMARY KEY(id_rol));
GO
--26
CREATE TABLE Usuario(id_usuario INTEGER not null IDENTITY(1,1), rol_id INTEGER,
empleado_id INTEGER not null, nombre_usuario varchar(100), clave varchar(100),
estado_servicio varchar(100), fecha_registro SMALLDATETIME,
CONSTRAINT Usuario_PK PRIMARY KEY(id_usuario),
CONSTRAINT Rol_FK FOREIGN KEY(rol_id) REFERENCES
Rol(id_rol) ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado) ON UPDATE CASCADE);
GO
CREATE TABLE Permisos(id_permiso INTEGER not null IDENTITY(1,1), rol_Id INTEGER
not null, nombre varchar(100), fecha SMALLDATETIME,
CONSTRAINT Permisos_PK PRIMARY KEY(id_permiso),
FOREIGN KEY(rol_id) REFERENCES Rol(id_rol) ON DELETE
CASCADE ON UPDATE CASCADE);
GO

-----Tablas de
Inventario -----
CREATE TABLE Marca(id_marca INTEGER not null IDENTITY(1,1), nombre_marca
varchar(100),
CONSTRAINT Marca_PK PRIMARY KEY(id_marca));
GO
CREATE TABLE Categoria_Producto(id_categoria INTEGER not null IDENTITY(1,1),
nombre_categoria varchar(100),
CONSTRAINT Categoria_Producto_PK PRIMARY
KEY(id_categoria));
GO
CREATE TABLE Unidad_Medida(id_unidad_medida INTEGER not null IDENTITY(1,1),
nombre_unidad varchar(100), magnitud varchar(100), abreviatura varchar(5)
CONSTRAINT Unidad_Medida_PK PRIMARY
KEY(id_unidad_medida));

```

```

GO
CREATE TABLE Lote(id_lote varchar(100) not null, ubicacion varchar(100),
fabricante varchar(100), fecha_fabricacion DATE,
CONSTRAINT Lote_PK PRIMARY KEY(id_lote));

GO
--32
CREATE TABLE Producto(id_producto varchar(100) not null, nombre varchar(100),
marca_id INTEGER not null, unidad_medida_id INTEGER not null,
lote_id varchar(100) not null, descripcion
varchar(200), precio DECIMAL(8,2) CHECK(precio>0), fecha_caducidad DATE
CHECK(fecha_caducidad>GETDATE()),
CONSTRAINT Producto_PK PRIMARY KEY(id_producto),
CONSTRAINT Marca_FK FOREIGN KEY(marca_id) REFERENCES
Marca(id_marca),
CONSTRAINT Unidad_Medida_FK FOREIGN
KEY(unidad_medida_id) REFERENCES Unidad_Medida(id_unidad_medida),
CONSTRAINT Lote_FK FOREIGN KEY(lote_id) REFERENCES
Lote(id_lote));
GO

----- CAMPO CALCULADO-----
ALTER TABLE Producto ADD estado as
CASE
WHEN fecha_caducidad <= GETDATE() THEN 'Caducado'
ELSE 'Vigente'
END;

GO

-----
CREATE TABLE Paquete(id_paquete varchar(100) not null, producto_id varchar(100)
not null , unidades INTEGER CHECK(unidades>0),
CONSTRAINT Paquete_PK PRIMARY KEY(id_paquete),
CONSTRAINT Producto_FK FOREIGN KEY(producto_id)
REFERENCES Producto(id_producto));
GO
CREATE TABLE Inventario_Productos(id_inventario INTEGER not null
IDENTITY(1,1),sucursal_id INTEGER not null, producto_id varchar(100) not null,
existencia_unidades INTEGER, existencia_paquetes INTEGER,
existencia_medida DECIMAL(8,2),fecha DATE,
CONSTRAINT Inventario_PK PRIMARY KEY(id_inventario),
FOREIGN KEY(producto_id) REFERENCES Producto(id_producto),
FOREIGN KEY(sucursal_id) REFERENCES Sucursal(id_sucursal));
GO
CREATE TABLE Entrada_Producto(id_entrada_Prod INTEGER not null IDENTITY(1,1),
producto_id varchar(100) not null, cantidad_unidades INTEGER,
cantidad_paquetes INTEGER, cantidad_medida DECIMAL(8,2),
razon varchar(100), fecha_entrada SMALLDATETIME,
CONSTRAINT Entrada_Producto_PK PRIMARY KEY(id_entrada_Prod),
FOREIGN KEY(producto_id) REFERENCES Producto(id_producto));
GO
--36
CREATE TABLE Salida_Producto(id_salida_prod INTEGER not null IDENTITY(1,1),
producto_id varchar(100) not null, cantidad_unidades INTEGER,
cantidad_paquetes INTEGER, cantidad_medida DECIMAL(8,2),
razon varchar(100), fecha_entrada SMALLDATETIME,
CONSTRAINT Salida_Producto_PK PRIMARY KEY(id_salida_prod),
FOREIGN KEY(producto_id) REFERENCES Producto(id_producto));
GO
CREATE TABLE Proveedor(id_proveedor INTEGER not null IDENTITY(1,1),
nombre_responsable varchar(100) not null, apellido_responsable varchar(100) not
null, nombre_empresa varchar(100) not null,
CONSTRAINT Proveedor_PK PRIMARY KEY(id_proveedor));

```

GO

-----Tablas de control de entradas-----  
-----

```
CREATE TABLE Entradas_Proveedor(id_entrada_proveedor INTEGER not null
IDENTITY(1,1), proveedor_id INTEGER not null, fecha_entrada SMALLDATETIME,
CONSTRAINT Entradas_Proveedor_PK PRIMARY
KEY(id_entrada_proveedor),
CONSTRAINT Proveedor_K FOREIGN KEY(proveedor_id) REFERENCES
Proveedor(id_proveedor));
```

GO

```
CREATE TABLE Entrada_Devoluciones(id_devolucion INTEGER not null IDENTITY(1,1),
cliente_id INTEGER not null, empleado_id INTEGER not null, fecha_devolucion
SMALLDATETIME,
CONSTRAINT Entrada_Devoluciones_PK PRIMARY
KEY(id_devolucion),
CONSTRAINT Cliente_FK FOREIGN KEY(cliente_id)
REFERENCES Cliente(id_cliente),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado));
```

GO

-----Tablas de detalles de entradas-----  
-----

```
CREATE TABLE Detalle_Entradas_Devoluciones_Producto(id_entrada_dev INTEGER not
null IDENTITY(1,1), devolucion_id INTEGER not null, producto_id varchar(100),
cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
CONSTRAINT Detalle_Entradas_Devoluciones_Producto_PK
PRIMARY KEY(id_entrada_dev),
CONSTRAINT Entrada_Devoluciones_FK FOREIGN
KEY(devolucion_id) REFERENCES Entrada_Devoluciones(id_devolucion),
FOREIGN KEY(producto_id) REFERENCES
Producto(id_producto));
```

GO

--41

```
CREATE TABLE Detalle_Entradas_Proveedor_Producto(id_entrada_prov INTEGER not
null IDENTITY(1,1), entrada_proveedor_id INTEGER not null, producto_id
varchar(100),
cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
CONSTRAINT Detalle_Entradas_Proveedor_Producto_PK
PRIMARY KEY(id_entrada_prov),
CONSTRAINT Entradas_Proveedor_FK FOREIGN
KEY(entrada_proveedor_id) REFERENCES Entradas_Proveedor(id_entrada_proveedor),
FOREIGN KEY(producto_id) REFERENCES
Producto(id_producto));
```

GO

```
CREATE TABLE Detalle_Entradas_Proveedor_Paquete(id_entrada_prov_Paq INTEGER not
null IDENTITY(1,1), entrada_proveedor_id INTEGER not null, paquete_id
varchar(100),
cantidad_unidades INTEGER, precio DECIMAL(8,2),
CONSTRAINT Detalle_Entradas_Proveedor_Paquete_PK
PRIMARY KEY(id_entrada_prov_Paq),
FOREIGN KEY(entrada_proveedor_id) REFERENCES
Entradas_Proveedor(id_entrada_proveedor),
CONSTRAINT Paquete_FK FOREIGN KEY(paquete_id)
REFERENCES Paquete(id_paquete));
```

GO

```
CREATE TABLE Detalle_Proveedor_Medido(id_detalle_prov_med INTEGER not null
IDENTITY(1,1), entrada_proveedor_id INTEGER not null, paquete_id varchar(100),
cantidad_masa_volumen DECIMAL(8,2)
CHECK(cantidad_masa_volumen>0), precio DECIMAL(8,2),
CONSTRAINT Detalle_Proveedor_Medido_PK PRIMARY
KEY(id_detalle_prov_med),
FOREIGN KEY(entrada_proveedor_id) REFERENCES
Entradas_Proveedor(id_entrada_proveedor),
FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO
```

```
CREATE TABLE Detalle_Entradas_Devoluciones_Paquete(id_entrada_dev_Paq INTEGER
not null IDENTITY(1,1), devolucion_id INTEGER not null, paquete_Id varchar(100),
cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
CONSTRAINT Detalle_Entradas_Devoluciones_Paquete_PK
PRIMARY KEY(id_entrada_dev_Paq),
FOREIGN KEY(devolucion_id) REFERENCES
Entrada_Devoluciones(id_devolucion),
FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO
```

-----Tablas de control de Salidas-----  
-----

```
CREATE TABLE Salida_Orden(id_salida_orden INTEGER not null IDENTITY(1,1),
empleado_id INTEGER not null, cliente_id INTEGER not null, total_descripcion
varchar(100), total DECIMAL(8,2),
importe_exonerado DECIMAL(8,2), importe_exento
DECIMAL(8,2), importe_gravado_15 DECIMAL(8,2), importe_gravado_18
DECIMAL(8,2), isv_15 DECIMAL(8,2),
isv_18 DECIMAL(8,2), total_pagar DECIMAL(8,2),
fecha_orden SMALLDATETIME, direccion_entrega varchar(100),
CONSTRAINT Salida_Orden_PK PRIMARY
KEY(id_salida_orden),
FOREIGN KEY(cliente_id) REFERENCES
Cliente(id_cliente),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado));
GO
```

--46

```
CREATE TABLE Caja(id_caja varchar(3) not null, nombre_caja varchar(100),
CONSTRAINT Caja_PK PRIMARY KEY(id_caja));
GO
```

```
CREATE TABLE Salida_Caja(id_salida_caja INTEGER not null IDENTITY(1,1), caja_id
varchar(3), empleado_id INTEGER not null, cliente_id INTEGER not null,
total_descripcion varchar(100), total DECIMAL(8,2),
importe_exonerado DECIMAL(8,2), importe_exento
DECIMAL(8,2), importe_gravado_15 DECIMAL(8,2), importe_gravado_18
DECIMAL(8,2), isv_15 DECIMAL(8,2),
isv_18 DECIMAL(8,2), total_pagar DECIMAL(8,2)
CHECK(total_pagar>0), fecha_salida_caja SMALLDATETIME,
CONSTRAINT Salida_Caja_PK PRIMARY
KEY(id_salida_caja),
CONSTRAINT Caja_FK FOREIGN KEY(caja_id) REFERENCES
Caja(id_caja),
FOREIGN KEY(cliente_id) REFERENCES
```

```

Cliente(id_cliente),
                                FOREIGN KEY(emplado_id) REFERENCES
Empleado(id_empleado));
GO

-----Tablas de detalles de las salidas-----
-----
CREATE TABLE Detalle_Orden_Producto(id_detalle_orden_p INTEGER not null
IDENTITY(1,1), salida_orden_id INTEGER not null, producto_id varchar(100),
                                cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
                                CONSTRAINT Detalle_Orden_Producto_PK PRIMARY
KEY(id_detalle_orden_p),
                                CONSTRAINT Salida_Orden_FK FOREIGN
KEY(salida_orden_id) REFERENCES Salida_Orden(id_salida_orden),
                                FOREIGN KEY(producto_id) REFERENCES
Producto(id_producto));
GO

CREATE TABLE Detalle_Orden_Paquete(id_detalle_orden_paq INTEGER not null
IDENTITY(1,1), salida_orden_id INTEGER not null, paquete_id varchar(100),
                                cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
                                CONSTRAINT Detalle_Orden_Paquete_PK PRIMARY
KEY(id_detalle_orden_paq),
                                FOREIGN KEY(salida_orden_id) REFERENCES
Salida_Orden(id_salida_orden),
                                FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO
--50
CREATE TABLE Detalle_Orden_Medido(id_detalle_orden_med INTEGER not null
IDENTITY(1,1), salida_orden_id INTEGER not null, paquete_id varchar(100),
                                cantidad_masa_volumen DECIMAL(8,2)
CHECK(cantidad_masa_volumen>0), precio DECIMAL(8,2),
                                CONSTRAINT Detalle_Orden_Medido_PK PRIMARY
KEY(id_detalle_orden_med),
                                FOREIGN KEY(salida_orden_id) REFERENCES
Salida_Orden(id_salida_orden),
                                FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO

CREATE TABLE Detalle_Caja_Paquete(id_detalle_caja_paq INTEGER not null
IDENTITY(1,1), salida_caja_id INTEGER not null, paquete_id varchar(100),
                                cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
                                CONSTRAINT Detalle_Caja_Paquete_PK PRIMARY
KEY(id_detalle_caja_paq),
                                CONSTRAINT Salida_Caja_FK FOREIGN KEY(salida_caja_id)
REFERENCES Salida_Caja(id_salida_caja),
                                FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO

CREATE TABLE Detalle_caja_Medido(id_detalle_caja_med INTEGER not null
IDENTITY(1,1), salida_caja_id INTEGER not null, paquete_id varchar(100),
                                cantidad_masa_volumen DECIMAL(8,2)
CHECK(cantidad_masa_volumen>0), precio DECIMAL(8,2),
                                CONSTRAINT Detalle_caja_Medido_PK PRIMARY
KEY(id_detalle_caja_med),

```



```

FOREIGN KEY(salida_caja_Id) REFERENCES
Salida_Caja(id_salida_caja),
FOREIGN KEY(paquete_id) REFERENCES
Paquete(id_paquete));
GO

CREATE TABLE Detalle_Caja_Producto(id_detalle_caja_p INTEGER not null
IDENTITY(1,1), salida_caja_Id INTEGER not null, producto_id varchar(100),
cantidad_unidades INTEGER CHECK(cantidad_unidades>0),
precio DECIMAL(8,2),
CONSTRAINT Detalle_Caja_Producto_PK PRIMARY
KEY(id_detalle_caja_p),
FOREIGN KEY(salida_caja_Id) REFERENCES
Salida_Caja(id_salida_caja),
FOREIGN KEY(producto_id) REFERENCES
Producto(id_producto));
GO

--54
CREATE TABLE Distribucion(id_distribuidor INTEGER not null IDENTITY(1,1),
empleado_id INTEGER not null, salida_orden_id INTEGER not null, fecha
SMALLDATETIME,
CONSTRAINT Distribucion_PK PRIMARY
KEY(id_distribuidor),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado),
FOREIGN KEY(salida_orden_id) REFERENCES
Salida_Orden(id_salida_orden));
GO

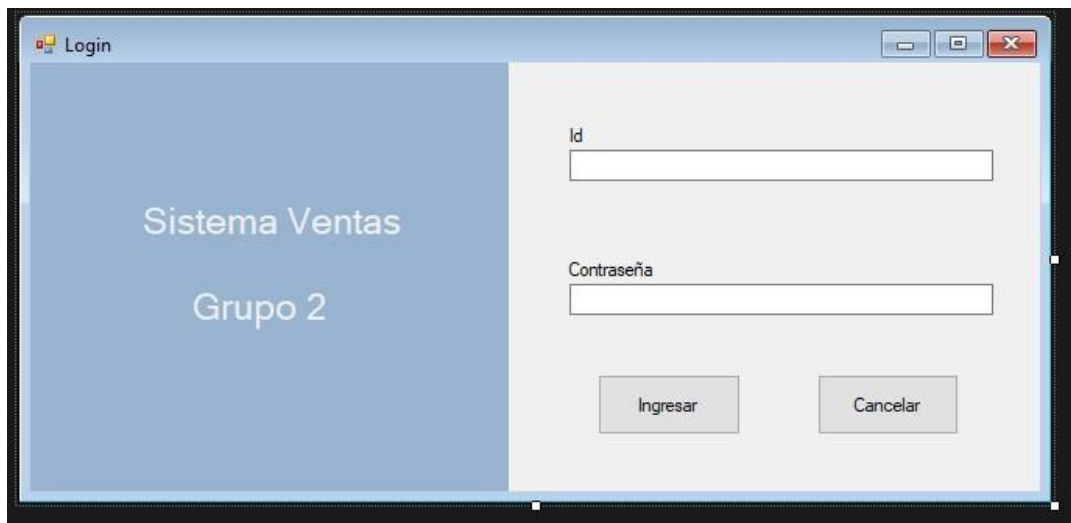
CREATE TABLE Metodo_Pago(id_metodo_pago INTEGER not null IDENTITY(1,1) ,
nombre_metodo varchar(100),
CONSTRAINT Metodo_Pago_PK PRIMARY
KEY(id_metodo_pago));
GO

CREATE TABLE Factura(id_factura varchar(19), empleado_id INTEGER not null ,
sucursal_id INTEGER not null , salida_caja_id INTEGER not null UNIQUE ,
salida_orden_id INTEGER not null UNIQUE,
CAI varchar(30), fecha_limite_emicion DATE, rango
varchar, no_correlativo_orden_compra_exenta varchar(100),
no_correlativo_constancia_reg_exonerado varchar(100),
no_Identi_reg_sag varchar(100), fecha_factura SMALLDATETIME,
CONSTRAINT Factura_PK PRIMARY KEY(id_factura),
FOREIGN KEY(empleado_id) REFERENCES
Empleado(id_empleado),
FOREIGN KEY(sucursal_id) REFERENCES
Sucursal(id_sucursal),
FOREIGN KEY(salida_caja_Id) REFERENCES
Salida_Caja(id_salida_caja),
FOREIGN KEY(salida_orden_id) REFERENCES
Salida_Orden(id_salida_orden));
GO
--57
CREATE TABLE Factura_X_Metodo_Pago( metodo_pago_id INTEGER not null
IDENTITY(1,1), factura_id varchar(19),
CONSTRAINT Metodo_Pago_FK FOREIGN
KEY( metodo_pago_id) REFERENCES Metodo_Pago(id_metodo_pago),
CONSTRAINT Factura_FK FOREIGN KEY(factura_id)
REFERENCES Factura(id_factura));

```

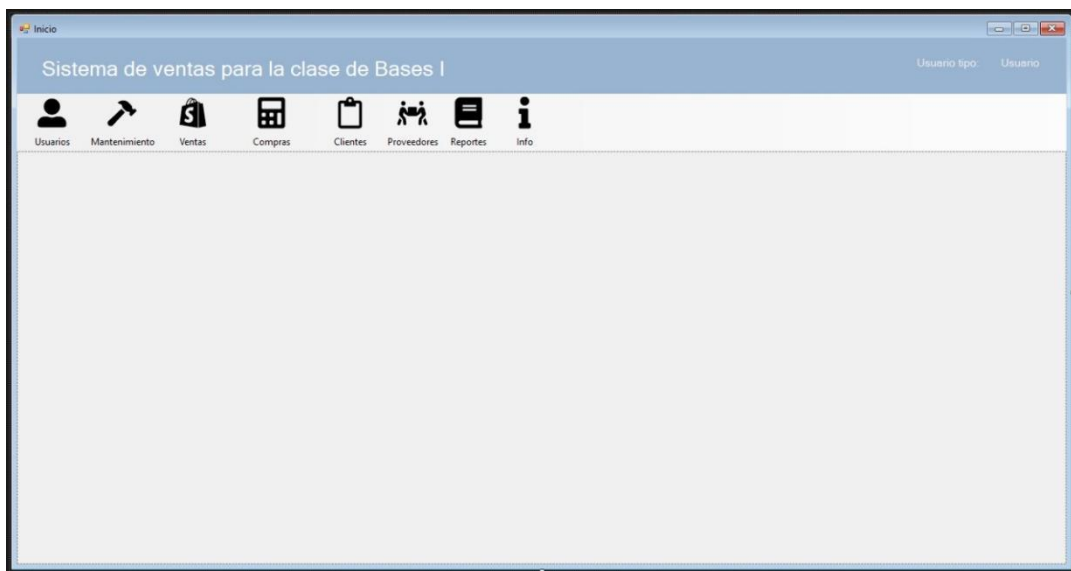
```
INSERT INTO Departamento('Francisco Morazan');
INSERT INTO Ciudad( 1,'Tegucigalpa');
INSERT INTO Colonia( 1,'Kennedy');
INSERT INTO Sucursal( 1,'080119990032412','Darwin´s Kennedy', 'Mercaditos
Darwin´s', 'Entrada por anillo periferico junto a la Colonia', 'A454');
INSERT INTO Estado_Civil('Soltero');
INSERT INTO Estado_Civil('Casado');
INSERT INTO Persona('08011998005239',1,1,'Carlos','Alberto', 'Gonzalo',
'Alvarez', 'Masculino');
INSERT INTO Cargo('Gerente');
INSERT INTO Cargo('Ventas');
INSERT INTO Empleado(1,1,1,'Colonia Miraflores, Sector 1, Bloque 4, Casa 23');
INSERT INTO Rol('no se que rol', GETUTCDATE());
INSERT INTO Permisos(1,'no se que permiso', GETUTCDATE() );
```

## Capturas de Pantalla del Sistema de Ventas y Facturación



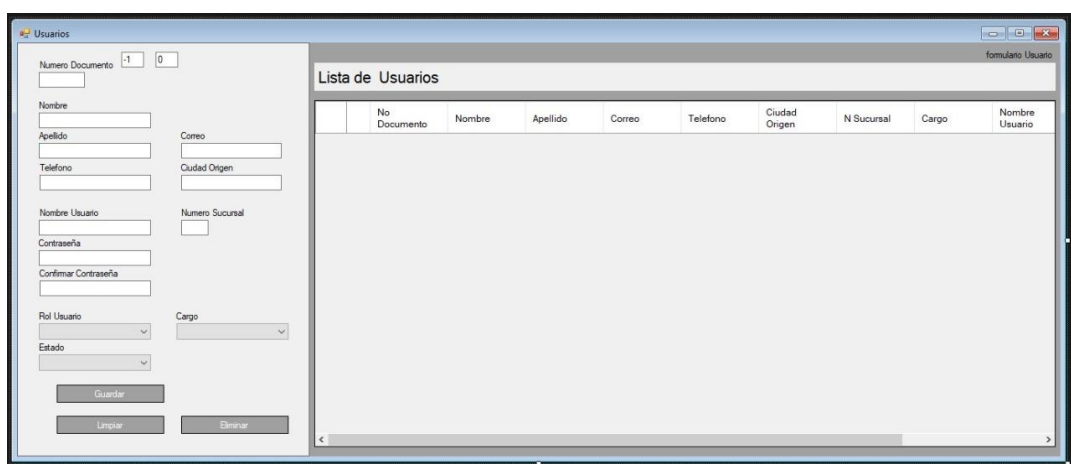
The screenshot shows a login window titled "Login". On the left, a blue panel displays "Sistema Ventas" and "Grupo 2". On the right, there are two input fields labeled "Id" and "Contraseña". Below these fields are two buttons: "Ingresar" and "Cancelar".

1. Pantalla de Ingreso al sistema con usuario y contraseña



The screenshot shows the main menu of the "Sistema de ventas para la clase de Bases I". The top bar includes the title and "Usuario tipo: Usuario". Below the title is a navigation bar with icons and labels: Usuarios, Mantenimiento, Ventas, Compras, Clientes, Proveedores, Reportes, and Info. The main area is a large, empty light gray rectangle.

2. Visualización General de los Componentes del Sistema



The screenshot shows the "Usuarios" (Users) management screen. It is divided into two main sections. The left section is a form for creating or editing a user, with fields for: Numero Documento, Nombre, Apellido, Correo, Telefono, Ciudad Origen, Nombre Usuario, Numero Sucursal, Contraseña, Confirmar Contraseña, Rol Usuario, Estado, and Cargo. There are buttons for "Guardar", "Limpiar", and "Eliminar". The right section is titled "Lista de Usuarios" and contains a table with the following columns: No Documento, Nombre, Apellido, Correo, Telefono, Ciudad Origen, N Sucursal, Cargo, and Nombre Usuario. The table is currently empty.

3. Lista y Panel de creación de Usuarios

Mantenimiento Rol de Usuarios

Agregar un nuevo Rol

Rol  -1  
 0

Menus Disponibles

- ☐ Usuario
- ☐ Mantenedor
- ☐ Ventas
- ☐ Compras
- ☐ Clientes
- ☐ Proveedores
- ☐ Reportes
- ☐ Info

**Roles de Usuarios**

Rol Id	Descripcion Rol
--------	-----------------

formulario Rol de Usuarios

#### 4. Visualización y Panel de Creación de Usuarios

Mantenimiento Cargos de Empleados

Agregar un nuevo Cargo

Cargo

**Roles de Usuarios**

Cargo Id	Descripcion Cargo
----------	-------------------

formulario Cargos Empleados

#### 5. Visualización y Panel de creación de cargos

## Conclusiones

1. La demanda creciente de los sistemas informáticos para el manejo de negocios permite observar una clara incidencia de la importancia del procesamiento y almacenamiento de datos para las empresas y negocios, pudiendo ser de vital importancia para la proyección de la rentabilidad de un negocio a largo plazo.
2. La creación de bases de datos relacionales permite una visualización de las partes que conforman un sistema completa, desglosando dichos componentes en entidades relacionadas a través de un identificador asociativo, elemento que facilita la comunicación con la base de datos y permite una arquitectura comprensible a través de los denominados *querys* o diagramas generados por el motor de bases de datos.
3. La interfaz gráfica que permite la manipulación efectiva de la base de datos permite que un usuario ajeno al mundo de la programación pueda interactuar con los registros almacenados en las tablas asignadas, logrando uno de los elementos esenciales de la tecnología: Incluir a la mayor cantidad de usuarios potenciales para rendir provecho de los productos de software que son desarrollados, y por tanto, maximizar la rentabilidad de la creación de dichas piezas informáticas.
4. El manejo de los elementos del Servicio de Administración de Rentas (SAR) supone la comprensión de los elementos tributarios que deben integrarse y considerarse al momento de desarrollar un sistema que tiene por objetivo manejar, de alguna forma, cantidades variables de dinero e inventario de productos, desarrollando buenas prácticas de diseño y lógica, útiles al momento de trabajar con una empresa que requiere de un sistema funcional y que obedece las leyes de un país.

### **Link de video defensa**

<https://youtu.be/rvJr8-Ct9Ik>