



Proyecto  
Final



# INFORME

**POO | BASE DE  
DATOS**



# *Tecno e*

¿Listo para el **UPGRADE?**

...

Gabriel Cepeda

# INDICE

1. INTRODUCCIÓN
2. PLANTEAMIENTO DEL PROBLEMA QUE ABORDA LA APLICACIÓN
3. DISEÑO DE LA BASE DE DATOS
4. SENTENCIAS SQL DESARROLLADAS
5. CONCLUSIONES
6. ANEXOS



# INTRODUCCION

TecnoC es un sistema de escritorio que permite la automatización de inventario y facturación para una tienda de componentes de hardware. El mismo nace con el fin de brindarle una experiencia de comodidad tanto visual como funcional, así como también de eficiencia a los dueños de Tiendas de este tipo de servicios.

Nuestro sistema inicialmente con la versión 1.0 fue creado con un sistema de archivos. El mismo consiste en ir guardando todos los datos e información en archivos, para luego poder ser recuperados por el usuario. Esto trajo consigo muchos problemas a la hora de mantener los datos actualizados en tiempo real, así como también un alto costo para los usuarios en mantener el sistema de archivos bien cuidado, puesto a que, si este llega a ser eliminado o corrompido, el usuario podría perder toda la información. Este y otros problemas más como lo son la eficiencia, seguridad y la integridad de los datos, dan la idea a este pequeño grupo de integrarle Base de Datos Relacional para así contrarrestar muchos de estos puntos negativos que existen en los sistemas de archivos.

Pero... ¿Qué es una Base de Datos Relacional? Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila de la tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen atributos de los datos, y cada registro generalmente tiene un valor para cada atributo, lo que facilita el establecimiento de las relaciones entre los puntos de datos.

La diferencia entre el sistema de archivos y la base de datos es que el sistema de archivos administra solo el acceso físico, mientras que la base de datos administra tanto el acceso físico como lógico a los datos. En un sistema de archivos, los archivos permiten almacenar datos mientras que una base de datos es una colección de datos organizados

Con esta nueva idea es que nace la versión 2.0 de nuestro sistema de escritorio "TecnoC". La cual a continuación les estaremos presentando más a fondo sus características únicas del mercado.

# PLANTEAMIENTO

En la actualidad, un gran número de pequeñas y medianas empresas de la República Dominicana deben su éxito a la automatización de sus tediosos procesos manuales por medio del uso de herramientas como las bases de datos. Una buena organización de los datos, ligada con una adecuada facilidad de acceso a los mismos, representa una considerable ventaja para cualquier establecimiento comercial que desee mantenerse al tanto de sus productos, y proporcionar un servicio rápido y de calidad.

A pesar de estas evidentes ventajas, la tecnología del manejo de las bases de datos no ha alcanzado a todos los rincones de nuestro país, y su falta afecta a miles de empresas y compañías crecientes. Tal es el caso de una pequeña empresa de ventas de piezas de computador, quienes no poseen un sistema digital de base de datos para manejar y administrar su inventario, sus clientes y sus ventas. Debido a esto, la empresa se está viendo rezagada en el tan competitivo mercado actual de piezas de computador, donde predomina la automatización y el servicio eficiente.

Dada esta realidad, para la compañía en cuestión resultaría esencial implementar un sistema de administración y manejo con base de datos que le permita mantenerse a la par con las tendencias tecnológicas del momento, y, a la vez, mejorar considerablemente su gestión laboral interna, desde el manejo de registro de sus productos hasta el control de la facturación y las ventas.

Para lograrlo, presentamos Tecno C, una aplicación desarrollada con el propósito de agilizar la gestión de tiendas y puntos de venta, teniendo como objetivo principal asistir a los vendedores a lo largo del proceso de facturación, haciéndolo más simple e intuitivo. Esto, a su vez, se extiende hasta los clientes, quienes tendrán la posibilidad de recibir una mejor asistencia a la hora de seleccionar y conformar su computador según sus deseos.

En cuanto a la tienda de piezas de computador, son cuatro los productos o componentes principales que vende, cada uno de los cuales posee distintas propiedades que pueden resultar interesante considerar a la hora de ensamblar una PC. Estos son, entre otros:

- Tarjetas Madre (MotherBoard); de las cuales se conocen la marca, modelo, tipo de conector o Socket para microprocesador, tipo de Memoria RAM con que funciona y el tipo de conexiones de discos duros que acepta.
- Microprocesadores; de los cuales se conoce la marca y modelo, su tipo de conector o Socket y su velocidad de procesamiento en GHz.
- Memorias RAM; de las cuales se conoce su marca, su capacidad en MB o GB, y su tipo (DDR, DDR-2, DDR-3, SDRAM, etc.).
- Discos Duros; de los cuales se conoce marca y modelo, su capacidad de almacenamiento en GB, y su tipo de conexión (IDE, SATA, SATA-2, SATA-3)

Además de cada una de estas propiedades particulares, de manera general, de todos los componentes se almacena su precio, su cantidad disponible en stock y su número de

serie único. Cada una de estas propiedades serán manejadas por Tecno C empleando una desarrollada e íntegra base de datos.

Además de manejar la existencia de productos disponibles, nuestra aplicación también servirá como un gestor general de la tienda, permitiéndole mantener un registro actualizado de cada una de las facturas procesadas hasta el momento, de los clientes que se han registrado con la tienda, de las órdenes de compra realizadas a distribuidores sobre un producto, e incluso de un sistema estadístico proporciona un gran número de datos relevantes para el manejo de la tienda, pues le permitirá estudiar las tendencias relativas mensuales del mercado. De forma adicional, este sistema también permite observar los clientes con mayores compras, y los vendedores con mayores ventas, así como su respectiva comisión por las mismas; incrementando aún más la facilidad de manejo de la empresa como tal.

Finalmente, esta aplicación también permitirá el manejo de combos, que son ofertas únicas de un conjunto de productos que se venden como uno, a un precio reducido que es determinado por un descuento específico.

# DISEÑO

Cuando se habla de la creación y el desarrollo de cualquier base de datos, resulta importante establecer, en primer lugar, todas las características y particularidades propias de su llamado diseño de base de datos. Para estas, un buen diseño juega un rol fundamental para garantizar la consistencia e integridad de los datos, la eliminación de cualquier posible redundancia de información, y la eficiente ejecución de consultas y aplicaciones de alto rendimiento. Es por esto que, en nuestro caso particular, gran parte del tiempo invertido en el proceso de desarrollo del presente proyecto final fueron dedicados a diseñar y crear un buen diseño general que facilite la comprensión de nuestro proyecto, a la vez que pueda servir como representación de cada uno de sus componentes.

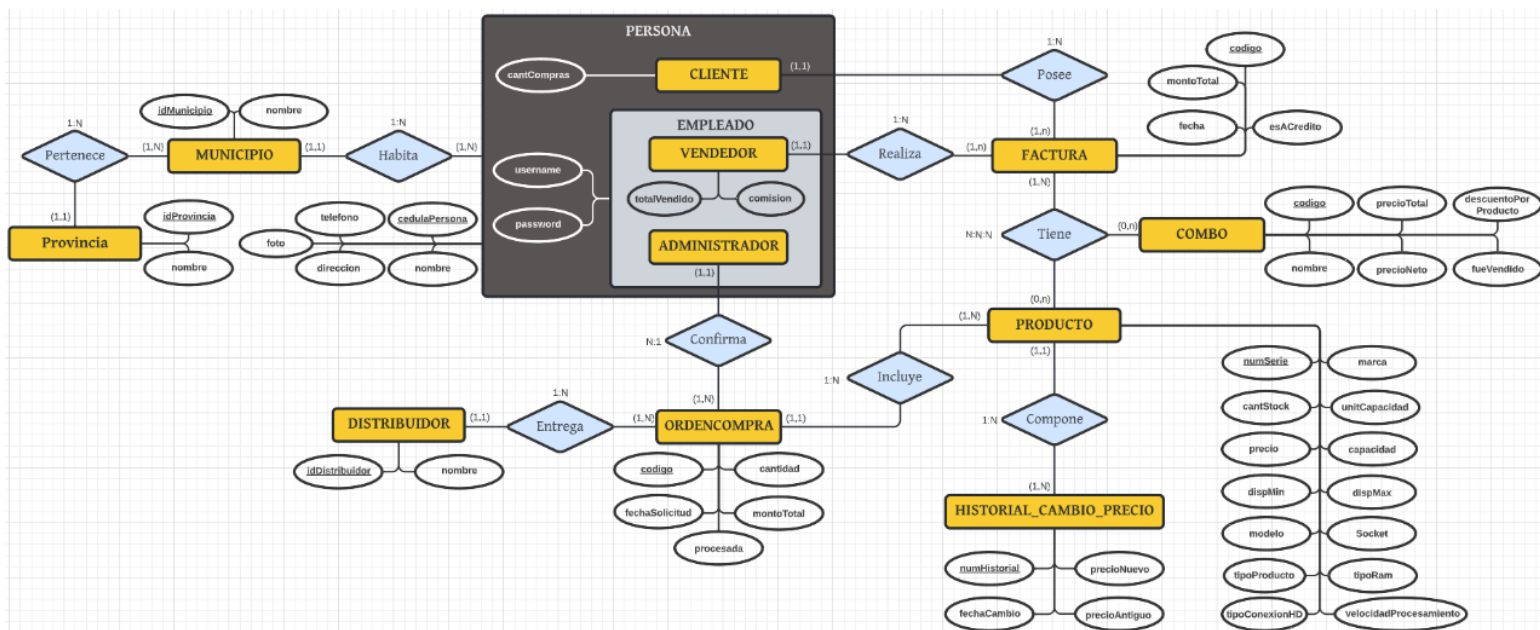
Para iniciar el proceso de diseño de la base de datos de nuestro proyecto final TecnoC, esbozamos un diagrama Entidad-Relación, un tipo de diagrama de flujo que ilustra de forma gráfica el comportamiento de las entidades que componen la base de datos, así como la forma en la que se relacionan entre sí dentro del sistema. El diagrama de nuestra base de datos TecnoC fue realizado utilizando la aplicación web Lucidchart, siguiendo la notación Chen de diagramas entidad-relación. De manera consiguiente, a partir de este DER, se desarrolló el modelo relacional de la base de datos, con el cual se asegura la normalización de la misma, y se inician a definir las tablas y las llaves primarias y foráneas que fundamentan todo el sistema. En último lugar, el proceso de diseño nos llevó a realizar un detallado diccionario de datos que contenga cada una de las entidades, y la definición de sus respectivos atributos.

De forma general, nuestro proceso de diseño concluyó teniendo un total de 13 entidades diferentes, cada una con su conjunto determinado de atributos que las definen, y con un total de 9 relaciones que las unen entre sí; de las cuales se puede resaltar un caso interesante de relación de “herencia” existente entre 5 de las entidades ya mencionadas. En cuanto a las cardinalidades que definen estas relaciones, se puede observar cómo cada una de ellas están definidas por una relación 1 a muchos (1:N), a excepción de una relación ternaria muchos a muchos (N:N:N). Teniendo esto en cuenta, las ya mencionadas 13 entidades dieron lugar a un total de 16 tablas relacionales, con las cuales se completan las relaciones existentes en nuestro proyecto final.

A continuación, se presentan y se explican en detalle cada uno de estos ya mencionados componentes resultantes del tan importante proceso inicial de diseño de nuestra base de datos. En primer lugar, se dará inicio con nuestro diagrama Entidad-Relación:

## DIAGRAMA ENTIDAD RELACION PROYECTO FINAL

*TecnoC: Tienda de Componentes de Computador*



### EXPLICACION DE LAS ENTIDADES

**Entidad Persona:** Esta entidad es el padre de otras entidades. Esta entidad hace referencia a las cualidades de una persona que consideramos necesarias conocer para nuestro programa.

**Entidad Provincia:** Esta entidad hace referencia a las provincias previamente insertadas por el administrador de Bases de Datos, las cuales para nuestro sistemas está compuesta por las provincias de la Republica Dominicana.

**Entidad Municipio:** Esta es una entidad que está compuesta por los municipios de cada provincia de la Republica Dominicana. Al igual que la entidad provincia, esta es insertada previamente por el administrador de base de datos para su uso.

**Entidad Cliente:** Esta es una entidad débil que necesita de la entidad persona para ser creada. Esta está compuesta por los atributos de interés propio de la empresa sobre un cliente.

**Entidad Empleado:** Esta entidad, aunque es el padre de otras, también es una entidad débil, pues el empleado primero necesita ser una Persona para luego ser creado como empleado. Consta de 2 entidades hijas tales como: Administrador y Vendedor.

**Entidad Administrador:** Esta entidad es una entidad débil pues es necesario primero haber tenido al Administrador registrado como empleado. La misma hace referencia a los administradores del sistema.



**Entidad Vendedor:** Esta entidad al igual que Administrador es una entidad débil, ya que necesita haberse creado primero como Empleado para luego crearse como Vendedor. Esta entidad hace referencia a los vendedores que tendrá el sistema.

**Entidad Distribuidor:** Esta entidad es una entidad fuerte, pero para la versión 2.0 (la actual) no tiene mucha funcionalidad en el sistema. Pues únicamente está para asignarle a una orden de compra a la hora de un administrador confirmarla.

**Entidad ordenCompra:** Esta entidad es aquella que hace referencia a las órdenes de compra que se crean automáticamente cuando un producto llega a su disponibilidad mínima. La misma tiene que ser confirmada por el Administrador para proceder con el pedido al distribuidor.

**Entidad Producto:** Dicha entidad es el propósito de nuestro sistema, pues compone los atributos propios de los productos que venderemos en la versión actual de nuestro sistema.

**Entidad Combo:** Esta entidad hace referencia a las ofertas o más bien a los combos que serán creados por el Administrador para su venta en el sistema.

**Entidad HistorialCambioPrecio:** Esta entidad fue creada con el fin de tener un historial de cuando los precios de los productos sean actualizados por el Administrador. Para así tener una mejor claridad de lo que ha transcurrido en el sistema.

**Entidad Factura:** Esta entidad es aquella que hace referencia a la compra de uno o varios artículos del sistema. La misma consiste en la factura que es creada al cliente a la hora de procesarse la venta.

## EXPLICACION DE LAS RELACIONES

**Verbo Pertenece:** Es aquel que relaciona al municipio con la provincia. Esta relación consiste en que un municipio pertenece a una provincia, y una provincia le pertenece a 1 o muchos municipios.

**Verbo Habita:** Este verbo relaciona a la Entidad Persona con Municipio, pues indica el municipio en el que habita una persona. La relación de este consiste en que una persona habita en un municipio, y en un municipio habitan 1 o muchas personas.

**Verbo Confirma:** Es aquel que indica la relación de Administrador con Orden de Compra, pues indica qué Administrador confirmó dicha Orden De Compra. La relación de este consiste en que una Orden de Compra es confirmada por 1 y solo 1 Administrador, y que 1 Administrador confirma 1 o muchas Órdenes de Compra.

**Verbo Entrega:** Este verbo relaciona a Orden de Compra con Distribuidor, pues cuando el Administrador confirma una Orden de Compra se la asigna a un Distribuidor. Esta relación consiste en que una Orden de Compra es entregada por 1 y solo 1 Distribuidor, y un Distribuidor entrega 1 o muchas Órdenes de Compra.

**Verbo Compone:** Este verbo es el que relaciona a Producto con Historial Cambio Precio, pues indica a qué Producto se le ha cambiado el precio entre otras informaciones. La relación que existe de dicho verbo consiste en que un Historial Cambio Precio es compuesta por 1 y solo 1 Producto, y que 1 Producto puede tener 1 o varios Historial Cambio Precio.

**Verbo Realiza:** Este verbo relaciona a la Entidad Vendedor y la Entidad Factura, pues indica que Vendedor realizo una Factura. Esta relación consiste en que un Vendedor realiza 1 o muchas Facturas, mientras que una Factura es realizada por 1 y solo 1 Vendedor.

En segundo lugar, y tras haber analizado cada uno de los componentes del diagrama entidad-relación resultante, el proceso de diseño continuó con el desarrollo de un modelo relacional para el sistema, acompañado de un diccionario de datos propio para cada una de las tablas que lo componen. Los resultados de dichos procesos de análisis pueden ser observados a continuación:

PRODUCTO		PROVINCIA		HISTORIAL_CAMBIO_PRECIO	
PK	numSerie	PK	idProvincia	PK	numHistorial
	cantStock		nombre		fechaCambio
	precio				precioAntiguo
	marca				precioNuevo
	modelo			FK	numSerie
	dispMin				
	dispMax				
	socket				
	tipoProducto				
	capacidad				
	unitCapacidad				
	tipoRam				
	velocidadProcesamiento				
COMBO		MUNICIPIO		ORDEN_COMPRA	
PK	codigo	PK	idMunicipio	PK	codigo
	nombre		nombre		cantidad
	precioTotal				fechaSolicitud
	precioNeto				montoTotal
	descuentoPorProducto				procesada
	fechaVendido			FK	cedulaAdministrador
				FK	numSerie
				FK	idDistribuidor
COMBO_PRODUCTO		DISTRIBUIDOR			
PK FK	numSerie	PK	idDistribuidor		
PK FK	codigoCombo		nombre		
	cantidad				
CLIENTE					
PK FK	cedula				
	cantCompras				

PERSONA		FACTURA	
PK	cedulaPersona	PK	codigo
	nombre		fecha
	direccion		montoTotal
	telefono		esACredito
	foto	FK	cedulaCliente
FK	idMunicipio	FK	cedulaVendedor
EMPLEADO		DETALLE_FACTURA_COMBO	
PK FK	cedulaPersona	PK FK	codigoFactura
	username	PK FK	codigoCombo
	password		cantidad
VENDEDOR		DETALLE_FACTURA_PRODUCTO	
PK FK	cedulaEmpleado	PK FK	codigoFactura
	totalVendido	PK FK	numSerie
	comision		cantidad

## Diccionario de Datos

Tabla	Factura		Fecha	21/7/2022
Descripción	Tabla que contiene las informaciones de las facturas			
Campo	Tipo de Dato	Tamaño	Default	Descripción
codigo	INT	5		Identificador único para cada factura.
fecha	Datetime	16		Fecha en que se creó la factura
montoTotal	Decimal	7		Monto total en la factura
esACredito	Bit	1	0	Indica si la factura es a credito o no
cedulaCliente	Varchar	12		Cédula del cliente
cedulaVendedor	Varchar	12		Cédula del vendedor
Relaciones			Claves	
Relacion con la tabla cliente			cedulaCliente (PK)	
Relación con la tabla Vendedor			cedulaVendedor(PK)	

Tabla	Persona		Fecha	21/7/2022
Descripción	Tabla que contiene las informaciones de las personas			
Campo	Tipo de Dato	Tamaño	Default	Descripción
cedulaPersona	Varchar	14		Identificador único para cada persona.
nombre	Varchar	50		Nombre de la persona
direccion	Varchar	50		Dirección de la persona
telefono	Varchar	12		Teléfono de la persona
foto	varbinary	MAX		Imagen guardada de la Persona
Relaciones			Claves	

Tabla	Detalle_Factura_Producto		Fecha	21/7/2022
Descripción	Tabla que contiene un producto comprado perteneciente a una factura en			
Campo	Tipo de Dato	Tamaño	Default	Descripción
codigoFactura	INT	5		Identificador único para cada factura.
numSerie	INT	10		Número de serie del producto
cantidad	INT	7		Cantidad del producto
Relaciones			Claves	
Relacion con la tabla factura			codigoFactura (PK)	
Relación con la tabla producto			numSerie(PK)	

<b>Tabla</b>	Detalle Factura Combo		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene un combo comprado perteneciente a una factura en			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
codigoFactura	INT	5		Identificador único para cada factura.
codigoCombo	INT	5		Identificador único para el combo.
cantidad	INT	7		cantidad del combo
<b>Relaciones</b>			<b>Claves</b>	
Relacion con la tabla factura			codigoFactura (PK)	
Relación con la tabla combo			codigoCombo(PK)	

<b>Tabla</b>	Combo		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los Combos			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
codigo	INT	5		Identificador único para un combo
nombre	Varchar	7		Nombre del combo
precioTotal	Decimal	7		Precio total del combo
precioNeto	Decimal	7		Precio neto (sin descuento) del combo
descuentoPorProducto	Decimal	4		Descuento aplicado al combo
fueVendido	BIT	1	0	Indica si el combo fue vendido o no
<b>Relaciones</b>			<b>Claves</b>	

<b>Tabla</b>	OrdenCompra		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Orden de compra para la suma de stock de un producto			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
codigo	INT	5		Identificador único para cada orden compra.
cantidad	INT	5		Cantidad de productos en la orden de compra
fechaSolicitud	Datetime	16		Fecha en que se realizó la orden de compra
montoTotal	Decimal	7		El monto total de la orden de compra
procesada	Bit	1		Indica si la orden se encuentra confirmada o no
cedulaAdministrador	Varchar	12	null	Cedula del administrador
numSerie	INT	10		Número de serie del producto
idDistribuidor	INT	5		Identificador único para cada Distribuidor.
<b>Relaciones</b>			<b>Claves</b>	
Relacion con la tabla Administrador			cedulaAdministrador (PK)	
Relación con la tabla Distribuidor			idDistribuidor (PK)	
Relación con la tabla Producto			numSerie(PK)	

<b>Tabla</b>	Cliente		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los Clientes			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
cedulaPersona	Varchar	12		Identificador único de un Cliente
cantCompras	INT	7		Cantidad de compras de un cliente
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Persona			cedulaPersona(PK)	

<b>Tabla</b>	Vendedor		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los vendedores			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
cedulaEmpleado	INT	5		Identificador único para un vendedor
totalVendido	Decimal	7		Monto total de ventas realizadas
comision	Decimal	7		Comisión asignada al empleado por venta
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Empleado			cedulaEmpleado(PK)	

<b>Tabla</b>	Distribuidor		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los distribuidores			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
idDistribuidor	INT	5		Identificador único para un Distribuidor
nombre	Varchar	50		Nombre del distribuidor
<b>Relaciones</b>			<b>Claves</b>	

<b>Tabla</b>	Administrador		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los Administradores			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
cedulaEmpleado	INT	5		Identificador único para un administrador
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Empleado			cedulaEmpleado(PK)	

<b>Tabla</b>	Empleado		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los Empleador			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
cedulaPersona	Varchar	14		Identificador único de un empleado
username	Varchar	15		Nombre de usuario del empleado
password	Varchar	16		Contraseña del empleado
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Persona			cedulaPersona(PK)	

<b>Tabla</b>	Producto		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los productos			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
numSerie	INT	5		Identificador único para cada producto.
cantStock	INT	7		Cantidad disponible del producto
precio	Decimal	7		Precio del producto
marca	Varchar	50		Marca del producto
modelo	Varchar	50		Modelo del producto
dispMin	INT	3		Disponibilidad mínima del producto
dispMax	INT	3		Disponibilidad máxima del producto
tipoConexionHD	Varchar	5	null	Tipo de Conexion HD que posee el producto
socket	Varchar	50	null	Socket del producto
tipoProducto	Varchar	50		Tipo del producto
capacidad	INT	7	null	Capacidad de almacenamiento del producto
unitCapacidad	Decimal	2	null	Unidad de medida de la capacidad de un producto
tipoRam	Varchar	5	null	Tipo de Ram
velocidadProcesamiento	Decimal	7	null	Velocidad de procesamiento
<b>Relaciones</b>			<b>Claves</b>	

<b>Tabla</b>	Provincia		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las provincias del país.			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
idProvincia	INT	5		Identificador único para cada provincia
nombre	Varchar	50		Nombre de la provincia
<b>Relaciones</b>			<b>Claves</b>	

<b>Tabla</b>	Combo_Producto		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene cuáles productos se encuentran en un combo			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
numSerie	INT	5		Identificador único para un producto
codigoCombo	INT	5		Identificador único para un combo
<b>Relaciones</b>			<b>Claves</b>	
Relacion con la tabla combo			codigoCombo (PK)	
Relación con la tabla Producto			numserie(PK)	

<b>Tabla</b>	Historial_Cambio_Precio		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene un registro de los cambios de precio de los productos			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
numHistorial	INT	5		Identificador único para un registro
fechaCambio	Datetime	16		Fecha y hora en la que se produjo el cambio
precioAntiguo	Decimal	11		Antiguo precio el producto
precioNeto	Decimal	11		Nuevo precio del producto
numSerie	Varchar	10		Identificador único de un producto
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Producto			numSerie(PK)	

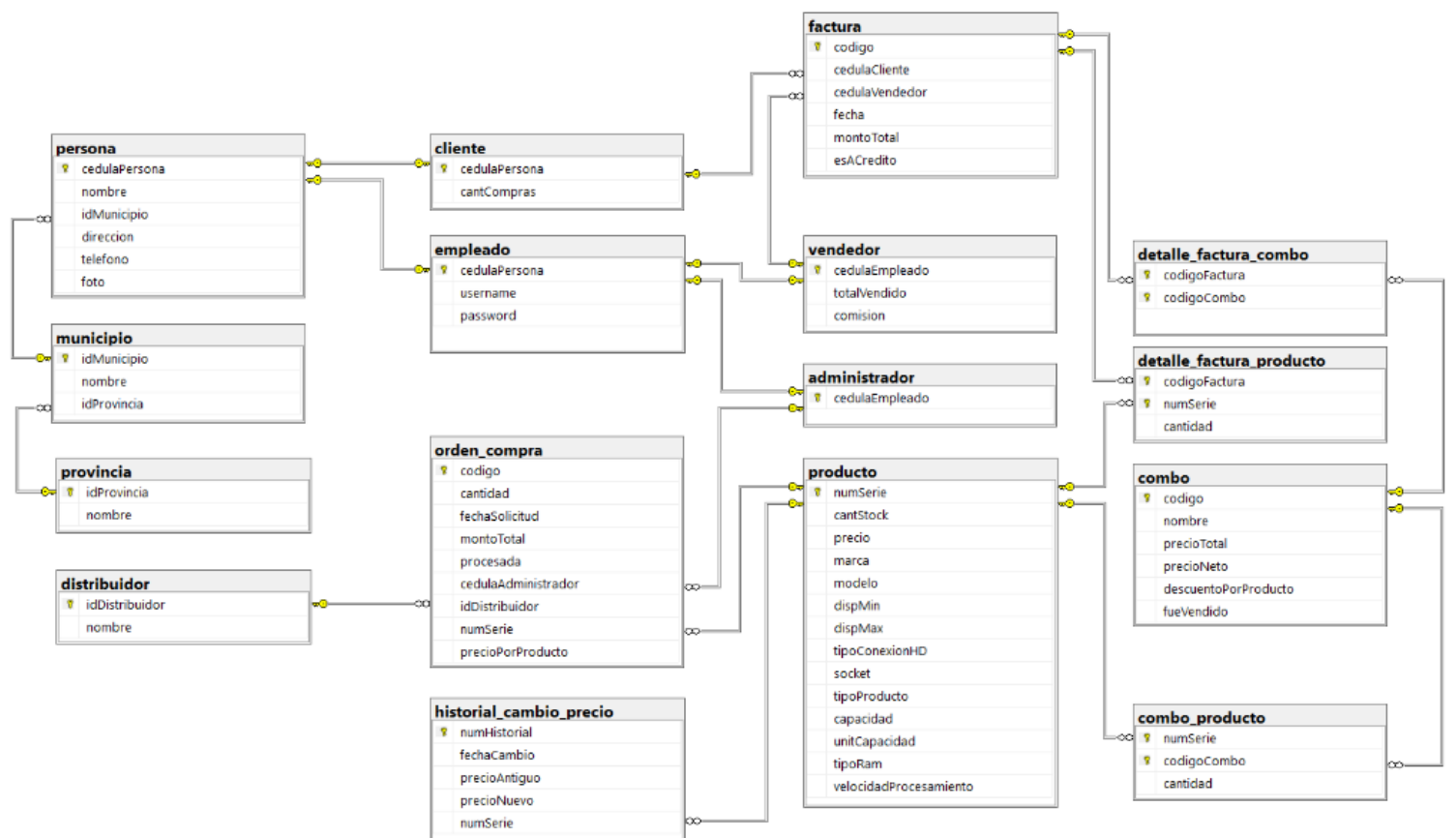
<b>Tabla</b>	Municipio		<b>Fecha</b>	21/7/2022
<b>Descripción</b>	Tabla que contiene las informaciones de los municipios del país.			
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Tamaño</b>	<b>Default</b>	<b>Descripción</b>
idMunicipio	INT	5		Identificador único de un municipio
nombre	Varchar	50		Nombre del municipio
idProvincia	INT	5		Identificador único para una provincia
<b>Relaciones</b>			<b>Claves</b>	
Relación con la tabla Provincia			idProvincia (PK)	

Como ya fue explicado anteriormente, este modelo relacional cuenta con 16 tablas relacionales entre cada una de las entidades, de las cuales tres de ellas son resultantes de las relaciones mucho a mucho existentes en nuestra base de datos.

De forma general, se puede resaltar el uso del tipo de dato entero (INT) para la mayoría de las claves primarias de las tablas, con excepción, evidentemente, de aquellas tablas relacionadas con Persona, pues las mismas utilizan la cédula (varchar) como clave primaria.

Otro aspecto considerable a resaltar fue la decisión de generar dos tablas resultantes para componer el detalle de cada factura. Esta decisión de diseño fue tomada con el objetivo de permitir que las facturas puedan ser registradas con libertad teniendo ya sea un producto, un combo, o la combinación de ambas. De haber implementado un sólo detalle de factura, esto no habría sido posible debido a la clave primaria resultante en la misma, la cual obligaría a contener un Código de factura, un código de combo y un número de serie de producto de forma simultánea.

Para obtener una visión más detallada de este modelo, y de la forma en la que las claves se relacionan entre cada una de las entidades, a continuación, se presenta el diagrama generado por nuestra base de datos en Microsoft SQL Server Management Studio 2019:



# SENTENCIAS SQL

A lo largo del proceso de diseño, desarrollo, creación e implementación de la Base de Datos en este proyecto final, se utilizaron un sinnúmero de sentencias y consultas que sirvieron como base fundamental de la programación de nuestro programa, y que representaron el eje principal de muchas de sus funcionalidades.

El conjunto de estas consultas y sentencias fue realizado y registrado en su totalidad empleando el sistema de Querys en nuestra base de datos *TecnoC*, dentro del servidor con ip 192.168.100.118 proporcionado por nuestra facilitadora y por la Universidad, utilizando el manejador de base de datos Microsoft SQL Server Management Studio 2019.

Para observar la totalidad de cada una de estas sentencias y consultas realizadas durante el desarrollo de este proyecto final, se puede referir al documento adjunto de nombre *TecnoC\_Query*, de tipo .sql. En el mismo, además de las ya mencionadas sentencias, que van desde la creación de la base de datos y sus respectivas tablas empleando el lenguaje de definición de datos (DDL) , hasta la declaración de funciones y procedimientos almacenados avanzados empleando el lenguaje de manipulación de datos (DML); también se encuentra la documentación que explica de forma breve y concisa la funcionalidad de cada una de ellas.

Independientemente de este archivo anexo, a continuación, también se evidencian, de una forma más concisa, las principales sentencias SQL desarrolladas. Junto a ciertas consultas se encuentra su justificación del porqué su existencia.

```

/*****          PROYECTO FINAL ICC-223          *****/
/****   TecnoC: Tienda de Piezas de Computador   ****/

-- Creando la Base de Datos para el proyecto.
CREATE DATABASE TecnoC

-- Seleccionando la Base de Datos para su uso.
USE TecnoC

/*****          Creación de Tablas          *****/

-- Creando la tabla Provincia.
CREATE TABLE provincia(
    idProvincia INT NOT NULL PRIMARY KEY identity(1,1),
    nombre varchar(50) NOT NULL
);

-- Creando la tabla Municipio.
CREATE table municipio(
    idMunicipio INT NOT NULL PRIMARY KEY identity(1,1),
    nombre varchar(50) NOT NULL,
    idProvincia INT NOT NULL
    FOREIGN KEY (idProvincia) REFERENCES Provincia(idProvincia)
);
```



```

-- Creando la tabla Persona.
CREATE table persona(
    cedulaPersona varchar(13) NOT NULL PRIMARY KEY,
    nombre varchar(50) NOT NULL,
    idMunicipio int NOT NULL,
    direccion varchar(50) NOT NULL,
    telefono varchar(12) NOT NULL,
    foto varbinary(MAX)
    FOREIGN KEY (idMunicipio) REFERENCES municipio(idMunicipio)
);

-- Creando la tabla Empleado
CREATE TABLE empleado(
    cedulaPersona varchar(13) PRIMARY KEY,
    username varchar(15) NOT NULL,
    password varbinary NOT NULL,
    FOREIGN KEY (cedulaPersona) REFERENCES persona(cedulaPersona)
);

-- Creando la tabla Administrador.
CREATE TABLE administrador(
    cedulaEmpleado varchar(13) PRIMARY KEY,
    FOREIGN KEY (cedulaEmpleado) REFERENCES empleado(cedulaPersona)
);

-- Creando la tabla Vendedor
CREATE TABLE vendedor(
    cedulaEmpleado varchar(13) PRIMARY KEY,
    totalVendido numeric(12,2) NULL,
    comision numeric(12,2) NULL
    FOREIGN KEY (cedulaEmpleado) REFERENCES empleado(cedulaPersona)
);

-- Creando la tabla Cliente.
CREATE table cliente(
    cedulaPersona varchar(13) NOT NULL PRIMARY KEY,
    cantCompras int,
    FOREIGN KEY (cedulaPersona) REFERENCES persona(cedulaPersona)
);

-- Creando la tabla Distribuidor.
CREATE table distribuidor(
    idDistribuidor INT NOT NULL PRIMARY KEY identity(1,1),
    nombre varchar(50) NOT NULL
);

-- Creando la tabla Producto.
CREATE TABLE producto(
    numSerie varchar(10) NOT NULL PRIMARY KEY,
    cantStock INT NOT NULL,
    precio numeric(11,2) NOT NULL,
    marca varchar(50) NOT NULL,
    modelo varchar(50) NOT NULL,
    dispMin INT NOT NULL,
    dispMax INT NOT NULL,
    tipoConexionHD varchar(5),
    socket varchar(50),
    tipoProducto varchar(50) NOT NULL,
    capacidad INT,
    unitCapacidad varchar(2),
    tipoRam varchar(10),

```

```

        velocidadProcesamiento decimal(7,2)
    );

-- Creando la tabla Combo.
CREATE TABLE combo (
    codigo INT NOT NULL PRIMARY KEY identity(1,1),
    nombre varchar(50) NOT NULL,
    precioTotal numeric(12,2),
    precioNeto numeric(12,2),
    descuentoPorProducto numeric(12,2) NOT NULL,
    fueVendido BIT NOT NULL DEFAULT 0
);

-- Creando la tabla Combo_Producto.
CREATE TABLE combo_producto (
    numSerie varchar(10) NOT NULL,
    codigoCombo INT NOT NULL,
    cantidad INT NOT NULL,
    PRIMARY KEY (numSerie, codigoCombo) ,
    FOREIGN KEY (numSerie) REFERENCES producto(numSerie),
    FOREIGN KEY (codigoCombo) REFERENCES combo(codigo)
);

-- Creando la tabla Factura.
CREATE TABLE factura (
    codigo INT NOT NULL PRIMARY KEY identity(1,1),
    cedulaCliente varchar(13) NOT NULL,
    cedulaVendedor varchar(13) NOT NULL,
    fecha datetime NOT NULL,
    montoTotal numeric(12,2) NOT NULL,
    esACredito BIT NOT NULL DEFAULT 0,
    FOREIGN KEY (cedulaCliente) REFERENCES cliente(cedulaPersona),
    FOREIGN KEY (cedulaVendedor) REFERENCES vendedor(cedulaEmpleado)
);

-- Creando la tabla Detalle_Factura_Producto
CREATE TABLE detalle_factura_producto(
    codigoFactura int NOT NULL,
    numSerie varchar(10) NOT NULL,
    cantidad int NOT NULL,
    PRIMARY KEY(codigoFactura, numSerie),
    FOREIGN KEY (codigoFactura) REFERENCES factura(codigo),
    FOREIGN KEY (numSerie) REFERENCES producto(numSerie),
);

-- Creando la tabla Detalle_Factura_Combo
CREATE TABLE detalle_factura_combo (
    codigoFactura INT NOT NULL,
    codigoCombo INT NOT NULL,
    cantidad INT NOT NULL,
    PRIMARY KEY (codigoFactura, codigoCombo) ,
    FOREIGN KEY (codigoFactura) REFERENCES factura(codigo),
    FOREIGN KEY (codigoCombo) REFERENCES combo(codigo)
);

-- Creando la tabla orden_compra
CREATE TABLE orden_compra (
    codigo INT NOT NULL PRIMARY KEY identity(1,1),
    cantidad INT NULL,
    fechaSolicitud DATETIME NOT NULL,
    montoTotal NUMERIC(12,2) NULL,
    procesada BIT NULL DEFAULT 0,

```

```

        cedulaAdministrador VARCHAR(13) NULL,
        idDistribuidor INT NULL,
        numSerie VARCHAR(10) NOT NULL,
        FOREIGN KEY (cedulaAdministrador) REFERENCES
administrador(cedulaEmpleado),
        FOREIGN KEY (idDistribuidor) REFERENCES distribuidor(idDistribuidor),
        FOREIGN KEY (numSerie) REFERENCES producto(numSerie)
);

-- Creando la tabla historial_cambio_precio
CREATE TABLE historial_cambio_precio (
    numHistorial INT NOT NULL PRIMARY KEY identity(1,1),
    fechaCambio date NOT NULL,
    precioAntiguo numeric(11,2) NOT NULL,
    precioNuevo numeric(11,2) NOT NULL,
    numSerie varchar(10) NOT NULL FOREIGN KEY REFERENCES producto(numSerie)
);

```

/\*\*\*\*\* Creación de Procedimientos Almacenados \*\*\*\*\*/

--Procedimiento encargado de sumar el stock de un articulo dado. Dicho procedimiento fue creado con la finalidad de ser llamado por algunos triggers. Facilitando asi las consultas.

```

CREATE PROCEDURE suma_stock
    @numSerie varchar(10), @cantidad int
AS
UPDATE producto
set producto.cantStock = producto.cantStock + @cantidad from producto
WHERE producto.numSerie = @numSerie;

```

--Procedimiento encargado de restar el stock de un articulo dado. Dicho procedimiento fue creado con la finalidad de ser llamado por unos trigger. Facilitando asi las consultas.

```

CREATE PROCEDURE resta_stock
    @numSerie varchar(10), @cantidad int
AS
UPDATE producto
set producto.cantStock = producto.cantStock - @cantidad from producto
WHERE producto.numSerie = @numSerie

```

--Procedimiento encargado de restar el stock de los productos contenidos en un combo. Dicho procedimiento fue creado con la finalidad de ser llamado por unos trigger. Facilitando asi la creacion de los mismos.

```

CREATE PROCEDURE resta_stock_combo
    @codCombo int, @cantidad int
AS
UPDATE producto
SET producto.cantStock = producto.cantStock - (1 * @cantidad)
from producto
INNER JOIN combo_producto
ON combo_producto.codigoCombo = @codCombo
WHERE combo_producto.numSerie = producto.numSerie;

```

--Procedimiento encargado de crear, en primer lugar, una persona y posteriormente un cliente. Dicho procedimiento fue creado con la finalidad de que desde el backend se llame a este Procedure y este Procedure se encargue de crear la entidad fuerte primero y luego la debil.

```
CREATE PROCEDURE crea_persona_y_cliente
    @cedula varchar(13), @nombre varchar(50), @idMunicipio int, @direccion
    varchar(50),
    @telefono varchar(12), @cantCompras int
AS
BEGIN
    DECLARE @valida bit, @return int
    SELECT @valida = dbo.valida_cedula(@cedula)
    IF @valida = 1
        BEGIN
            BEGIN TRANSACTION
            INSERT INTO persona values(@cedula, @nombre, @idMunicipio,
@direccion, @telefono, null)
            IF @@error <> 0
                BEGIN
                    SELECT @return = -2
                    ROLLBACK TRANSACTION
                END
            ELSE
                BEGIN
                    BEGIN TRANSACTION
                    INSERT INTO cliente values(@cedula, @cantCompras)
                    IF @@error <> 0
                        BEGIN
                            SELECT @return = -3
                            ROLLBACK TRANSACTION
                        END
                    ELSE
                        BEGIN
                            COMMIT TRANSACTION
                        END
                    COMMIT TRANSACTION
                    SELECT @return = 1
                END
            END
        END
    ELSE
        BEGIN
            SELECT @return = -1
        END
    RETURN @return
END
```

--Procedimiento encargado de verificar si es posible generar una orden de compra para un producto dado, y crearla de ser posible. Este procedure es llamado a la hora de que un producto llegue a su disponibilidad minima. Verifica si es posible crear la orden de compra. De ser asi, la crea.

```
CREATE PROCEDURE PCD_crear_orden_compra
    @numSerie varchar(10)
AS
DECLARE @esPosible bit
SELECT @esPosible = dbo.es_posible_crear_orden_compra(@numSerie)

IF @esPosible = 1
    BEGIN
        INSERT into orden_compra (fechaSolicitud, procesada, numSerie)
        values(getdate(), 0, @numSerie);
    END
```

/\*\*\*\*\* Creación de Triggers/Disparadores \*\*\*\*\*/

--Trigger para la tabla orden\_compra que suma el stock del producto al que se le confirmó la orden de compra. Este trigger es disparado cuando la orden de compra es confirmada por el Administrador del sistema.

```
CREATE TRIGGER TR_suma_stock_orden_compra
on orden_compra for UPDATE
AS
DECLARE @numSerie_func varchar(10), @cantidad_func int, @procesada_func bit
SELECT @numSerie_func = numSerie, @cantidad_func = cantidad, @procesada_func =
procesada from inserted
IF @procesada_func = 1
BEGIN
EXEC suma_stock @numSerie = @numSerie_func, @cantidad =
@cantidad_func;
END
```

--Trigger para la tabla detalle\_factura\_combo que resta el stock de todos los productos del combo insertados en dicha tabla. Este se encarga de facilitar el proceso de restar del inventario el producto del combo vendido.

```
CREATE TRIGGER TR_resta_stock_detalle_factura_combo
on detalle_factura_combo for INSERT
AS
DECLARE @codCombo_func int, @cantidad_func int
SELECT @codCombo_func = inserted.codigoCombo, @cantidad_func = inserted.cantidad
from inserted
EXEC resta_stock_combo @codCombo = @codCombo_func, @cantidad = @cantidad_func
```

--Trigger para la tabla detalle\_factura\_producto que resta el stock del producto insertado en dicha tabla. Este se encarga de facilitar el proceso de restar del inventario el producto vendido.

```
CREATE TRIGGER TR_resta_stock_detalle_factura_producto
on detalle_factura_producto for INSERT
AS
DECLARE @numSerie_func varchar(10), @cantidad_func int
SELECT @numSerie_func = numSerie, @cantidad_func = cantidad from inserted
EXEC resta_stock @numSerie = @numSerie_func, @cantidad = @cantidad_func;
```

--Trigger para la tabla producto que verifica la disponibilidad minima de un producto y ejecuta el procedure que se encarga de verificar si existe una orden de compra o no. De no existir, crea una orden de compra  
-- dado el caso en que el stock sea menor o igual a dispMin.

```
CREATE TRIGGER TR_crear_orden_compra
ON producto for UPDATE
AS
DECLARE @numSerie_local varchar(10), @cant int, @dispMin int
SELECT @numSerie_local = i.numSerie, @cant = i.cantStock, @dispMin = i.dispMin
from inserted i
IF @cant <= @dispMin
BEGIN
EXEC PCD_crear_orden_compra @numSerie = @numSerie_local
END
```

--Trigger para la tabla orden\_compra que calcula el monto total de una orden de compra. Este trigger facilita el calculo de de una orden de compra, ya que cuando se hace el UPDATE en la tabla inserted se tienen todos los datos necesarios.

```
CREATE TRIGGER calcula_monto_total_pedido
ON orden_compra FOR UPDATE
AS
DECLARE @precio decimal(10,2), @cantidad int, @codigo int, @procesada int
SELECT @codigo = i.codigo, @precio = i.precioPorProducto, @cantidad = i.cantidad,
@procesada = i.procesada
FROM inserted i
IF @procesada = 1
BEGIN
    UPDATE orden_compra
    SET montoTotal = @precio * @cantidad
    WHERE orden_compra.codigo = @codigo
END
```

--Trigger para la tabla combo\_producto que va sumando el total de un combo mientras se le inserten productos. Este trigger mantiene actualizado el precio del combo, ya que mientras se vayan agregando productos al combo, el mismo se va actualizando.

```
CREATE TRIGGER suma_total_combo
ON combo_producto FOR INSERT
AS
DECLARE @codigoCombo INT, @numSerie VARCHAR(10), @cantidad INT, @precio decimal(11,2)
SELECT @codigoCombo = i.codigoCombo, @cantidad = i.cantidad, @numSerie = i.numSerie,
@precio = (SELECT precio from producto where numSerie = @numSerie)
FROM inserted i
UPDATE combo SET precioNeto = precioNeto + (@precio * @cantidad)
FROM combo
INNER JOIN combo_producto cp on cp.numSerie = @numSerie
WHERE combo.codigo = @codigoCombo;
```

--Trigger para la tabla producto que guarda el historial de cambio de precio de un producto. Este trigger es creado de manera que se pueda saber que ha pasado con el precio de un producto cuando el mismo sea cambiado.

```
CREATE TRIGGER cambio_historial_precio
ON producto FOR UPDATE
AS
DECLARE @numSerie VARCHAR(10), @precioAntiguo decimal(11,2), @precioNuevo decimal(11,2)
SELECT @numSerie = i.numSerie, @precioAntiguo = d.precio, @precioNuevo = i.precio
FROM inserted i, deleted d
INSERT INTO historial_cambio_precio values(GETDATE(), @precioAntiguo,
@precioNuevo, @numSerie);
```

--Trigger para la tabla combo\_producto que resta del stock los productos que se agregan a un combo. Este trigger se ejecuta para que cuando un producto sea insertado en un combo, el mismo sea retirado del inventario. Este no lo ejecutamos en la version final pues consideramos que era preferible para el dueño de la tienda vender el producto individualmente que con el combo.

```
CREATE TRIGGER resta_stock_combo_producto
ON combo_producto FOR INSERT
AS
DECLARE @numSerie varchar(10), @cantidad INT
SELECT @numSerie = inserted.numSerie, @cantidad = inserted.cantidad
FROM inserted
UPDATE producto SET producto.cantStock = producto.cantStock - @cantidad
WHERE producto.numSerie = @numSerie;
```

/\*\*\*\*\* Creación de Funciones \*\*\*\*\*/

--Función encargada de validar una cedula dominicana dada. Esta funcion se llama a la hora de insertar una persona, con el fin de verificar que la cedula insersitada por el usuario no sea una cedula falsa.

```
CREATE FUNCTION valida_cedula (@cedula varchar(20))
RETURNS bit
as
BEGIN
    DECLARE @len_cedula int, @ind1 int, @ind2 int, @return bit
    SELECT @len_cedula = len(@cedula);
    SELECT @ind1 = CHARINDEX('-', @cedula, 0);
    SELECT @ind2 = CHARINDEX('-', @cedula, 5);
    IF @len_cedula = 13 AND @ind1 = 4 AND @ind2 = 12
        SELECT @return = 1
    ELSE
        SELECT @return = 0

    RETURN @return
END
```

--Funcion encargada de validar si ya hay creada una orden de compra para un producto dado. Esta funcion verifica que no haya una orden de compra ya creada y sin procesar de un producto dado.

```
CREATE FUNCTION es_posible_crear_orden_compra (@numSerie varchar(10))
RETURNS bit
AS
BEGIN
    DECLARE @posible bit
    SELECT @posible =
        (SELECT IIF(EXISTS (SELECT oc.procesada
            from producto
            right join orden_compra oc
            on producto.numSerie = oc.numSerie
            where oc.numSerie LIKE @numSerie AND
            oc.procesada = 0),
            0, 1) )

    RETURN @posible
END
```

/\*\*\*\*\* Creación de Vistas \*\*\*\*\*/

--Vista que muestra todos los datos de un cliente, junto con sus datos pertenecientes a persona. Esta view facilita la consulta en backend para cuando se necesite la informacion de una persona (nombre, direccion, telefono, municipio) dado un cliente.

```
CREATE VIEW datos_persona_cliente
AS
SELECT p.cedulaPersona, p.nombre, p.direccion, p.telefono, c.cantCompras,
(SELECT nombre FROM municipio m WHERE m.idMunicipio = p.idMunicipio) AS
nombreMunicipio
FROM cliente c
INNER JOIN persona p
ON p.cedulaPersona = c.cedulaPersona ;
```

--Vista que muestra los datos de un empleado, junto con sus datos pertenecientes a persona. Esta view facilita la consulta en backend para cuando se necesite la informacion de una persona (nombre, direccion, telefono, municipio) dado un empleado.

```
CREATE VIEW datos_persona_empleado
```

```
AS
```

```
SELECT p.cedulaPersona, p.nombre, p.direccion, p.telefono, e.username,  
e.password, p.foto,  
(SELECT nombre FROM municipio m WHERE m.idMunicipio = p.idMunicipio) AS  
nombreMunicipio  
FROM empleado e  
INNER JOIN persona p  
ON p.cedulaPersona = e.cedulaPersona ;
```

--Vista que muestra el Cliente del mes, es decir, el Cliente con mayor monto de consumo en el mes actual. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW cliente_del_mes
```

```
AS
```

```
SELECT MAX(totalComprado) AS maximoComprado, cedulaPersona  
FROM(  
    SELECT SUM(montoTotal) AS totalComprado, cedulaPersona  
    FROM cliente c  
    INNER JOIN factura f  
    ON c.cedulaPersona = f.cedulaCliente  
    WHERE MONTH(GETDATE()) = MONTH(f.fecha) AND YEAR(GETDATE()) =  
    YEAR(f.fecha)  
    GROUP BY cedulaPersona  
    ) AS maxComprado  
GROUP BY cedulaPersona;
```

--Vista que muestra el cliente con mas compras realizadas en la Tienda. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW cliente_con_mas_compras
```

```
AS
```

```
SELECT cedulaPersona, cantCompras  
FROM cliente  
WHERE cantCompras = (SELECT MAX(cantCompras) AS maxCantCompra FROM cliente);
```

--Vista que muestra el cliente con menos compras realizadas en la Tienda. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW cliente_con_menos_compras
```

```
AS
```

```
SELECT cedulaPersona, cantCompras  
FROM cliente  
WHERE cantCompras = (SELECT MIN(cantCompras) AS maxCantCompra FROM cliente);
```

--Vista que muestra el producto mas comprado por los Clientes. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW producto_mas_comprado
```

```
AS
```

```
SELECT TOP(1) MAX(totalVendido) AS maxVendido, numSerie  
FROM(  
    SELECT SUM(cantidad) AS totalVendido, p.numSerie  
    FROM detalle_factura_producto dfp  
    INNER JOIN producto p
```



```

        ON dfp.numSerie = p.numSerie
        GROUP BY p.numSerie
    ) AS maximo
GROUP BY numSerie
ORDER BY MAX(totalVendido) DESC;

```

--Vista que muestra el producto menos comprado por los Clientes. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habría que realizarle un select y se tendría en tiempo real el resultado.

```

CREATE VIEW producto_menos_comprado
AS
SELECT TOP(1) MIN(totalVendido) AS minimoVendido, numSerie
FROM(
    SELECT SUM(cantidad) AS totalVendido, p.numSerie
    FROM detalle_factura_producto dfp
    RIGHT JOIN producto p
    ON dfp.numSerie = p.numSerie
    GROUP BY p.numSerie
) AS minimo
GROUP BY numSerie
ORDER BY MIN(totalVendido) ASC;

```

--Vista que muestra el vendedor con más ventas realizadas. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habría que realizarle un select y se tendría en tiempo real el resultado.

```

CREATE VIEW vendedor_con_mas_compras
AS
SELECT TOP(1) MAX(totalVendido) AS masCompras, cedulaEmpleado
FROM(
    SELECT COUNT(codigo) AS totalVendido, cedulaEmpleado
    FROM vendedor v
    INNER JOIN factura f
    ON v.cedulaEmpleado = f.cedulaVendedor
    GROUP BY cedulaEmpleado
) AS maxVendido
GROUP BY cedulaEmpleado, totalVendido
ORDER BY MAX(totalVendido) DESC;

```

--Vista que muestra el vendedor con menos ventas realizadas. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habría que realizarle un select y se tendría en tiempo real el resultado.

```

CREATE VIEW vendedor_con_menos_compras
AS
SELECT TOP(1) MIN(totalVendido) AS menosCompras, cedulaEmpleado
FROM(
    SELECT COUNT(codigo) AS totalVendido, cedulaEmpleado
    FROM vendedor v
    RIGHT JOIN factura f
    ON v.cedulaEmpleado = f.cedulaVendedor
    GROUP BY cedulaEmpleado
) AS minVendido
GROUP BY cedulaEmpleado, totalVendido
ORDER BY MIN(totalVendido) ASC;

```

--Vista que muestra el vendedor del mes, es decir, el vendedor cuyas ventas generaron el mayor ingreso. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habría que realizarle un select y se tendría en tiempo real el resultado.

```

CREATE VIEW vendedor_del_mes
AS
SELECT MAX(totalVendido) AS maximoVendido, cedulaEmpleado

```

```

FROM(
    SELECT SUM(montoTotal) AS totalVendido, cedulaEmpleado
    FROM vendedor v
    INNER JOIN factura f
    ON v.cedulaEmpleado = f.cedulaVendedor
    WHERE MONTH(GETDATE()) = MONTH(f.fecha) AND YEAR(GETDATE()) =
YEAR(f.fecha)
    GROUP BY cedulaEmpleado
    ) AS maxVendido
GROUP BY cedulaEmpleado;

```

--Vista que muestra la factura más cara o con el monto mas alto. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```

CREATE VIEW factura_mas_cara
AS
SELECT montoTotal, codigo
FROM factura f
WHERE f.montoTotal = (SELECT MAX(montoTotal) FROM factura);

```

--Vista que muestra la factura menos cara que ha sido registrada. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```

CREATE VIEW factura_menos_cara
AS
SELECT montoTotal, codigo
FROM factura f
WHERE f.montoTotal = (SELECT MIN(montoTotal) FROM factura);

```

--Vista que muestra la cantidad de facturas a credito registradas en el Sistema. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```

CREATE VIEW facturas_adeudadas
AS
SELECT COUNT(codigo) AS facturasACredito
FROM factura
WHERE factura.esACredito = 1;

```

--Vista que muestra la cantidad de facturas corrientes (sin credito). Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```

CREATE VIEW facturas_pagadas
AS
SELECT COUNT(codigo) AS facturasACredito
FROM factura
WHERE factura.esACredito = 0;

```

--Vista que muestra el total vendido de todas factura registradas en el Sistema. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```

CREATE VIEW total_vendido
AS
SELECT SUM(montoTotal) AS total
FROM factura;

```

--Vista que muestra el combo mas comprado por los Clientes. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW combo_mas_comprado
AS
SELECT TOP(1) MAX(totalVendido) AS maxVendido, codigo, nombre
FROM(
    SELECT SUM(c.codigo) AS totalVendido, codigo, nombre
    FROM detalle_factura_combo dfc
    INNER JOIN combo c
    ON dfc.codigoCombo = c.codigo
    GROUP BY c.codigo, c.nombre
) AS maximo
GROUP BY codigo, nombre
ORDER BY MAX(totalVendido) DESC;
```

--Vista que muestra el combo menos comprado por los Clientes. Este view facilita mucho a la hora de presentar las estadísticas de la Tienda, ya que solamente habria que realizarle un select y se tendria en tiempo real el resultado.

```
CREATE VIEW combo_menos_comprado
AS
SELECT TOP(1) MIN(totalVendido) AS maxVendido, codigo, nombre
FROM(
    SELECT SUM(c.codigo) AS totalVendido, codigo, nombre
    FROM detalle_factura_combo dfc
    INNER JOIN combo c
    ON dfc.codigoCombo = c.codigo
    GROUP BY c.codigo, c.nombre
) AS maximo
GROUP BY codigo, nombre
ORDER BY MIN(totalVendido) DESC;
```

--Vista que muestra los productos vendidos en un combo. Este view facilita a la hora de presentar los detalles de una factura de un combo. Pues este devuelve las informaciones necesarias para mostrar en la ventana.

```
CREATE VIEW productos_vendido_combo
AS
SELECT codigoFactura, dfc.codigoCombo, cp.cantidad, numSerie,
descuentoPorProducto, precioTotal
FROM detalle_factura_combo dfc
INNER JOIN combo_producto cp ON dfc.codigoCombo = cp.codigoCombo
INNER JOIN combo c ON c.codigo =dfc.codigoCombo ;
```

# CONCLUSIONES

Sin lugar a duda, este no fue un trabajo sencillo, es bien sabido que actualizar un sistema/aplicación a una nueva tecnología es un trabajo arduo que requiere de bastantes horas. Durante el proceso de actualización de sistema de ficheros a una base de datos remota nos vimos envueltos en muchos problemas los cuáles tuvimos que solucionar uno a uno, siendo el primero el gran caos que existía en el código antiguo, las cosas no estaban organizadas, había mucho código que no servía para nada, cosas repetidas, funciones que no se podían entender, en resumen, un desastre. Pero, nos pusimos manos a la obra y logramos organizar y maquetar de una forma más entendible el programa.

A la hora de diseñar la base de datos se nos complicó un poco la parte de las relaciones puesto que en el sistema al manejarse con arraylist no se necesitaba tantas complicaciones, pues ahora teníamos que buscar la forma de que las relaciones no sean incoherentes, especialmente en la sección que tiene que ver con las facturas y sus detalles respecto al producto y combo, luego de pensarlo por un tiempo y consultarlo con la maestra la mejor idea fue modularizar esa parte mediante la creación de dos tablas detalles, una del lado del producto y otra del lado del combo.

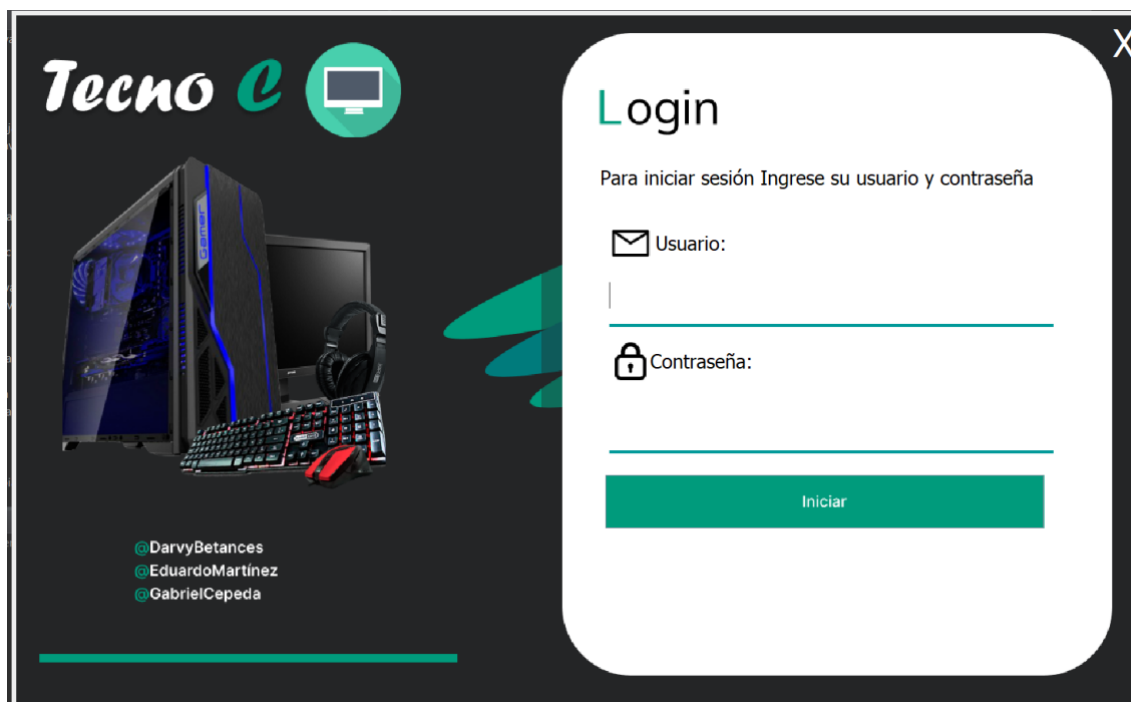
No teníamos idea del potencial que tiene una base de datos cuándo se integran en un sistema, muchas cosas se facilitaron, no todo fue problemas, un ejemplo de esto es el apartado de las estadísticas donde había mucho código problemático en java, esa parte se solucionó simplemente haciendo consultas simples a la base de datos (Como el cliente con mayor cantidad de compras, por ejemplo). Tres meses atrás hubiera sido imposible para nosotros creer que todo eso era posible con un lenguaje que creíamos que estaba pensado solo para crear tablas.

Otra de las facilidades que conseguimos al integrar nuestra base de datos al sistema fue el manejo de fotos, es increíble lo fácil que es manejar archivos de imágenes en SQL, esa sección que fue tan difícil de elaborar con java se solucionó SQL y su tipo de dato VARBINARY que nos permite guardar la imagen y posteriormente con BLOB recuperarla en el sistema. Aunque no se puede decir lo mismo del apartado de inicio de sesión, fue una tarea bastante complicada por la misma razón que se facilitan las imágenes, la contraseña de los usuarios para proteger su seguridad se guardaba en Bytes dentro de la base de datos lo que dificultaba su comparación posteriormente cuándo se vaya a comprobar si el usuario existe dentro de la base de datos, hubo que descomponer estos bytes para poder compararlo posteriormente con el dato que envíe el usuario.

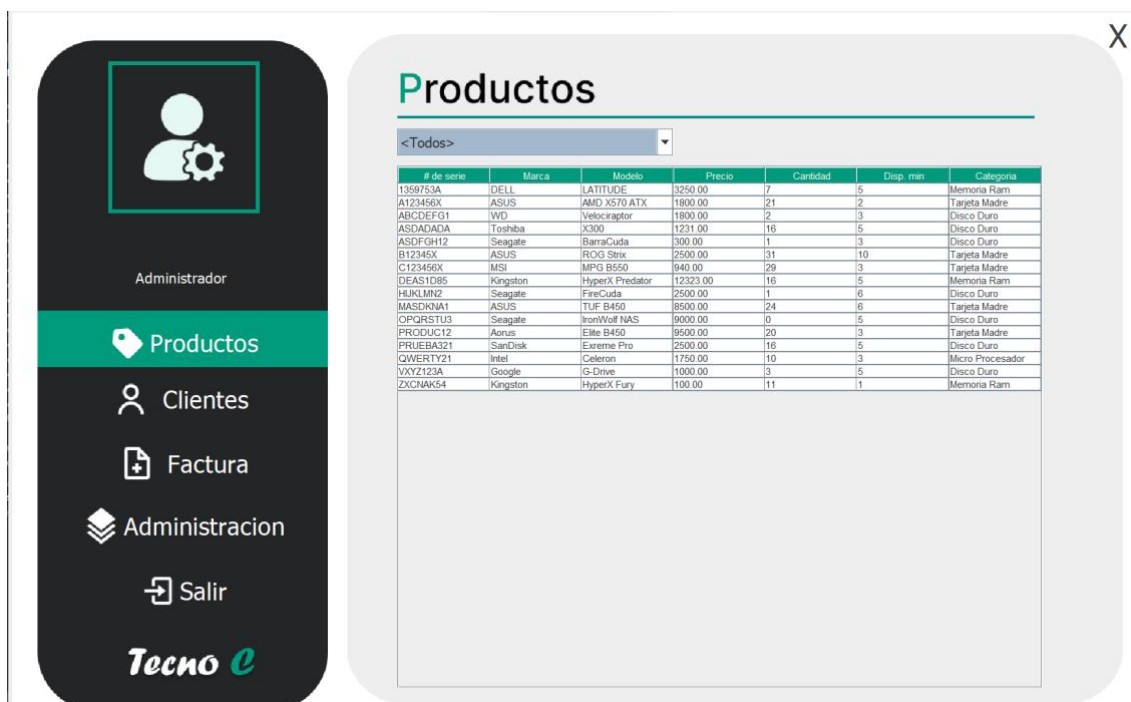
Así sucesivamente fueron apareciendo muchos errores, problemas de diseños (La base de datos y los diagramas cambiaron bastante a medida que se iba integrando la base de datos al sistema). Pero todas las horas de trabajo, los días de trasnoches, los consejos de la maestra y el gran trabajo en equipo permitieron que el proyecto saliera adelante.

# ANEXOS

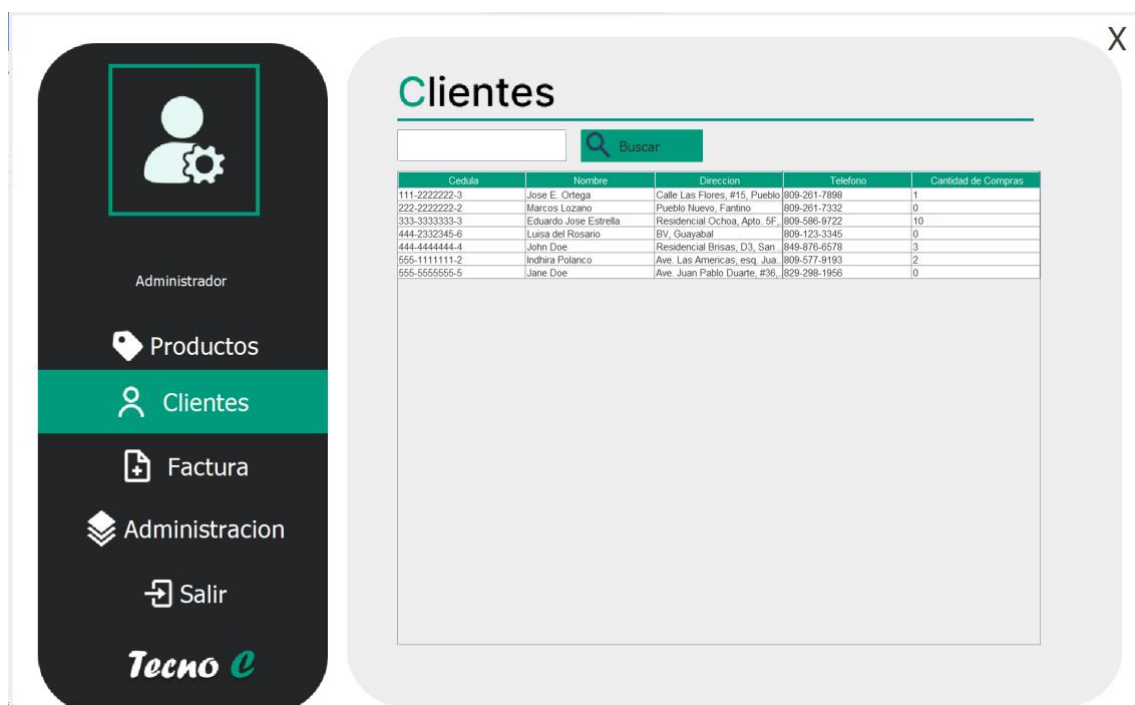
Aquí se presentarán algunas imágenes de las funcionalidades del programa:



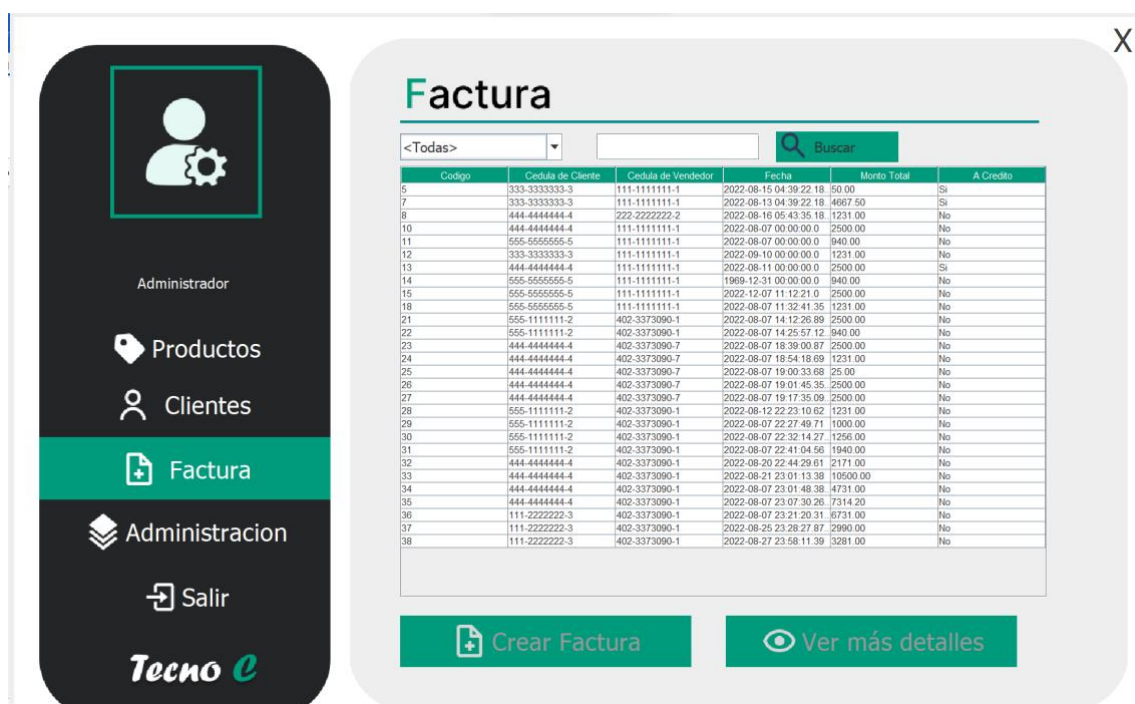
Pantalla de Inicio de sesión.



Pantalla del Home para el administrador con vista rápida al inventario de productos.



Vista rápida desde el home al listado de clientes



Vista rápida a las facturas para el administrador.

Tecno

C

PDF

FACTURA

Facturar a

John Doe  
849-876-6578  
Residencial Brisas, D3, San Fra...

Factura#

13

Fecha

11-08-2022

Vendedor

Yuberkys Ochoa

Articulos

Num. de Serie	Modelo	Tipo de Prod...	Cantidad	Precio	Total
B12345X	ASUS ROG ...	Tarjeta Madre	1	2500.0	2500.0

Cantidad Articulos

1

Precio total

\$2500.0

Detalles de una factura con la opción de imprimir la misma.

Crear Factura:

X

Datos del Cliente:

Cédula:

Buscar

Nombre:

Telefono:

Provincia:

1-Azua

Direccion:

Municipio

2-Azua de Compostela

Registrar

Datos de la Factura:

Código:

39

Precio Total:

RD\$ 0.0

Factura a Credito

Agregar Productos

Agregar Combos

Productos disponibles:

DELL LATITUDE (1359753A | Memoria Ram)  
ASUS AMD X570 ATX (A123456X | Tarjeta Mad  
WD Velociraptor (ABCDEFG1 | Disco Duro)  
Toshiba X300 (ASDADADA | Disco Duro)  
Seagate BarraCuda (ASDFGH12 | Disco Duro)  
ASUS ROG Strix (B12345X | Tarjeta Madre)  
MSI MPG B550 (C123456X | Tarjeta Madre)

Seleccionados:

↑↑

↓↓

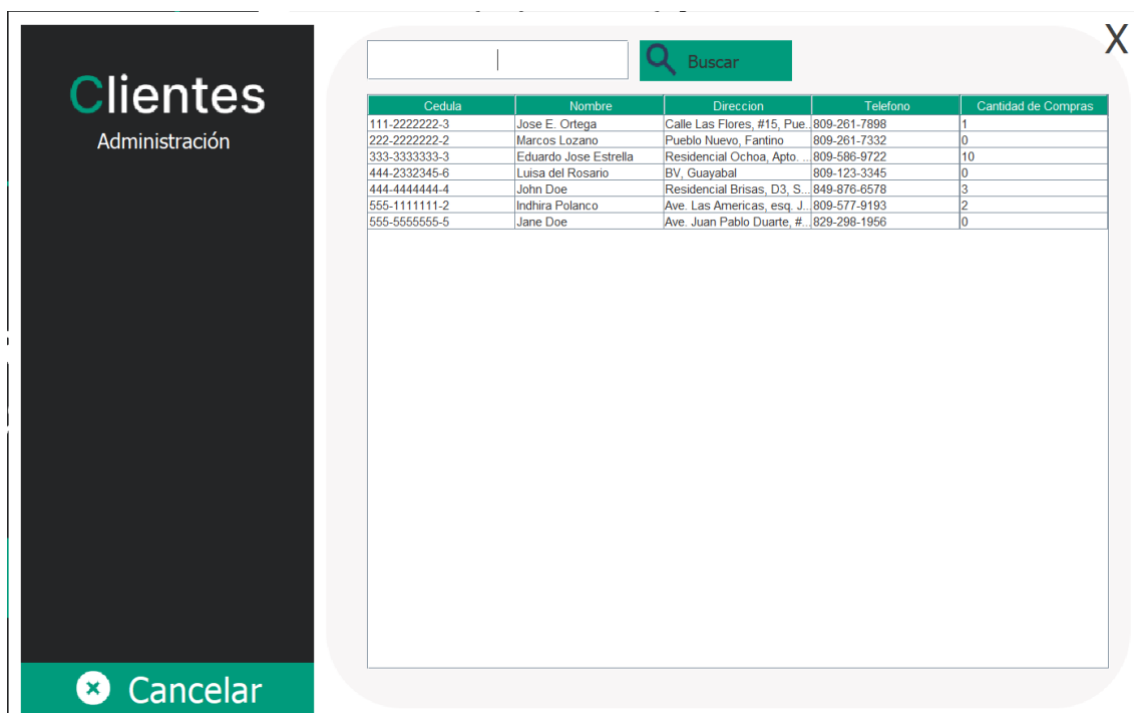
Cancelar

Registrar

Ventana para crear una factura (Solo para usuarios de tipo empleado-vendedor).



Ventana de Administración (Solo disponible para usuarios de tipo empleado-Administrador).



Administración de Clientes.



# Productos

Administración

+ Nuevo

✎ Modificar

✕ Cancelar

Todos

# de serie	Marca	Modelo	Precio	Cantidad	Disp. min	Categoría
1359753A	DELL	LATITUDE	3250.00	15	5	Memoria Ram
A123456X	ASUS	AMD X570 ATX	1800.00	21	2	Tarjeta Madre
ABCDDEFG1	WD	Velociraptor	1800.00	2	3	Disco Duro
ASDADADA	Toshiba	X300	1231.00	16	5	Disco Duro
ASDFGH12	Seagate	BarraCuda	300.00	1	3	Disco Duro
B12345X	ASUS	ROG Strix	2500.00	31	10	Tarjeta Madre
C123456X	MSI	MPG B550	940.00	29	3	Tarjeta Madre
DEAS1D85	Kingston	HyperX Predator	12323.00	16	5	Memoria Ram
HUKLMN2	Seagate	FireCuda	2500.00	1	6	Disco Duro
MASDKNA1	ASUS	TUF B450	8500.00	24	6	Tarjeta Madre
OPQRSTU3	Seagate	IronWolf NAS	9000.00	0	5	Disco Duro
PRODUC12	Aorus	Elite B450	9500.00	20	3	Tarjeta Madre
PRUEBA321	SanDisk	Extreme Pro	2500.00	16	5	Disco Duro
QWERTY21	Intel	Celeron	1750.00	10	3	Micro Procesador
VXYZ123A	Google	G-Drive	1000.00	3	5	Disco Duro
ZXCNAK54	Kingston	HyperX Fury	100.00	11	1	Memoria Ram

Administración de productos.

## Registrar Producto:

Num. De serie:

Marca:

Precio:

Cantidad:  0

Disp. Mínima:  0      Disp. Máxima:  0

☒ Disco Duro  
 ☐ Memoria Ram  
 ☐ Tarjeta Madre  
 ☐ Micro Procesa...

Modelo:

Capacidad:  GB

Tipo de Socket: <Seleccione>

Registrar y/o modificar productos.

Combos

Administración

+

 Crear

Modificar

✕

 Cancelar

Código	Nombre	Precio Total	Precio Neto	Descuento por Producto
1	Gaming Starter Pack	0.00	0.00	15.00
2	UpToRAM	6892.44	4962.20	80.00
3	Semana Santa 2K22	39623.00	19811.50	50.00
4	Navidad 2021	1300.00	1040.00	20.00
5	Liquidacion Discos Duros	1825.00	1642.50	10.00
6	Black Friday 2022 (A)	3440.00	3268.00	5.00
16	Black Friday 2022 (B)	0.00	0.00	0.00
17	Black Friday 2022 (C)	940.00	808.40	14.00
18	Cyber Monday 2022	5940.00	5049.00	15.00
19	TheBestia	11400.00	10260.00	10.00

Administración de Combos.

Registrar Combo:

Código:

20

Nombre:

Descuento:

0

%

Disponibles:

# Serie	Categoría	Marca
1359753A	MemoriaR...	DELL
A123456X	MotherBoard	ASUS
ABCDEF61	DiscoDuro	WD
ASDADADA	DiscoDuro	Toshiba
ASDFGH12	DiscoDuro	Seagate
B12345X	MotherBoard	ASUS
C123456X	MotherBoard	MSI
DEAS1D85	MemoriaR...	Kingston
HJKLMN2	DiscoDuro	Seagate
MASDKNA1	MotherBoard	ASUS
PRODUC12	MotherBoard	Aorus
PRUEBA321	DiscoDuro	SanDisk
QWERTY21	MicroProce	Intel

>>

<<

Elegidos:

# Serie	Categoría	Marca
---------	-----------	-------

Precio:

0

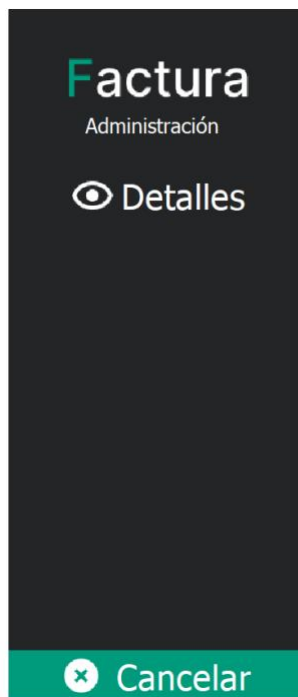
Precio neto:

0

Cancelar

Registrar

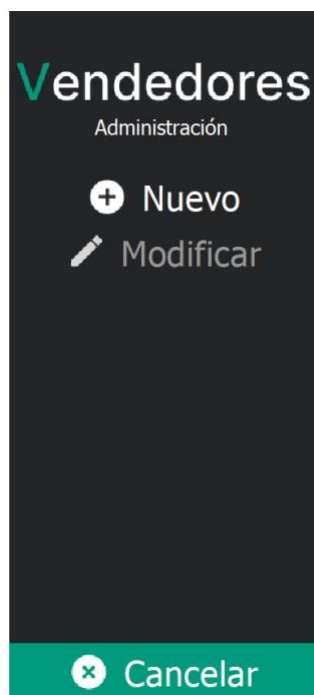
Registrar y/o modificar un combo.



<Todas>

Codigo	Cedula de Cliente	Cedula de Vendedor	Fecha	Monto Total	A Credito
5	333-333333-3	111-111111-1	2022-08-15 04:39:22	50.00	Si
7	333-333333-3	111-111111-1	2022-08-13 04:39:22	4667.50	Si
8	444-444444-4	222-222222-2	2022-08-16 05:43:35	1231.00	No
10	444-444444-4	111-111111-1	2022-08-07 00:00:00	2500.00	No
11	555-555555-5	111-111111-1	2022-08-07 00:00:00	940.00	No
12	333-333333-3	111-111111-1	2022-09-10 00:00:00	1231.00	No
13	444-444444-4	111-111111-1	2022-08-11 00:00:00	2500.00	Si
14	555-555555-5	111-111111-1	1969-12-31 00:00:00	940.00	No
15	555-555555-5	111-111111-1	2022-12-07 11:12:21	2500.00	No
18	555-555555-5	111-111111-1	2022-08-07 11:32:41	1231.00	No
21	555-111111-2	402-3373090-1	2022-08-07 14:12:26	2500.00	No
22	555-111111-2	402-3373090-1	2022-08-07 14:25:57	940.00	No
23	444-444444-4	402-3373090-7	2022-08-07 18:39:00	2500.00	No
24	444-444444-4	402-3373090-7	2022-08-07 18:54:18	1231.00	No
25	444-444444-4	402-3373090-7	2022-08-07 19:00:33	25.00	No
26	444-444444-4	402-3373090-7	2022-08-07 19:01:45	2500.00	No
27	444-444444-4	402-3373090-7	2022-08-07 19:17:35	2500.00	No
28	555-111111-2	402-3373090-1	2022-08-12 22:23:10	1231.00	No
29	555-111111-2	402-3373090-1	2022-08-07 22:27:49	1000.00	No
30	555-111111-2	402-3373090-1	2022-08-07 22:32:14	1256.00	No
31	555-111111-2	402-3373090-1	2022-08-07 22:41:04	1940.00	No
32	444-444444-4	402-3373090-1	2022-08-20 22:44:29	2171.00	No
33	444-444444-4	402-3373090-1	2022-08-21 23:01:13	10500.00	No
34	444-444444-4	402-3373090-1	2022-08-07 23:01:48	4731.00	No
35	444-444444-4	402-3373090-1	2022-08-07 23:07:30	7314.20	No
36	111-222222-3	402-3373090-1	2022-08-07 23:21:20	6731.00	No
37	111-222222-3	402-3373090-1	2022-08-25 23:28:27	2990.00	No
38	111-222222-3	402-3373090-1	2022-08-27 23:58:11	3281.00	No

Administración de Facturas con la opción de ver sus detalles.



Cedula	Nombre	Username	Contraseña	Telefono	Comision	Total vendido	Direccion
111-111111-1	Yuberlys Ochoa	ychoa	123456789	809-435-2000	2087.74	41754.67	Monte Cristi, S...
222-222222-2	Marcos Lozano	mlozano	12	809-261-7332			Sanchez Rami...
402-3373090-1	Gabriel Cepeda	gcepeda	1234	809-729-0203	1416.25	28325.00	Azua, Azua de ...
402-3373090-7	Jose Carlos Be...	DavyBM	123456789	809-586-2480	196.11	3922.20	Santiago, Santi...

Administración de vendedores

## Modificar Vendedor: X

Cédula:

402-3373090-7

Nombre:

Jose Carlos Betances

Número:

809-586-2480

Username:

DarvyBM

Contraseña:

••••••••

Provincia:

29-Santiago

Municipio:

29-Santiago de los Caballeros

Dirección:

Bella Vista

Cargar Imagen

Cancelar

Modificar

Registrar y/o modificar un vendedor.

## Pedidos

Administración

Confirmar

+ Distribuidor

Cancelar

<Todos>

Codigo	Distribuidor	Producto	Cantidad	Precio total	Fecha de Solicitud	Procesado
1	Cecomsa	ASDFGH12	14	1680.00	2022-08-08	0
2	PC Shop	DEAS1D85	2	24646.00	2022-08-08	1
3	PC Shop	ABCDEFG1	20	20000.00	2022-08-08	0
4	Globatech SRL	HJKLMNOP2	19	35150.00	2022-08-08	0
5	CeducComp	OPQRSTU3	13	84500.00	2022-08-08	0
6	CeducComp	VXYZ123A	20	20000.00	2022-08-08	0
7	Globatech SRL	PRODUC12	10	95000.00	2022-08-08	1
8	CeducComp	PRUEBA321	8	13600.00	2022-08-08	1
9	Globatech SRL	MASDKNA1	12	84000.00	2022-08-08	1
10	AMS Multiservice	B12345X	11	27500.00	2022-08-08	1
11	AMS Multiservice	A123456X	10	18000.00	2022-08-08	1
12	AMS Multiservice	DEAS1D85	6	73938.00	2022-08-08	1
13	CeducComp	ZXCNAK54	5	500.00	2022-08-08	1
14	PC Shop	ZXCNAK54	5	500.00	2022-08-08	1
15	CeducComp	ZXCNAK54	5	500.00	2022-08-08	1
16	AMS Multiservice	ZXCNAK54	3	300.00	2022-08-08	1
17	Cecomsa	1359753A	5	16250.00	2022-08-08	1
18	CeducComp	1359753A	5	16250.00	2022-08-08	1
19	PC Shop	ZXCNAK54	5	500.00	2022-08-08	1
20	CeducComp	ZXCNAK54	5	500.00	2022-08-08	1
21	PC Shop	1359753A	3	9750.00	2022-08-08	1
22	CeducComp	ZXCNAK54	5	500.00	2022-08-08	1

Administración de pedidos, con la opción de confirmar un pedido no procesado o agregar distribuidores.

## Confirmar pedido:

Cantidad:

1

Distribuidor:

<Seleccione>

Administrador:

Nombre empleado

Fecha Actual:

08-08-2022

Cancelar

Añadir

Confirmar un pedido.

## Añadir Distribuidor:

Nombre:

Cancelar

Añadir

Añadir un distribuidor.



Estadísticas

Administración

Clientes

productos

Vendedores

Facturas

Combos

Ventas Gráfico

Cancelar

Producto más comprado:

NumSerie: VXYZ123A Cant: 12

Producto menos comprado:

NumSerie: A123456X Cant: null

Estadística y datos de los productos.

Estadísticas

Administración

Clientes

productos

Vendedores

Facturas

Combos

Ventas Gráfico

Cancelar

Vendedor del mes

Yuberkys Ochoa | 111-1111111-1 | total: 11888.0

Vendedor con más facturas realizadas:

402-3373090-1 - Gabriel Cepeda - cant: 13.0

Vendedor con menos facturas realizadas:

222-2222222-2 - Marcos Lozano - cant: 1.0

Estadística y datos de los vendedores



Estadística y datos de las facturas.



Gráfico con las ventas realizadas durante el mes actual.



# Historial Cambio de Precios

numHistorial	Fecha	Precio Antiguo	Precio Nuevo	numSerie	Producto	Marca
1	2022-08-08 14:12...	45.00	25.00	ZXCNAK54	Memoria Ram	Kingston
2	2022-08-08 14:12...	225.00	300.00	ASDFGH12	Disco Duro	Seagate
3	2022-08-08 14:12...	1500.00	1231.00	ASDADADA	Disco Duro	Toshiba
4	2022-08-08 14:12...	10999.00	12323.00	DEAS1D85	Memoria Ram	Kingston
5	2022-08-08 14:12...	2000.00	1800.00	ABCDEFG1	Disco Duro	WD
6	2022-08-08 14:12...	2499.00	2500.00	HIJKLMN2	Disco Duro	Seagate
7	2022-08-08 14:12...	10000.00	9000.00	OPQRSTU3	Disco Duro	Seagate
8	2022-08-08 14:12...	995.00	1000.00	VXYZ123A	Disco Duro	Google
9	2022-08-08 14:12...	11000.00	9500.00	PRODUC12	Tarjeta Madre	Aorus
10	2022-08-08 14:12...	3500.00	2500.00	PRUEBA321	Disco Duro	SanDisk

Ventana que muestra todos los cambios realizados al precio de un producto.



*Tecno e*