

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR JOSÉ MIGUEL CARRERA**

“Sistema de compra y venta para local de repuestos de automóvil”

Trabajo de Titulación para optar al
Título de Técnico Universitario en
INFORMÁTICA

Alumnos:

Christian Andrés Concha Caro

Emmanuel Jesús Gamboa Serey

Profesor guía:

Dagoberto Cabrera Tapia

RESUMEN

KEYWORDS: SISTEMA DE COMPRA Y VENTA – REPUESTOS – AUTOMOVIL – BASE DE DATOS

Este trabajo de titulación tiene por objetivo el desarrollo de un sistema de compra y venta para ser utilizado por trabajadores de la empresa de venta de repuestos automovilísticos Autosave.

Este sistema tiene por primera función el ser un sistema que apoye el área de compra y venta de productos en la empresa llevando un registro de cada uno de estos procesos así como un detalle de estos, también apoya otros ámbitos de la empresa como la gestión en su diario quehacer, facilitando la administración del inventario, registro de clientes, usuarios, proveedores, medios de pago, categorías de productos, marcas de productos, movimientos de dinero y stock. También actúa como sistema de compra y venta, llevando un registro de cada venta o compra realizada por la empresa, así como un detalle de esta.

Este sistema de compra y venta fue desarrollado durante un periodo académico de 10 meses, mediante el entorno de desarrollo integrado Visual Studio y Las bases de datos están implementadas utilizando el motor MySQL.

Este texto se organiza en capítulos, cuyos contenidos comprenden los siguientes temas:

Capítulo 1: Descripción y breve historia de la organización (Empresa de venta de repuestos automovilísticos Autosave), identifica sus partes y las entidades que podemos reconocer, Se identifican los problemas (como la falta un sistema de inventario, o de un registro de las ventas) y las necesidades que tiene la empresa y se plantea el sistema como una solución.

Capítulo 2: Se detalla la parte de hardware en donde se desarrollará y funcionará el sistema. También se detallan el total de tablas que representan las entidades de información que el proyecto necesita para el correcto funcionamiento del sistema y que este pueda almacenar de la mejor manera la información con respecto a los repuestos de automóvil que la empresa dispone para su venta.

Capítulo 3: Aquí se describen el funcionamiento del sistema a través de diferentes diagramas, se presentan lista de la totalidad de los programas que constituyen el sistema y permiten su correcta funcionalidad (Realizar y llevar registro de ventas, Controlar Stock de productos). También se detallan aquellos programas que son más relevantes, adjuntando su código fuente.

En la parte final del documento se presentan las conclusiones del proyecto, incluyendo la importancia de la parte de análisis en la creación de un sistema de información para no crear carga de trabajo adicional y la importancia del trabajo en equipo la hora de programar. Finalmente se presentan páginas web que ayudaron a la finalización del proyecto.

ÍNDICE

RESUMEN

INTRODUCCIÓN

1

CAPÍTULO 1: ASPECTOS RELEVANTES DEL DISEÑO LÓGICO

3

1.1 Descripción de la organización.

4

1.1.1 Organigrama

4

1.1.2 Productos que suministra

5

1.2 Descripción de situación actual.

6

1.3 Problemas detectados

10

1.4 Descripción del sistema propuesto.

11

1.4.1 Objetivos

11

1.4.2 Descripción general del sistema propuesto.

12

1.4.3 Diagrama flujo administrativa

12

1.4.3.1 Nomenclatura utilizada

12

1.4.3.2 Diagrama para venta

13

1.4.3.3 Diagrama de compras

14

1.4.4 Funcionalidades del sistema

15

1.4.5 Entradas y salidas

18

1.4.5 Entradas y salidas.

19

| | |
|---------------------------------|----|
| 1.4.7 Modelo lógico de datos. | 20 |
| 1.4.8 Estructuras de código | 21 |
| 1.4.9 Condicionantes de diseño. | 21 |

CAPÍTULO 2: MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE LA BASE DE DATOS

| | |
|--|----|
| | 22 |
| 2.1 Descripción del recurso computacional | 23 |
| 2.1.1 Configuración del sistema | 23 |
| 2.1.2.1 Software utilizado | 25 |
| 2.1.2.2 Descripción del motor de base de datos | 26 |
| 2.2 Descripción de la base de datos. | 27 |

CAPÍTULO 3: DESCRIPCIÓN DE PROGRAMAS

| | |
|------------------------------|----|
| | 40 |
| 3.1 Diagramas | 41 |
| 3.1.1 Diagrama de menú | 41 |
| 3.1.2 Diagrama Modular | 42 |
| 3.2 Descripción de programas | 43 |

| | | |
|-------|-------------------------------|----|
| 3.2.1 | Tabla de programas | 44 |
| 3.3.1 | Detalle de programas | 46 |
| 3.3.1 | Inicio | 46 |
| 3.3.2 | Mantenedor de productos | 49 |
| 3.3.3 | Venta | 52 |
| 3.3.4 | Compra | 57 |
| 3.3.5 | Ajuste de stock | 60 |
| | Conclusión | 63 |
| | Bibliografía | 65 |
| | Anexo 1: Código fuente | 66 |

Introducción

Los Sistemas de Información y las Tecnologías de Información han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas o reducir la ventaja de los rivales.

Las Tecnologías de la Información han sido conceptualizadas como la integración y convergencia de la computación, las telecomunicaciones y la técnica para el procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

Con el fin de mejorar la productividad y el rendimiento de una organización competitiva, es fundamental evaluar las técnicas actuales y la tecnología disponible para desarrollar sistemas que brinden eficiencia y eficacia de la gestión de la información relevante.

La implementación de sistemas de información en una compañía, brindan la posibilidad de obtener grandes ventajas, incrementar la capacidad de organización de la empresa, y tornar de esta manera los procesos a una verdadera competitividad, para ello es necesario un sistema eficaz que ofrezca múltiples posibilidades, permitiendo acceder a los datos relevantes de manera frecuente y oportuna, por eso, quienes deseen convertir su emprendimiento en una verdadera empresa competitiva insertada en el mercado actual, deberán analizar detalladamente la implementación necesaria de sistemas de información precisos.

En el caso que plantea en este proyecto, la dueña de la tienda, al ser nueva en el rubro del comercio, desconoce la gran importancia de implementar un control sistematizado del flujo de información de inventario o de las actividades relacionadas con la venta. Esto,

usualmente genera como consecuencia, la tenencia inexacta de datos contables y financieros, pérdidas considerables en el ejercicio y una dinámica poco eficiente.

En todos los almacenes de venta, es necesario llevar un eficiente control de inventario, de lo contrario se genera descontrol en la organización, lo cual puede ser aprovechado para robos sistemáticos del tipo “merma”, lo que genera pérdidas considerables y a su vez causa un impacto negativo en el desarrollo y crecimiento de la empresa.

Si un microempresario no sabe con precisión lo que posee en almacenamiento, no puede saber qué es lo que debe ordenar. Si el balance de inventario es no perecedero, los elementos que hay en el punto de almacenamiento pueden quedar en residuos. Puesto que, en la mayoría de los casos el espacio de almacenamiento es un activo valioso, el uso de espacio de almacenamiento debe ser utilizado de manera eficiente.

Por ello se plantea desarrollar un sistema de información que satisfaga las necesidades que se presentan en la gestión de una empresa dedicada al compra y venta.

**CAPÍTULO 1:
ASPECTOS RELEVANTES DEL DISEÑO LÓGICO.**

Capítulo 1: Aspectos relevantes del diseño lógico.

1.1 Descripción de la organización.

Nombre: AutoSave

Representante legal: Daniela Iris Navia Bernal

Tipo empresa: AutoSave es una Sociedad de Responsabilidad limitada, con solo 2 miembros la Señora Daniela Navia y su esposo.

Objetivos: “Nuestro fin es de atender a nuestros clientes lo mejor posible y brindarles la mayor calidad, al precio más conveniente, abarcando las principales líneas de repuestos alternativos.

Nuestro afán constante de brindarle al cliente, la mejor atención, nos ha hecho día a día ir mejorando como empresa y eso nos ha permitido crecer en el tiempo, siendo una empresa en progreso nuestro valor agregado es nuestra atención, que el cliente siempre vuelva a nosotros y se sienta satisfecho con lo que le ofrecemos.”

1.1.1 Organigrama

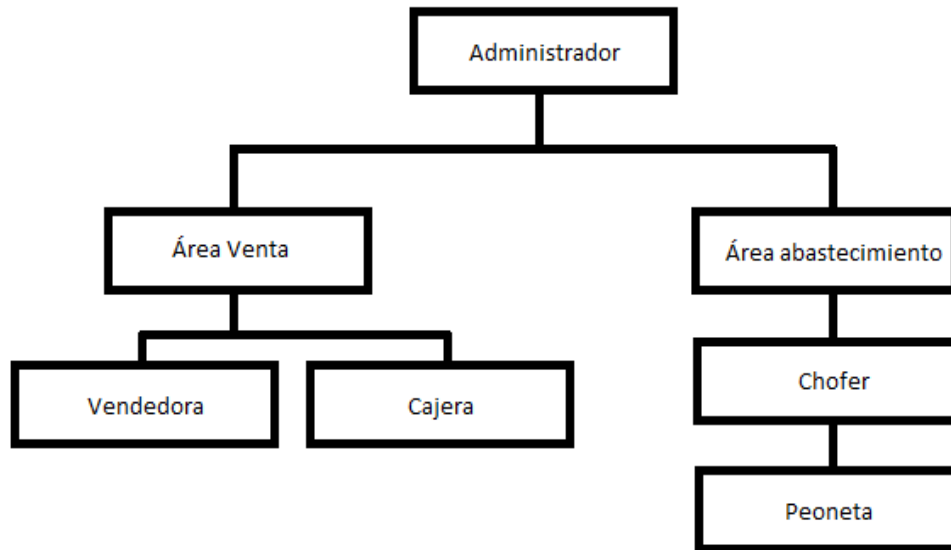


Figura 1-1. Organigrama de la empresa

1.1.2 Productos que suministra

Todos los repuestos que se venden son importados y de carácter alternativo, atendiendo necesidades de reparación o cambio de piezas de los automóviles pertenecientes al parque vehicular livianos (Automóviles, Todo terreno 4x4, Furgones, Camionetas, Camiones pequeños).

Todos los productos están basados en un precio que está directamente diferenciado por la calidad y el origen de éstos.

Los productos que mayormente se venden son:

- Luces led. (ver Figura 1-2)
- Ampolletas. (ver Figura 1-3)
- Aceites. (ver Figura 1-4)
- Filtros. (ver Figura 1-5)



Figura 1-2. Luces led



Figura 1-3. Ampolletas



Figura 1-4. Aceites



Figura 1-5, Filtros

1.2 Descripción de situación actual.

El local de repuestos automovilísticos Autosave se encuentra ubicado en la ciudad de Olmué, es un negocio familiar, el cual es atendido por la señora Daniela Navia quien es la dueña. Este local tiene una pequeña trayectoria de un poco más de un año funcionando, a pesar de este poco periodo de tiempo funcionando el local ha conseguido una clientela considerable, formando un negocio rentable (ver Figura 1-6). Actualmente está presentando problemas producidos debido a la competencia que ha ido creciendo en la ciudad y a la falta de nuevas tecnologías que les ayuden a mantener un orden sobre los productos que se venden en el local.



Figura 1-6. Repisas del local

En la actualidad el local presenta un horario de atención de 9:00 a 14:00 y de 17:00 a 20:00 de lunes a sábado y de 9:00 a 14:00 los días domingos, éste cuenta con una persona encargada de la atención de clientes, además de la dueña, un encargado de abastecer el local, por lo general estos puestos son ocupados por familiares de la dueña. También cuenta con un par de ayudantes, puestos que en ocasiones están vacíos ya que son prescindibles para el funcionamiento del local, éstos no presentan contrato de trabajo y se encargan de labores como carga y descarga de productos, de ayudar al cliente a llevar los productos que éstos compren en el local a su vehículo si lo necesitaran, o atender a clientes si la dueña así lo estipulara.

El recinto cuenta con una pequeña bodega ubicada en la parte trasera del local (ver Figura 1-7), la cual es abastecida regularmente de productos los cuales se encuentren bajo del monto mínimo que la dueña estima conveniente debe tener el local en cada momento, el encargado de abastecer el local también debe dejar una cantidad de productos en el hogar de la dueña, debido a que a ésta le gusta siempre tener una cantidad de reserva en casos de emergencia. Además de eso el encargado de abastecer tiene que preocuparse de la descarga de los productos que trae al local en caso de no haber ayudantes actualmente trabajando ahí.



Figura 1-7. Bodega

En el pasado, el local intentó utilizar Microsoft Excel para llevar un registro del stock del local, pero debido al desconocimiento de la dueña sobre la herramienta y la dificultad que resultaba para ésta el manejar un computador, este método fue abandonado. Luego intentaron mantener un registro del stock en cuadernos de una manera rústica, la cual consistía en tachar el número de stock anterior y escribir uno nuevo al costado o en ocasiones ocupaban corrector para borrar el stock anterior. Éste método al final fue eliminado debido a que los trabajadores no actualizaban con regularidad el stock por lo cual siempre que se revisaba había incongruencias.

Hoy en día su método consiste en mantener los estantes con los productos ordenados (ver Figura 1-8) para así poder saber rápidamente si tienen ese producto, y con el paso del tiempo los trabajadores se han acostumbrado y tienen un conocimiento aproximado siempre del stock que tiene la tienda. El problema aquí surge al momento de contratar nuevo personal, ya que a éstos les cuesta adaptarse debido a que constantemente deben verificar si los productos que vienen a comprar al local se encuentran con stock.

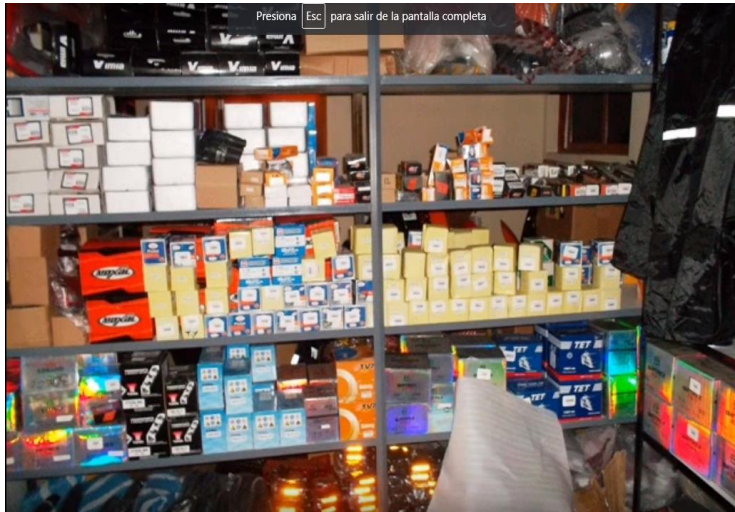


Figura 1-8. Bodega

Actualmente las labores correspondientes a la recopilación, el registro y la revisión de la información en la tienda son realizadas de forma manual, siendo éstas registradas en cuadernos

(ver Figura 1-9), los cuales se almacenan bajo el mostrador del local. Estos registros son constantemente revisados por los mismos trabajadores del local, producto de lo cual resultan dañados o son extraviados.

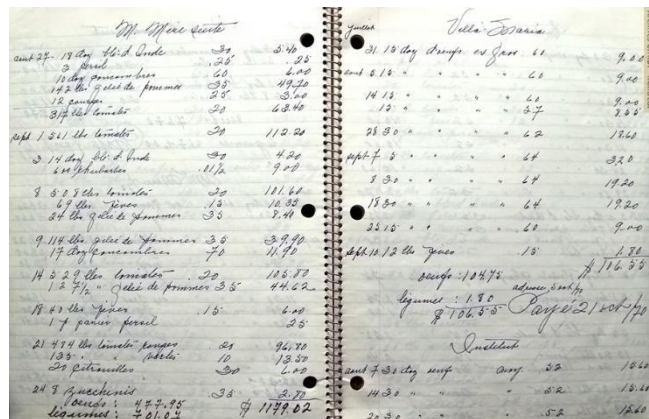


Figura 1-9, Cuaderno de ventas

Estos cuadernos contienen información respecto al valor de venta de los productos que tiene la tienda y un registro de los montos correspondientes a las ventas totales de los días. Al finalizar el día la dueña cuenta los montos que aparecen en las boletas o facturas de ventas que realizó durante el día y en base a ese monto que obtiene saca cálculos, con base

en eso ella ve si le fue mejor o peor que días anteriores, luego esta información la registra en cuadernos que mantiene guardados en su casa. Esta información que registra casi a diario no es precisa, si no es más una aproximación de lo que vende debido a que en ocasiones se les olvida hacer boletas de las ventas que realizan, sobre todo cuando son productos pequeños. Este método que utiliza la dueña le ayuda a saber el monto total de las ventas diarias, mas no le permite saber qué productos vendió en el local durante el día, para saber esto ella tiene que revisar el local para ver qué productos han disminuido o verificar el cuaderno con los precios de venta de los productos para ver qué productos tienen los valores que en las boletas o facturas aparecen.

Toda esta mala organización del local genera costos adicionales a la dueña, como es el caso al perder un cliente por no encontrar el producto que éste estaba buscando, o el extravío de productos que deberían estar en el local, de los cuales no tiene forma de verificar si fueron vendidos o no. Esto le produce dificultad en la toma de decisiones e inseguridad sobre los mismos trabajadores del local, razón por la cual estos mismos ayudantes son reemplazados con regularidad.

1.3 Problemas detectados

Tal como ha sido mencionado, el origen del problema principal de la empresa es debido al manejo manual de la información del negocio, donde se presentan los siguientes problemas relacionados con la gestión:

- Peligro de pérdida de información.
- Es difícil calcular las ganancias.
- No hay registros de los proveedores.
- No hay registros de los clientes.
- No hay registros de los precios de venta de los productos.
- No hay un control de stock que ayude a realizar pedidos a los proveedores.
- Cuando se requiere alguna información histórica, la búsqueda de ésta se hace complicada y demorosa, ya que hay que revisar cuaderno por cuaderno.

- Retraso en las acciones, lo cual puede producir pérdida de potenciales compradores
- Posibles inconsistencias en los datos.

1.4 Descripción del sistema propuesto.

1.4.1.1 Objetivo general

Automatizar la gestión del negocio en las áreas de ventas y stock por medio de un sistema computarizado de información el cual también facilitará información confiable para la toma de decisiones.

1.4.1.2 Objetivos específicos

- Crear un sistema seguro con usuario y contraseña.
- Crear un menú que permita realizar y organizar las diferentes funciones.
- Llevar un registro de los productos vendidos.
- Llevar un registro de los usuarios del sistema.
- Llevar un registro de productos devueltos.
- Llevar un registro de medios pago utilizables.

- Crear consulta de precio de un producto.
- Llevar un registro de los proveedores.
- Mostrar el producto más vendido.
- Mostrar el producto menos vendido.
- Alertar el stock crítico de productos.
- Crear e imprimir boleta o factura según corresponda.

1.4.2 Descripción general del sistema propuesto.

El sistema a construir facilitará al usuario la compra y venta de productos en un local de repuestos, entregando la información necesaria correspondiente. El sistema que se pretende realizar estará dividido en dos subsistemas: Gestión Ventas y Control de Stock. Para garantizar la seguridad de la información de la empresa, el sistema contará con control de acceso de usuarios, el cual restringirá las funciones críticas del sistema como son la información contable en caja, manejo de excepciones en el movimiento de los productos en bodega. El sistema tendrá 2 tipos de usuario los que serán vendedor el que tendrá acceso limitado al sistema solo podrá realizar ventas y agregar clientes si se requiere, y el usuario administrador será quien tenga acceso total al sistema.

1.4.3 Diagrama flujo administrativa

Los DFA que se presentan a continuación muestra el ciclo de trabajo en los procesos de Venta y control de stock, las cuales serán reflejadas en el sistema a desarrollar.

1.4.3.1 Nomenclatura utilizada

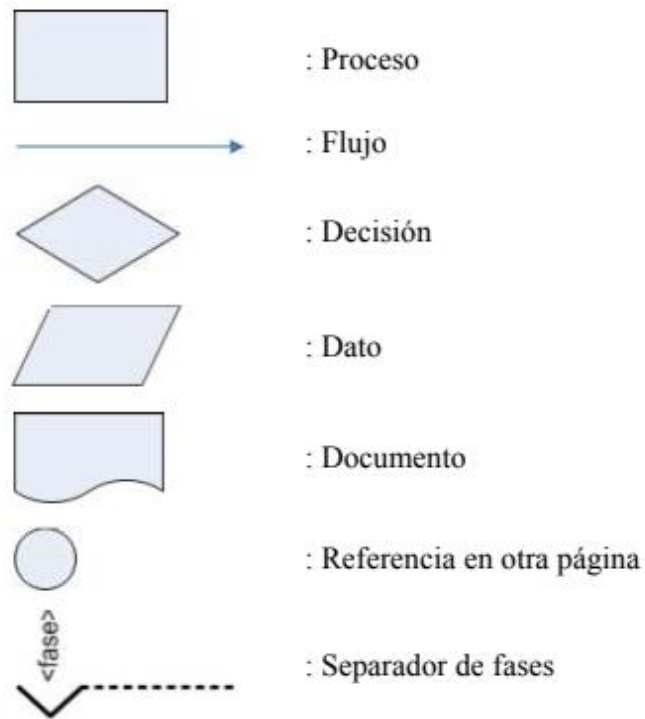


Figura 1-8. Nomenclatura del diagrama

1.4.3.2 Diagrama para venta

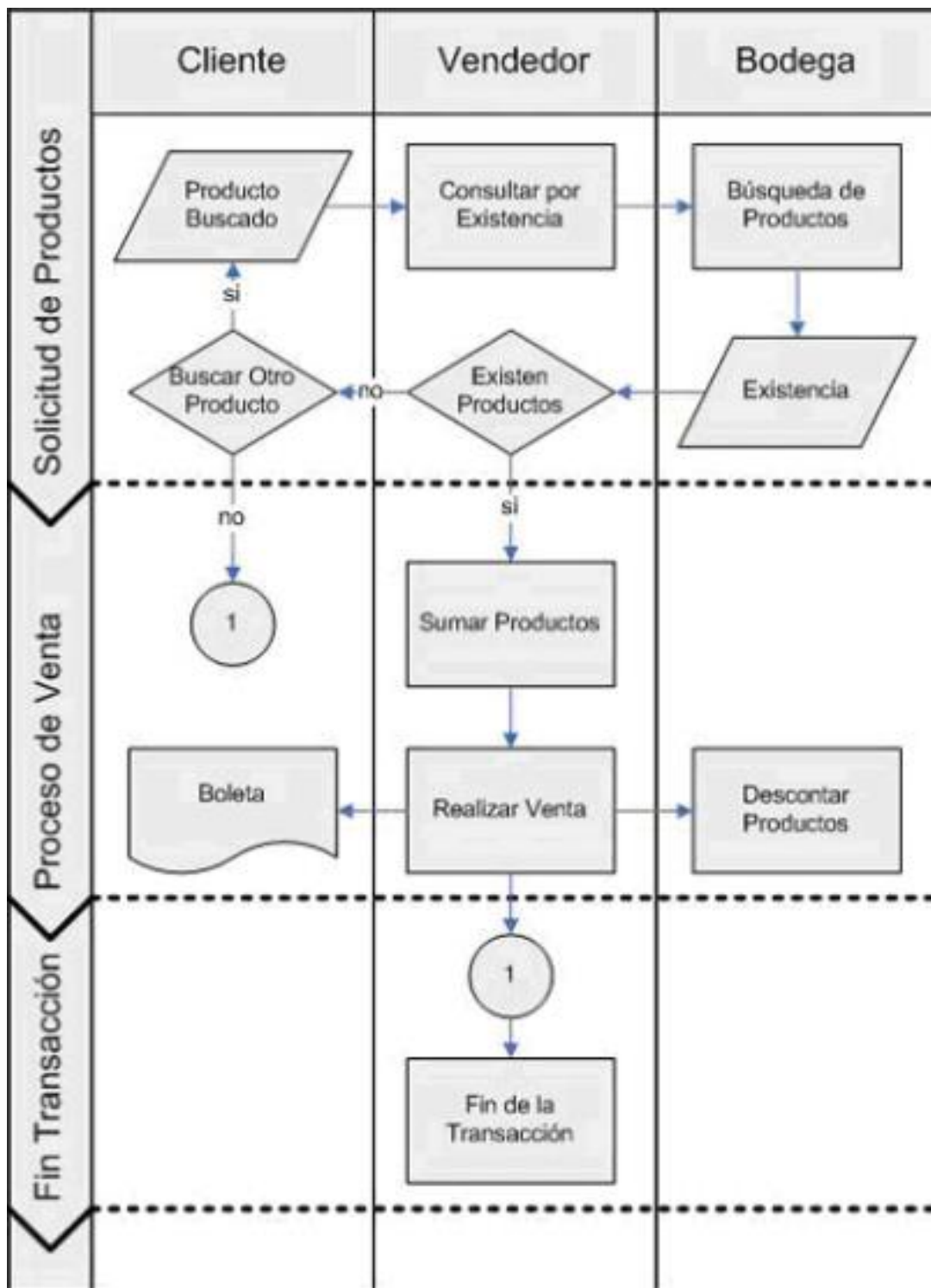


Figura 1-9. Diagrama de ventas

1.4.3.3 Diagrama de compras

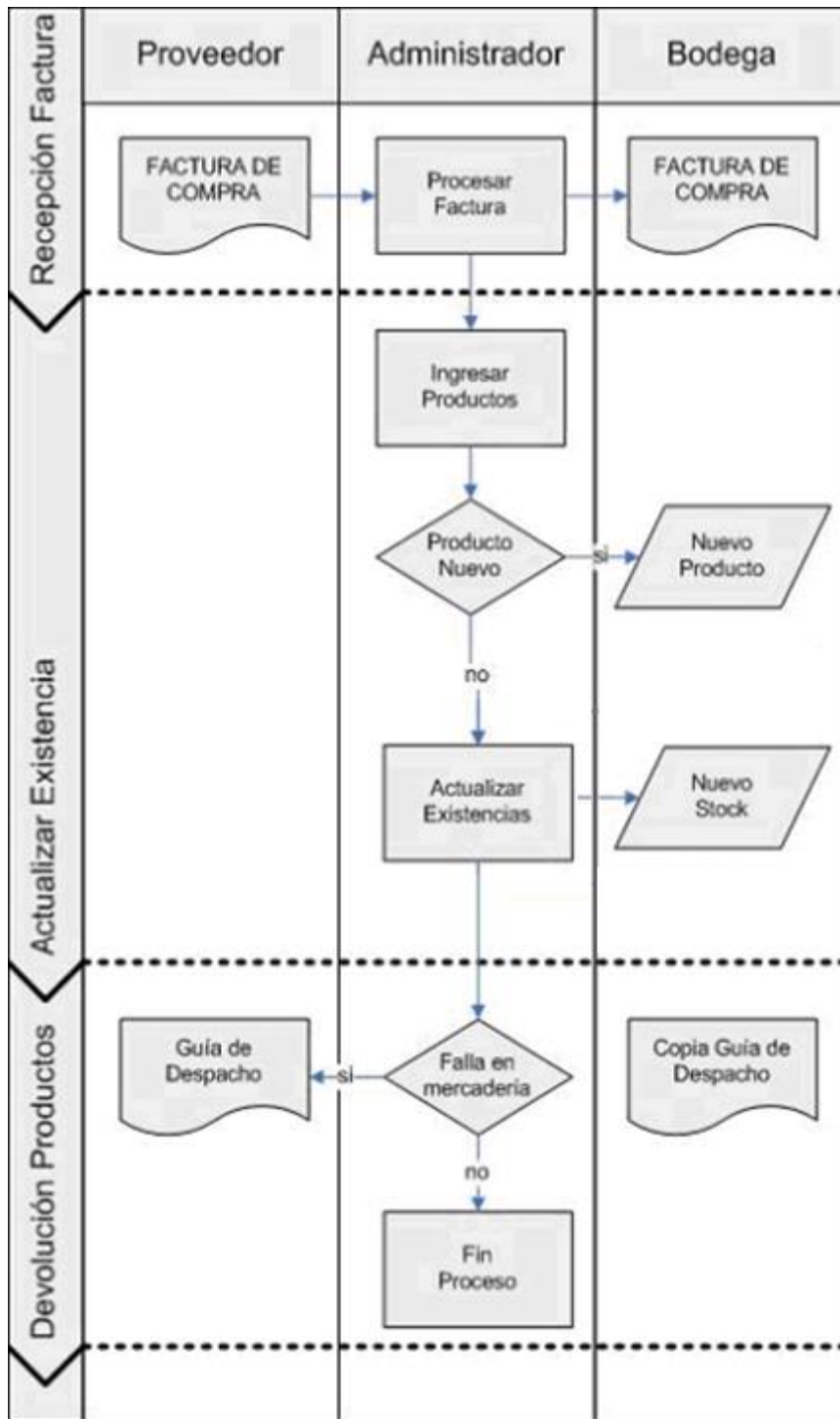


Figura 1-10. Diagrama de compras

1.4.4 Funcionalidades del sistema.

Control de venta:

- Venta: Este proceso está encargado de registrar cada venta efectuada en el local. Consta de los siguientes procesos:
- Ingresar productos que el cliente quiere comprar.
- Cuando la venta finalice se generará una boleta o factura según corresponda.
- Anular Venta: Este proceso consiste en la anulación de una venta realizada.
- Devolución: Este proceso consiste en registrar la devolución de un(os) producto(s) por parte del cliente en caso de haber un problema técnico.

Control de compra:

- Compra: Este proceso está encargado de registrar cada compra efectuada para el local. Consta de los siguientes procesos:
- Ingresar productos con el monto de éstos a comprar en una orden de compra
- Cuando la compra de los productos seleccionados se complete actualizar el stock de los productos comprados.
- Anular Compra: Este proceso consiste en la anulación de una orden de compra realizada.

Imprimir boleta/factura: Una vez generada una boleta o factura después de una venta, ésta se imprimirá

Consultar Precio: Este proceso consiste en obtener el precio de algún producto mediante una búsqueda por filtros de tipo, nombre o código.

Control de caja: Este proceso consiste en desplegar por pantalla los movimientos de dinero durante un período de tiempo previamente establecido.

Control de Stock: Este proceso llevará la cuenta de todos los productos existentes en el local. Consta de los siguientes procesos:

- Ingreso de un pedido: Permite agregar a stock los productos que sean comprados a los proveedores.
- Ingresos de ventas: Permite rebajar el stock total de algún producto que se haya vendido.
- Ingreso de otros movimientos: Permite el agregar o quitar stock de un producto por otra causa que no sea compra o venta. ejemplo: robo.

Mantenedores:

- Mantenedor de Categorías: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) las categorías de productos que se encuentren registradas.
- Mantenedor de clientes: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) los clientes habituales de la tienda.
- Mantenedor de Marcas: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) las marcas de productos que se encuentren registradas.
- Mantenedor de Medios de Pago: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) los medios de pago que se encuentren registrados.
- Mantenedor de productos: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) los productos que se encuentren registrados.
- Mantenedor de proveedores: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) los proveedores que se encuentren registrados.
- Mantenedor de usuarios: Permite agregar, modificar, consultar, eliminar e imprimir (Con opción de filtrado) los usuarios con acceso al sistema.

Generación de Informes:

- Reportes de categorías desplegará una lista con todos los datos de las categorías registradas en la tabla categoría, dichos datos podrán ser filtrados por medio de la clave principal `Codigo_Categoria`.
- Reportes de Clientes desplegará una lista con todos los datos de los clientes registrados en la tabla clientes, dichos datos podrán ser filtrados por medio de la clave principal `Rut_Cliente`.
- Reporte de Marcas desplegará una lista con todos los datos de las marcas registradas en la tabla marca, dichos datos podrán ser filtrados por medio de la clave principal `Codigo_Marca`.
- Reporte de Medios De Pago desplegará una lista con todos los datos de los medios de pago registrados en la tabla `medio_de_pago`, dichos datos podrán ser filtrados por medio de la clave principal `Codigo_Pago`.
- Reporte de Productos desplegará una lista con todos los datos de las productos registrados en la tabla productos, dichos datos podrán ser filtrados por medio de la clave principal `Codigo_Producto`, las claves foráneas `Codigo_Marca`, `Codigo_Categoria` o por medio del Nombre.
- Reporte de Proveedores desplegará una lista con todos los datos de los proveedores registrados en la tabla proveedores, dichos datos podrán ser filtrados por medio de la clave principal `Rut_Proveedor`.
- Reporte de Usuarios desplegará una lista con todos los datos de los usuarios registrados en la tabla usuarios, dichos datos podrán ser filtrados por medio de la clave principal `Rut_Usuario`.
- Alertas de stock crítico de productos: Permite conocer los productos que se encuentran con bajo stock. Se avisará mediante una alerta.
- Reportes de los productos más y menos vendidos: Permite saber cuál o cuáles son los productos más y menos vendidos según un periodo de tiempo que establezca un usuario.

1.4.5 Entradas y salidas.

Entradas:

- Datos de Categoría (Código Categoría, Nombre)
- Datos de cliente (Rut Cliente, Nombre, Apellido, Teléfono, Dirección).
- Datos de Marca (Código Marca, Nombre)
- Datos de Medios de Pago (Código Pago, Nombre)
- Datos de Productos: (Código Producto, Código Marca, Código Categoría, Nombre, precio de compra, precio de venta, stock mínimo, stock actual, descripción).
- Datos de Proveedores: (Rut Proveedor, Nombre proveedor o empresa, Teléfono, Email, dirección).
- Datos de Usuario: (Rut Usuario, Nombre, Apellido, Teléfono, Dirección, Contraseña, Nivel de acceso).

Salidas:

- Lista de Categoría: Se desplegará en pantalla una lista de categorías.
- Lista de Clientes: Se desplegará en pantalla una lista de los clientes.
- Lista de Marcas: Se desplegará en pantalla una lista de marcas.
- Lista de Medios de Pago: Se desplegará en pantalla una lista de medios de pago.
- Lista de Productos: Se desplegará en pantalla una lista de productos disponibles para la venta.
- Lista de Productos con bajo stock: Se desplegará en pantalla una lista de productos que tienen bajo stock.
- Lista de Proveedores: Se desplegará en pantalla una lista de los proveedores que tiene el local.
- Lista de Usuarios: Se desplegará por pantalla una lista de los usuarios del sistema.
- Informe de Flujo de Caja: Se desplegará por pantalla los movimientos totales del día.

- Informe de Ventas: Se desplegará por pantalla todos los productos vendidos por día o en un periodo de tiempo determinado por el usuario.
- Boleta/factura legal del SII: se desplegará por pantalla una boleta o factura según corresponda con los datos de una venta, posteriormente será impresa.

1.4.6 Entidades de información.

- Ajuste de Stock: contiene datos sobre los movimientos de stock de los productos del sistema.
- Categoría: contiene datos sobre la categoría de los productos que se venden en la tienda.
- Clientes: contiene datos sobre los clientes habituales de la tienda.
- Descuentos: contiene datos de los posibles descuentos para una venta
- Marca: contiene datos sobre la marca de los productos que se encuentran en la tienda.
- Medio de pago: contiene datos referentes a los métodos de pago accesibles en la tienda.
- Movimientos: contiene datos de los movimientos de dinero en la tienda.
- Productos: contiene datos sobre los productos que se venden en la tienda.
- Proveedores: contiene datos sobre los proveedores que posee la tienda.
- Usuario: contiene datos sobre los usuarios con permiso para acceder al sistema.
- Venta: contiene registros sobre las ventas realizadas en la tienda.
- Detalle venta: contiene datos referentes al detalle de las ventas realizadas en la tienda.
- Orden pedido: contiene datos referentes a los pedidos de productos realizados por la tienda a los proveedores.
- Detalle Orden pedido: contiene el detalle de los pedidos realizados por la tienda a los distintos proveedores.

1.4.7 Modelo lógico de datos.

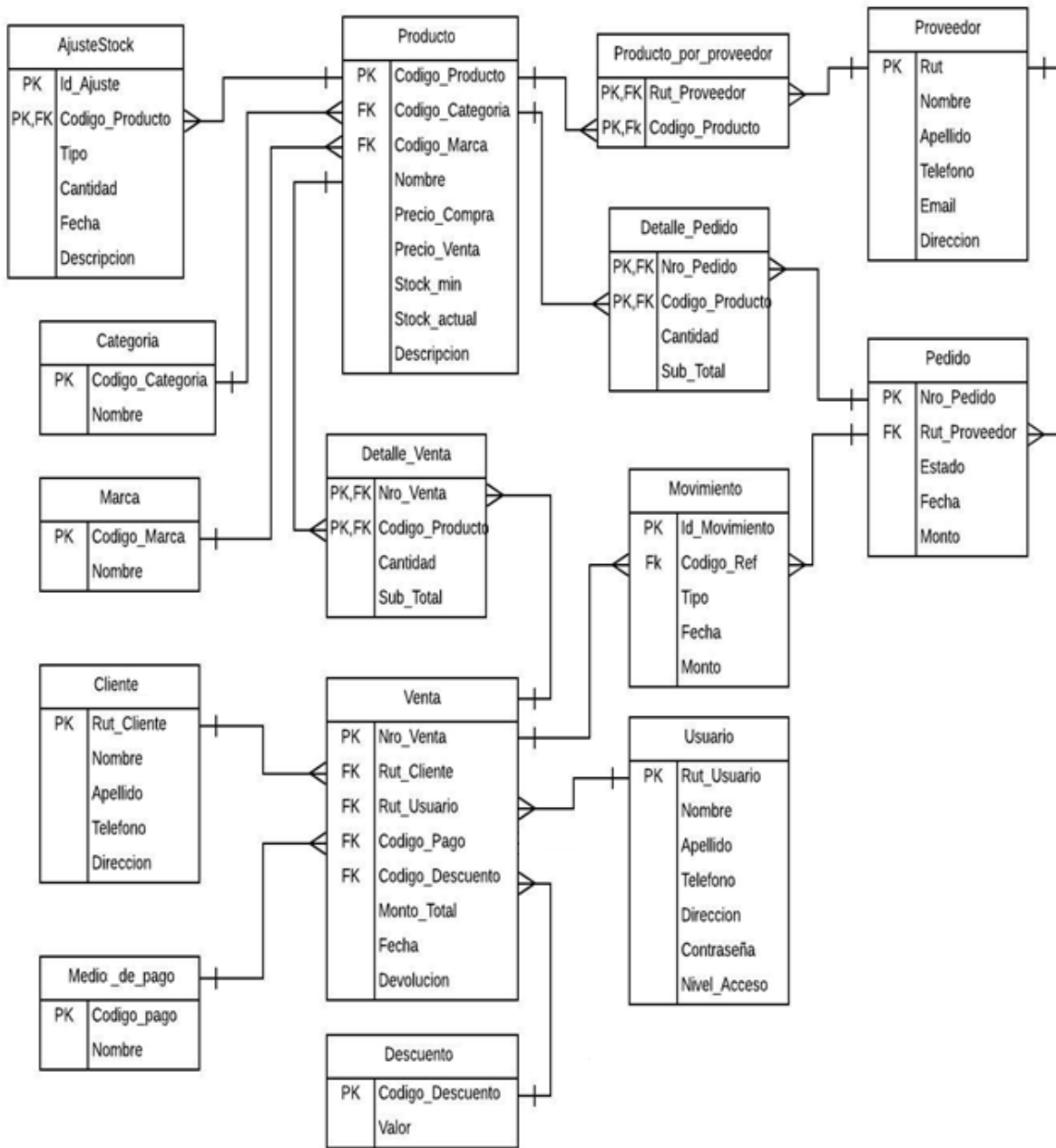


Figura 1-11. Modelo lógico de datos

1.4.8 Estructuras de código.

Todos los códigos utilizados por el programa son numéricos, pero el usuario no se deberá preocupar por la administración de estos ya que el mismo programa se encargará de generar códigos numéricos autoincrementable.

1.4.9 Condicionantes de diseño.

Arquitectura

Debido a que la tienda cuenta con tan solo una caja y tampoco los dueños piensan en expandirse, no es necesario que el sistema funcione en red, por lo que será construido con arquitectura standalone donde todo el software está concentrado en la misma máquina, es decir, que la aplicación y las bases de datos residen en la misma PC que a su vez, es desde el único lugar donde se lo puede y debe operar.

CAPÍTULO 2:
MEDIO AMBIENTE COMPUTACIONAL Y DESCRIPCIÓN DE LA BASE DE DATOS.

Capítulo 2: Medio ambiente computacional y descripción de la base de datos.

2.1 Descripción del recurso computacional

2.1.1 Configuración del sistema

A continuación, se detallarán los aspectos de hardware y software que se utilizará para diseñar y modelar el sistema.

Tabla 2-1. Hardware para el desarrollo del sistema

| | |
|-------------------|---------------------|
| Marca | Asus |
| Modelo | x456 |
| Procesador | Intel core I7-6500u |
| Velocidad | 2,5 GHz |
| Disco duro | 1TB |
| Memoria RAM | 8GB |
| Tarjeta gráfica | Nvidia GTX 940m |
| Usb | 3.0 x 1, 2.0 x 1 |
| Sistema operativo | Windows 10 64bits |

Tabla 2-2. Hardware donde funcionara el sistema

| | |
|------------|--------------------|
| Marca | HP |
| Modelo | Pavilion 14-n005la |
| Procesador | AMD A4-5000 |
| Velocidad | 1,5 GHz |

| | |
|-------------------|-------------------|
| Disco duro | 500GB |
| Memoria RAM | 8GB |
| Usb | 2.0 x 3 |
| Sistema operativo | Windows 10 64bits |

Tabla 2-3. Impresora que se utilizará

| | |
|---------------------------------|-------------------------------|
| Marca | Brother |
| Tecnología | Láser electrofotográfico |
| Interfaz | USB 2.0 |
| Sistemas operativos compatibles | Windows/Mac/Linux |
| Soporte impresion | Papel normal, papel reciclado |
| Tamaño soporte impresión | A5, carta, A4 y legal |

2.1.2.1 Software utilizado

Herramienta de desarrollo: Microsoft visual studio.

Microsoft visual studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic.Net, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, Visual

Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .Net.

Lenguaje a utilizar: Visual Basic.Net

Visual Basic es un lenguaje desarrollado por Microsoft a principios de la década de los noventa y que se ha seguido usando en distintas versiones, hasta hoy día. Tras su lanzamiento adquirió gran popularidad porque Microsoft facilitó un entorno de programación amigable, que era de fácil uso para los programadores y permitía crear aplicaciones para Windows con aspecto profesional y con rapidez.

Visual Basic es un diseñador de entornos de datos, donde es posible generar de manera automática conectividad entre controles y datos mediante la acción de arrastrar y colocar objetos sobre formularios.

Se utilizará este lenguaje de programación pues aprovecha al máximo la API de Windows y además porque desarrolla rápidamente las interfaces gráficas adaptadas al máximo al sistema operativo, lo que minimiza el tiempo de desarrollo y optimiza la ejecución del programa.

2.1.2.2 Descripción del motor de base de datos

MariaDB:

Sin dudar, MySQL es uno de los mejores motores de bases de datos más conocidos y usados a nivel mundial, compatible con decenas de lenguajes. Sin embargo, debido a la compra de la empresa por parte de Oracle, surge un problema, en los últimos años con este

software, y muy relevante para los desarrolladores. Básicamente, al caer en manos de Oracle, MySQL ya no es completamente Open Source, sino que se acerca el software comercial. Ante esta situación, se creó MariaDB, que es una variante de MySQL que mantiene toda la filosofía del software libre y totalmente compatible MySQL.

MariaDB se puede usar desde la mayoría de sistemas de administración existentes para MySQL, En nuestro caso se utilizará PhpMyAdmin.

PhpMyAdmin es un sistema relacional de bases de datos que en el último tiempo ha tenido un crecimiento que lo lleva a comprar su solidez y versatilidad con el líder del mercado Oracle, con la gran diferencia que este es de código abierto y totalmente libre de licencias.

2.2 Descripción de la base de datos.

2.2.1 Tipos de datos utilizados

Tabla 2-4. Nomenclatura utilizada por los tipos de datos SQL en MariaDB

| Nombre | Tipo | Descripción |
|---------|----------|---------------------------------------|
| TINYINT | Numérico | almacena un número con rango (0-255). |

| | | |
|-----------------|--------------------|--|
| | entero | |
| SMALLINT | Numérico entero | almacena un número con rango (-32768 a 32767). |
| BIGINT | Numérico entero | almacena un número con rango (-9223372036854775808 , 9223372036854775807) |
| INT | Numérico entero | almacena un número con rango (-2,147,483,648 a 2,147,483,647). |
| VARCHAR | Alfanumérico | almacena una cadena de caracteres de rango 0 a 65,535. |
| DATE | Fecha | almacena una fecha con formato AAAA-MM-DD |

2.2.2 Descripción de tablas

Productos

Nombre Lógico: producto

Descripción: Almacena los productos que se encuentran en la tienda.

Clave principal:Codigo_Producto

Clave foránea:Codigo_Marca referencia a tabla marca

Codigo_Categoria referencia a tabla categoría

Tabla 2-5. Productos

| Nombre | Tipo | Longitud | Descripción |
|------------------|----------|----------|-------------------------------|
| Codigo_Producto | INTEGER | 7 | código del producto. |
| Codigo_Marca | SMALLINT | 3 | código de la marca. |
| Codigo_Categoria | SMALLINT | 3 | código de la categoría. |
| Nombre | VARCHAR | 30 | nombre del producto. |
| Precio_Compra | INTEGER | 9 | Precio de compra del producto |
| Precio_Venta | INTEGER | 9 | precio de venta del producto. |
| Stock_Minimo | SMALLINT | 4 | stock mínimo del producto. |
| Stock_Actual | SMALLINT | 4 | stock actual del product |
| Descripción | VARCHAR | 40 | Descripción del product |

Venta

Nombre Lógico: venta

Descripción: Almacena las ventas realizadas en la tienda.

Clave principal: Nro_Venta

Claves foráneas: Rut_Usuario referencia a tabla usuario.

Rut_Cliente referencia a tabla cliente.

Codigo_pago referencia a tabla medio_de_pago.

Codigo_Descuento referencia a table descuento.

Tabla 2-6. Ventas

| Nombre | Tipo | Longitud | Descripción |
|------------------|----------|----------|--|
| Nro_Venta | INTEGER | 9 | Número de la venta |
| Rut_Usuario | INTEGER | 8 | Rut del Usuario |
| Rut_Cliente | INTEGER | 8 | Rut del cliente |
| Codigo_Pago | SMALLINT | 3 | Código del medio de pago |
| Codigo_Descuento | SMALLINT | 3 | Código del descuento aplicado |
| Monto_Total | INTEGER | 9 | Monto total de la venta |
| Fecha | DATE | | Fecha en la que se realizó la venta |
| Devolucion | TINYINT | 1 | Indica si el producto a sido devuelto. 0 – NO, 1 – SI |

Detalle venta

Nombre Lógico: detalle_venta

Descripción: Almacena el detalle de las ventas realizadas en la tienda.

Clave principal: Nro_venta + Codigo_Producto

Clave foránea: Codigo_Producto referencia a tabla producto

Tabla 2-7. Detalle venta

| Nombre | Tipo | Longitud | Descripción |
|-----------------|----------|----------|--|
| Nro_Venta | INTEGER | 9 | número de la venta |
| Codigo_Producto | INTEGER | 7 | código del product |
| Cantidad | SMALLINT | 3 | cantidad del product |
| Sub_Total | INTEGER | 9 | monto obtenido por la cantidad vendida |

Pedido

Nombre Lógico: pedido

Descripción: Almacena las órdenes de pedido realizados por la tienda a sus proveedores.

Clave principal: Nro_Pedido

Clave foránea: Rut_Proveedor referencia a tabla proveedor

Tabla 2-8. Pedido

| Nombre | Tipo | Longitud | Descripción |
|--------|------|----------|-------------|
|--------|------|----------|-------------|

| | | | |
|---------------|---------|----|--|
| Nro_Pedido | INT | 9 | número del pedido |
| Rut_Proveedor | INT | 8 | rut del proveedor |
| Fecha | DATE | | fecha del pedido |
| Valor | INT | 9 | Valor total del pedido |
| Estado | VARCHAR | 20 | Estado en el cual se encuentra el pedido: En Espera/Completado |

Detalle pedido

Nombre Lógico: detalle_pedido

Descripción: Almacena el detalle de los pedidos realizados en la tienda.

Clave principal: Nro_Pedido

Clave foránea: Nro_Pedido referencia a tabla pedido

Codigo_Producto referencia a tabla producto

Tabla 2-9. Detalle_pedido

| Nombre | Tipo | Longitud | Descripción |
|-----------------|----------|----------|-----------------------------|
| Nro_Pedido | INTEGER | 9 | Número del pedido realizado |
| Codigo_Producto | INTEGER | 7 | Código del producto |
| Cantidad | Smallint | 4 | Cantidad del producto |

Cliente

Nombre Lógico: cliente

Descripción: Almacena los clientes regulares de la tienda.

Clave principal: Rut_Cliente

Tabla 2-10. Clientes

| Nombre | Tipo | Longitud | Descripción |
|-------------|---------|----------|--------------------|
| Rut_Cliente | INT | 8 | Rut proveedor |
| Nombre | VARCHAR | 30 | Nombre del cliente |

| | | | |
|-----------|---------|----|-------------------------|
| Apellido | VARCHAR | 20 | Apellido del cliente |
| Teléfono | BIGINT | 11 | Teléfono del cliente |
| Dirección | VARCHAR | 40 | Dirección del proveedor |

Categoría

Nombre Lógico: Categoría

Descripción: Almacena las categorías de productos del sistema.

Clave principal: Código_categoria

Tabla 2-17. Categoría

| Nombre | Tipo | Longitud | Descripción |
|------------------|----------|----------|------------------------|
| Código_categoria | SMALLINT | 3 | código de la categoría |
| Nombre | VARCHAR | 20 | Nombre de la categoría |

Usuario

Nombre Lógico: usuario

Descripción: Almacena a los usuarios del sistema

Clave principal: Rut_Usuario

Tabla 2-11. Vendedor

| Nombre | Tipo | Longitud | Descripción |
|-------------|---------|----------|-------------|
| Rut_Usuario | INTEGER | 8 | Rut Usuario |

| | | | |
|--------------|---------|----|--|
| Nombre | VARCHAR | 30 | Nombre del Usuario |
| Apellido | VARCHAR | 20 | Apellido del Usuario |
| Teléfono | INTEGER | 9 | Teléfono del Usuario |
| Direccion | VARCHAR | 40 | Dirección del Usuario |
| Contrasena | VARCHAR | 20 | Contraseña del Usuario |
| Nivel_Acceso | TINYINT | 1 | Nivel de acceso del usuario al sistema 0-Vendedor/1-Administrador |

Proveedor

Nombre Lógico: proveedor

Descripción: Almacena a los proveedores de la tienda

Clave principal: Rut_Proveedor

Tabla 2-12. Proveedor

| Nombre | Tipo | Longitud | Descripción |
|---------------|---------|----------|----------------------|
| Rut_Proveedor | INT | 8 | Rut proveedor |
| Nombre | VARCHAR | 20 | Nombre del proveedor |

| | | | |
|-----------|---------|----|-------------------------|
| Apellido | VARCHAR | 20 | Apellido del proveedor |
| Telefono | INTEGER | 9 | Teléfono del proveedor |
| Email | VARCHAR | 40 | Email del proveedor |
| Direccion | VARCHAR | 40 | Dirección del proveedor |

Producto por proveedor

Nombre Lógico: producto_por_proveedor

Descripción: Almacena los proveedores correspondientes a cada producto que se tiene en la tienda.

Clave principal: Rut_Proveedor +Codigo_Producto

Clave foránea: Rut_Proveedor referencia tabla proveedor

Clave foránea: Codigo_Producto referencia tabla producto

Tabla 2-13. Producto por proveedor

| Nombre | Tipo | Longitud | Descripción |
|---------------|------|----------|-------------------|
| Rut_Proveedor | INT | 8 | Rut del proveedor |

| | | | |
|-----------------|---------|---|---------------------|
| Codigo_Producto | INTEGER | 7 | Código del producto |
|-----------------|---------|---|---------------------|

Marca

Nombre Lógico: marca

Descripción: Almacena las marcas de los productos que se encuentran en la tienda.

Clave principal: Codigo_Marca

Tabla 2-14. Marca

| Nombre | Tipo | Longitud | Descripción |
|--------------|----------|----------|--------------------|
| Codigo_Marca | SMALLINT | 3 | código de la marca |
| Nombre | VARCHAR | 15 | nombre de la marca |

Movimiento

Nombre Lógico: movimiento

Descripción: Guarda registros de todos los movimientos de dinero realizados.

Clave principal: ID_Movimiento

Clave foránea: Codigo_Ref referencia tabla Venta

Codigo_Ref referencia tabla Compra

Tabla 2-15. Movimientos

| Nombre | Tipo | Longitud | Descripción |
|---------------|---------|----------|-----------------------|
| ID_Movimiento | INTEGER | 9 | Número del movimiento |

| | | | |
|------------|---------|----|--|
| | | | realizado |
| Codigo_ref | INTEGER | 7 | Numero de ajuste, venta o compra. |
| Fecha | DATE | | Fecha en la que se realizó el movimiento |
| Tipo | VARCHAR | 30 | Descripción del tipo de movimiento realizado 1- Venta 2- Compra 3- Ajuste |
| Monto | INTEGER | 11 | Monto total del movimiento |

Descuento

Nombre Lógico: descuento

Clave principal: Codigo_Descuento

Descripción: Almacena el descuento actual de la tienda.

Tabla 2-16. Descuentos

| Nombre | Descuento | Longitud | Descripción |
|------------------|-----------|----------|-------------------------|
| Codigo_Descuento | SMARLLINT | 3 | Código del descuento |
| Total_Descuento | SMALLINT | 3 | porcentaje de descuento |

Medio de pago

Nombre Lógico: medio_de_pago

Descripción: Almacena los medios de pago que acepta la tienda

Clave principal:Codigo_Pago

Tabla 2-17. Medio de pago

| Nombre | Tipo | Longitud | Descripción |
|-------------|----------|----------|-------------------------|
| Codigo_Pago | SMALLINT | 3 | código del pago |
| Nombre | VARCHAR | 20 | Nombre del tipo de pago |

Ajuste Stock

Nombre Lógico: ajuste_stock

Descripción: Almacena registros sobre cualquier modificación en el stock de los productos.

Clave principal: ID_Ajuste + Codigo_Producto

Clave foránea: Codigo_Producto referencia tabla producto

Tabla 2-18. Ajuste de stock

| Nombre | Tipo | Longitud | Descripción |
|-----------|---------|----------|--------------------------|
| ID_Ajuste | INTEGER | 7 | Identificador del ajuste |

| | | | |
|-----------------|----------|----|---|
| Codigo_Producto | INTEGER | 7 | código del producto. |
| Cantidad | SMALLINT | 3 | Cantidad.del stock a aumentar o disminuir |
| Fecha | DATE | | Fecha del ajuste |
| Tipo | VARCHAR | 20 | Tipo de ajuste (entrada o salida) |
| Descripción | VARCHAR | 40 | Descripción del ajuste realizado |

CAPÍTULO 3:
DESCRIPCIÓN DE PROGRAMAS

Capítulo 3: Descripción de programas

3.1 Diagramas

A continuación, se señalan el diagrama modular y de menús, ambos muestran cómo es que funciona el sistema de una manera general.

3.1.1 Diagrama de menús

A continuación, en la figura 3-1 se muestra el diagrama de menús que contendrá el sistema el cual está dividido en 6 secciones que son Inicio, Venta, Compra, Mantenedores, Ajustes y Sistema.

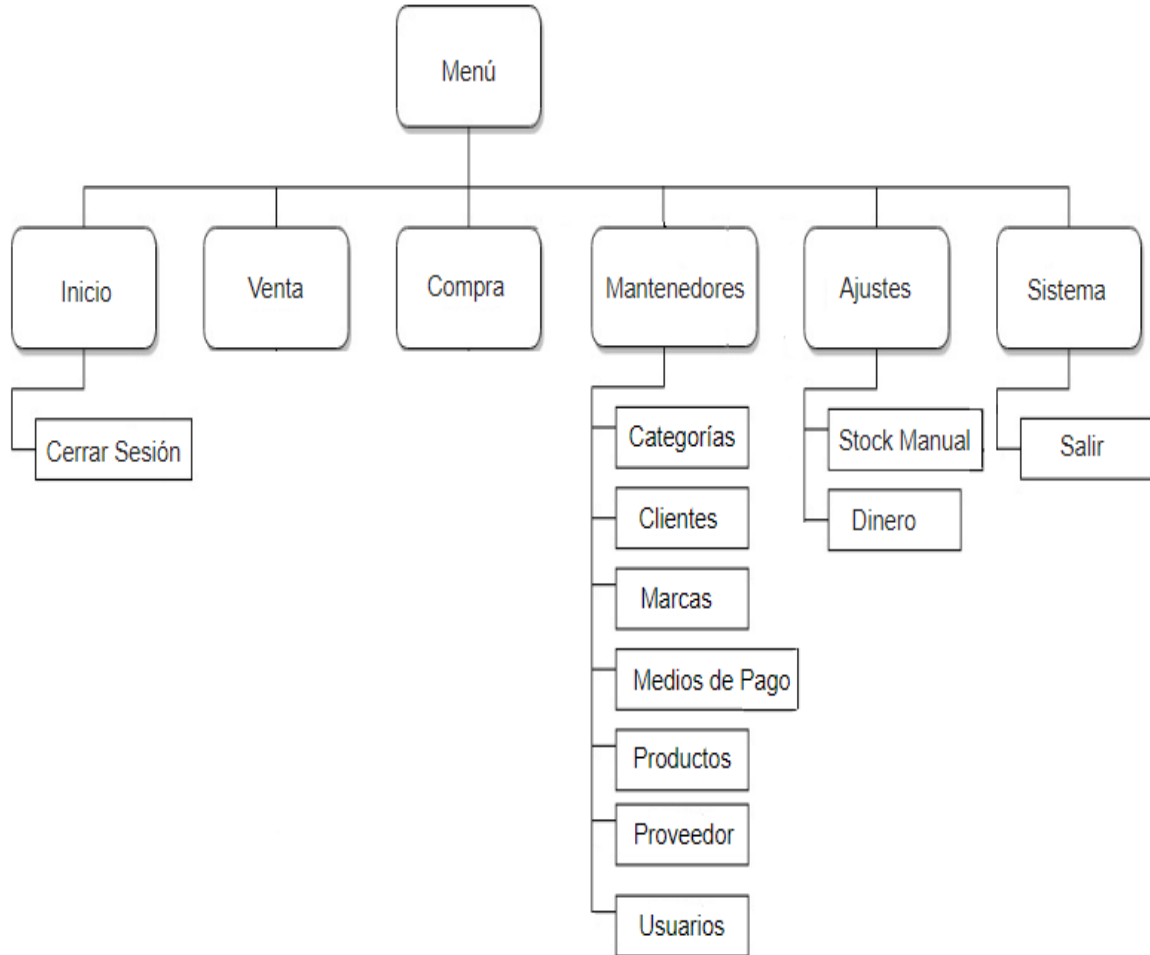


Figura 3-1. Diagrama de menú

3.1.2 Diagrama Modular

A continuación, en la figura 3-2 se muestra el diagrama general de los módulos principales del sistema.

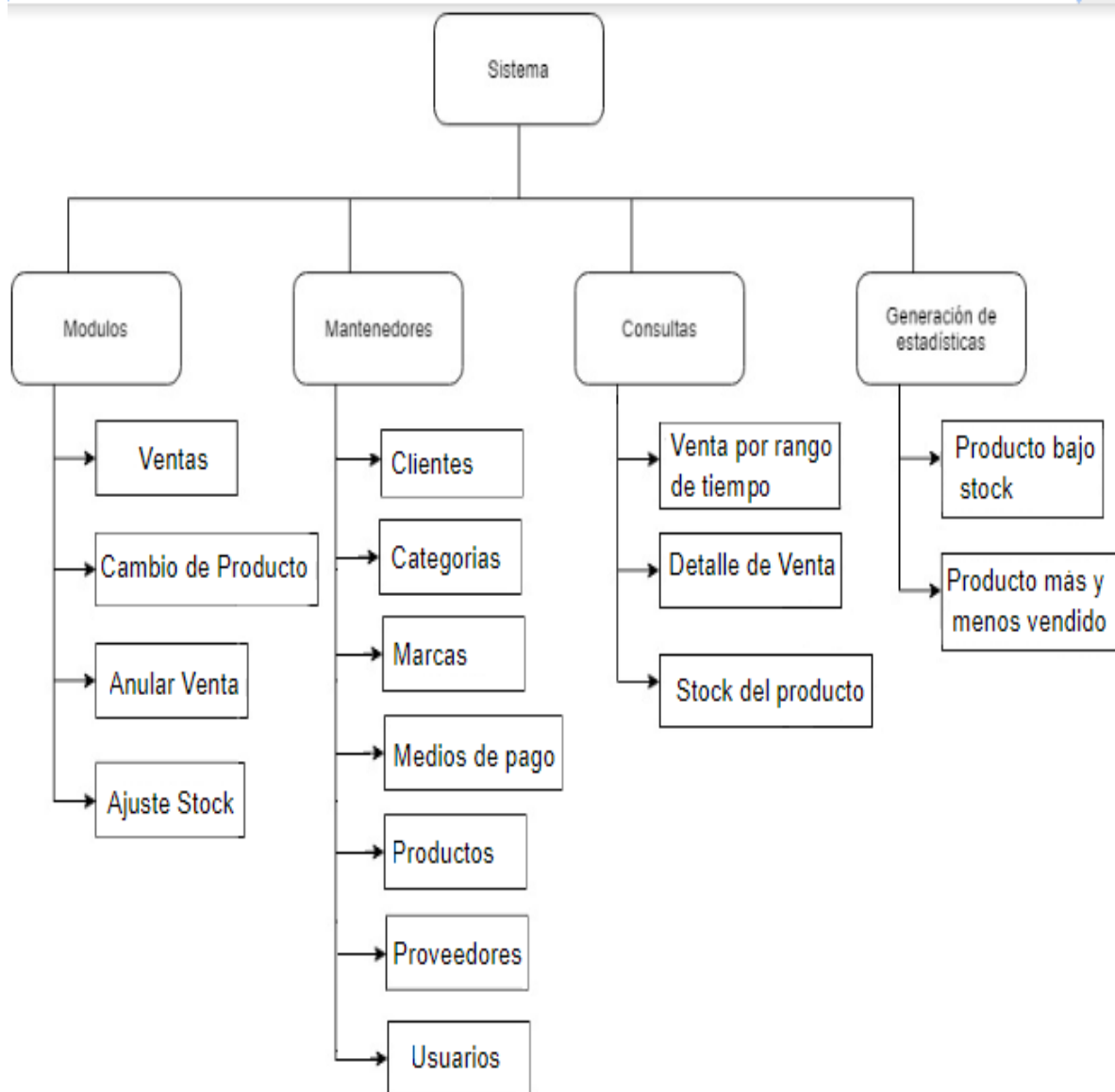


Figura 3-2. Diagrama Modular

3.2 Descripción de programas

El sistema realizar funciona en una pantalla principal de ingreso al programa, la cual es un formulario de Visual Basic, del cual dependen todos los otros formularios, lo que permite que al cerrar la aplicación principal se pueda garantizar que el sistema se cierre completamente, evitando así que alguna función quede cargada en el sistema operativo, para evitar fallas de seguridad.

En esta pantalla se puede acceder a la función de ingreso de usuario en el sistema, la cual, junto con autorizar el acceso a las funciones del programa, da acceso restringido a las mismas según el tipo de usuario registrado. En otras palabras, si un usuario no está registrado, no tiene acceso a las funciones del sistema, pero si un usuario se registra como Vendedor, tendrá acceso sólo a las funciones que le competen como vendedor, y si el usuario es del tipo Administrador, tendrá acceso completo a las funciones del sistema.

3.2.1 Tabla de programas

En la siguiente tabla se mostrarán todos los programas utilizados por el sistema con una breve descripción, los nombres de programas sucedidos con un * serán descritos detalladamente al final de la tabla.

Tabla 3-1, Tabla de programas

| Nombre | Descripción |
|---------------------------|--|
| Inicio * | Inicio de sesión, valida el ingreso de un usuario y otorga los privilegios según corresponda, también funciona como pantalla principal del sistema, contiene el menú con las funciones que este realiza. |
| Mantenedor categorías | Permite agregar, modificar, eliminar, consultar e imprimir un registro o registros en la base de datos categoría. |
| Mantenedor clientes | Permite agregar, modificar, eliminar, consultar e imprimir un cliente(s) de o en la base de datos cliente |
| Mantenedor marcas | Permite agregar, modificar, eliminar, consultar e imprimir un registro o registros en la base de datos marca. |
| Mantenedor medios de pago | Permite agregar, modificar, eliminar, consultar e imprimir un registro(s) en la base de datos de Medios de pago. |
| Mantenedor productos * | Permite agregar, modificar, eliminar, consultar e imprimir un producto o productos en la base de datos producto. |
| Mantenedor proveedores | Permite agregar, modificar, eliminar, consultar e imprimir un proveedor(s) de o en la tabla proveedor. |
| Mantenedor usuarios | Permite agregar, modificar, eliminar, consultar e imprimir un usuario(s) de o en la base de datos usuarios. |
| Venta* | Permite al sistema realizar una venta. |
| Devolución | Permite al sistema realizar una devolución de una venta. |
| Agregar compra | Permite al sistema agregar una nueva orden de compra. |
| Compras * | Permite al sistema Consultar órdenes de compra vigentes y actualizar el estado de estas una vez realizada la compra de los productos. |
| Productos bajo stock | Permite al sistema verificar la tabla productos y desplegar una lista con todos los productos que poseen un stock actual menor que el stock mínimo fijado para ese producto. |
| Ajuste stock * | Permite al sistema ingresar un registro en la tabla ajuste_stock en el cual se especifica el |

| | |
|---------------|--|
| | motivo, monto, stock del producto a ajustar, etc. Como así de ajustar este en la tabla del producto correspondiente. |
| Ajuste dinero | Permite al sistema ingresar un registro con respecto a un movimiento de dinero en la tabla movimientos. |

3.3 Detalle de programas

3.3.1 Inicio

Nombre: Inicio

Objetivo: Acceso de usuarios otorgando privilegios dependiendo del tipo de usuario, también funciona como formulario principal del sistema

Diagrama de bloque:



Figura 3-3, Diagrama de bloque para Inicio

Reglas de proceso:

El sistema despliega el formulario inicial del sistema donde el usuario debe ingresar su rut y contraseña (ver Figura 3-4).

Captura de pantalla de una interfaz de usuario web. La barra superior muestra un menú con opciones: Inicio, Venta, Compra, Mantenedores, Ajustes, Sistema, y la hora actual: 19:57:28. El formulario principal, titulado 'Login', contiene dos campos de texto: 'Rut Usuario:' y 'Contraseña:'. Debajo de estos campos hay dos botones: 'Aceptar' (con un borde punteado) y 'Salir'. En la esquina inferior izquierda del formulario, se encuentra el texto 'Bienvenido:'.

Figura 3-4, Inicio sesión

Al presionar el botón aceptar se valida que el usuario exista en la base de datos- tabla usuarios, Si los datos ingresados son incorrectos muestra un mensaje de error (ver Figura 3-5).

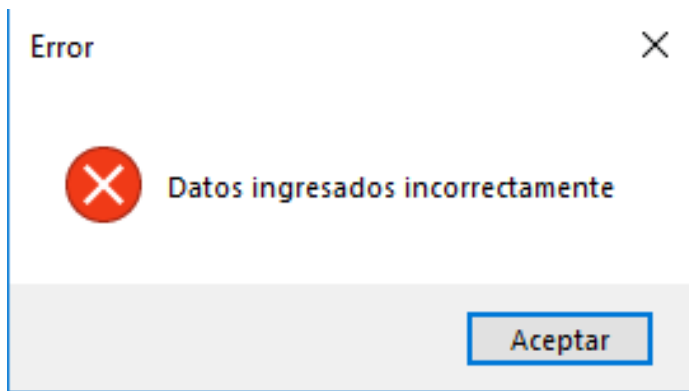


Figura 3-5 Error al iniciar sesión

Si los datos ingresados son válidos se habilitarán los privilegios del usuario ingresado (ver Figura 3-6).

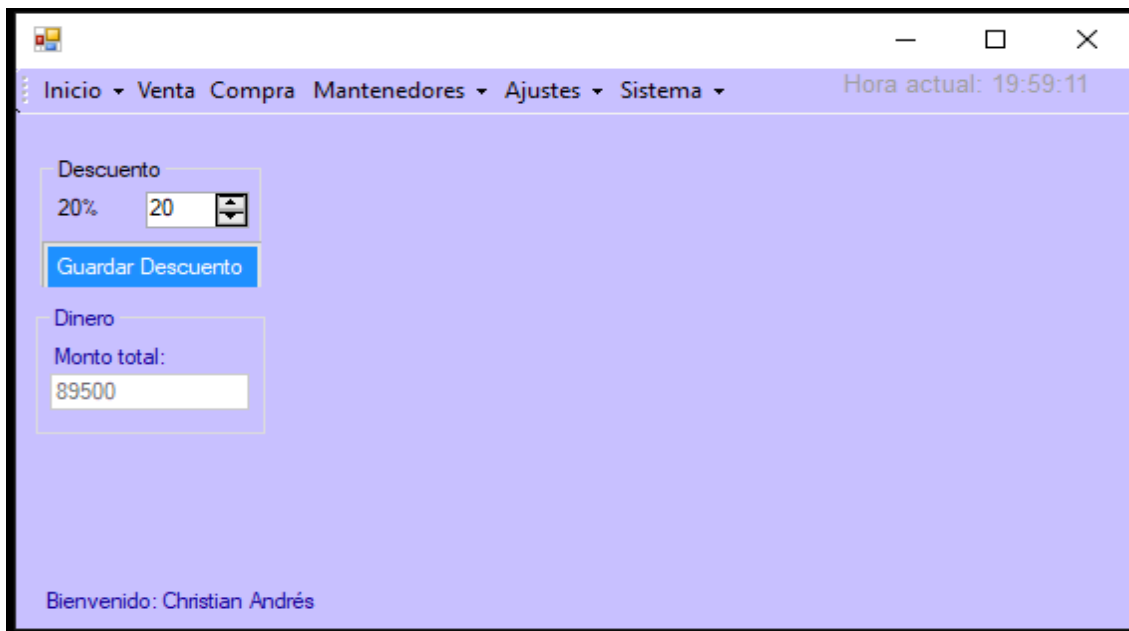


Figura 3-6, Formulario principal

En la parte superior posee el menú con las distintas funciones que tiene el sistema, en la central se muestra el dinero en caja y da la opción de aplicar un descuento general a las ventas.

Referencia código fuente: Pág. 66

3.3.2 Mantenedor de productos

Nombre: Mantenedor Productos

Objetivo: Consultar, agregar, eliminar o modificar productos.

Diagrama de bloque:

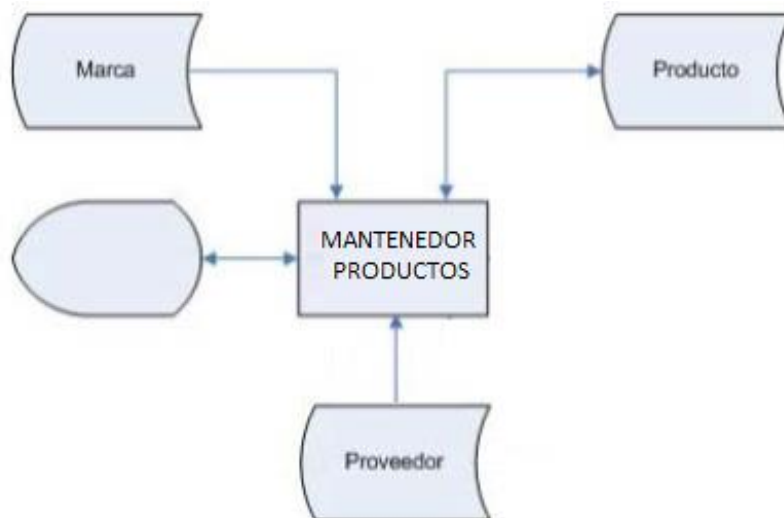


Figura 3-7, Diagrama de bloque para Mantenedor de productos

Reglas de proceso:

Este formulario es el encargado de mantener en la base de datos la tabla Productos, todas las acciones se realizan directamente sobre esta tabla. Al iniciar este programa se despliega un formulario en el cual se listarán todos los productos que actualmente están ingresados en la base de datos (ver Figura 3-8), se da la posibilidad de poder filtrar la búsqueda si se requiere un producto específico.

| Codigo_Producto | Codigo_Marca | Codigo_Categoria | Nombre | Precio_compra | Precio_Venta | Stock_Minimo | Stock_Actual | Descripcion |
|-----------------|--------------|------------------|----------------|---------------|--------------|--------------|--------------|----------------------|
| 0 | 8 | 4 | Led 24v | 5000 | 20000 | 4 | 3 | Luz frontal camion |
| 1 | 7 | 4 | Luz Xenon | 5000 | 15000 | 10 | 0 | KIT luz frontal a... |
| 2 | 3 | 1 | Aceite 22/50 | 5000 | 23000 | 10 | 0 | Aceite de caja p... |
| 3 | 5 | 3 | Neumatico 18xc | 5000 | 40000 | 4 | 0 | Neumatico auto... |
| 4 | 3 | 1 | aceite 40/50 | 5000 | 25000 | 10 | 7 | Aceite para motor |
| 5 | 4 | 2 | Llave de cruz | 5000 | 8000 | 10 | 7 | Llave de cruz pa... |
| 6 | 1 | 2 | Gata | 5000 | 60000 | 10 | 10 | Gata hidraulica |
| 7 | 10 | 6 | Pino | 5000 | 2000 | 10 | 6 | Pino aromatico... |
| 8 | 6 | 5 | filtro 1550 | 5000 | 9000 | 10 | 8 | filtro camionesta... |
| 9 | 6 | 5 | filtro 2222 | 5000 | 9000 | 10 | 9 | filtro de aire |
| 10 | 6 | 5 | filtro 678 | 5000 | 7500 | 10 | 4 | filtro de aceite |

Figura 3-8, Mantenedor de productos

Al presionar un producto de la lista la información de este será cargada (ver Figura 3-9) y se dará la opción de modificar algún dato si se requiere, o si es que desea eliminar algún producto, solo se podrá eliminar un producto que no esté en uso.

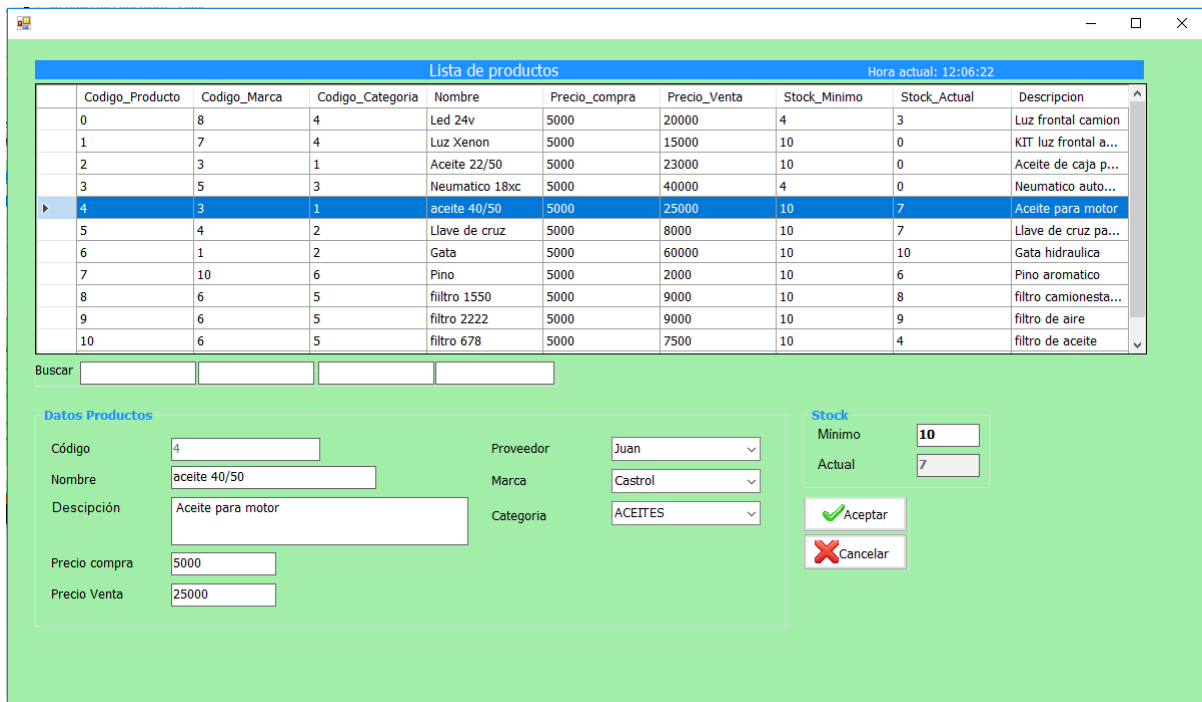


Figura 3-9, Mantenedor de productos con datos cargados

Para ingresar un nuevo producto el usuario debe presionar el botón ingresar para habilitar las opciones de ingresado, deberá ingresar manualmente los datos del producto, los datos marca y proveedor son cargados en sus respectivos selectores directamente desde sus tablas en las base de datos (ver figura 3-10).

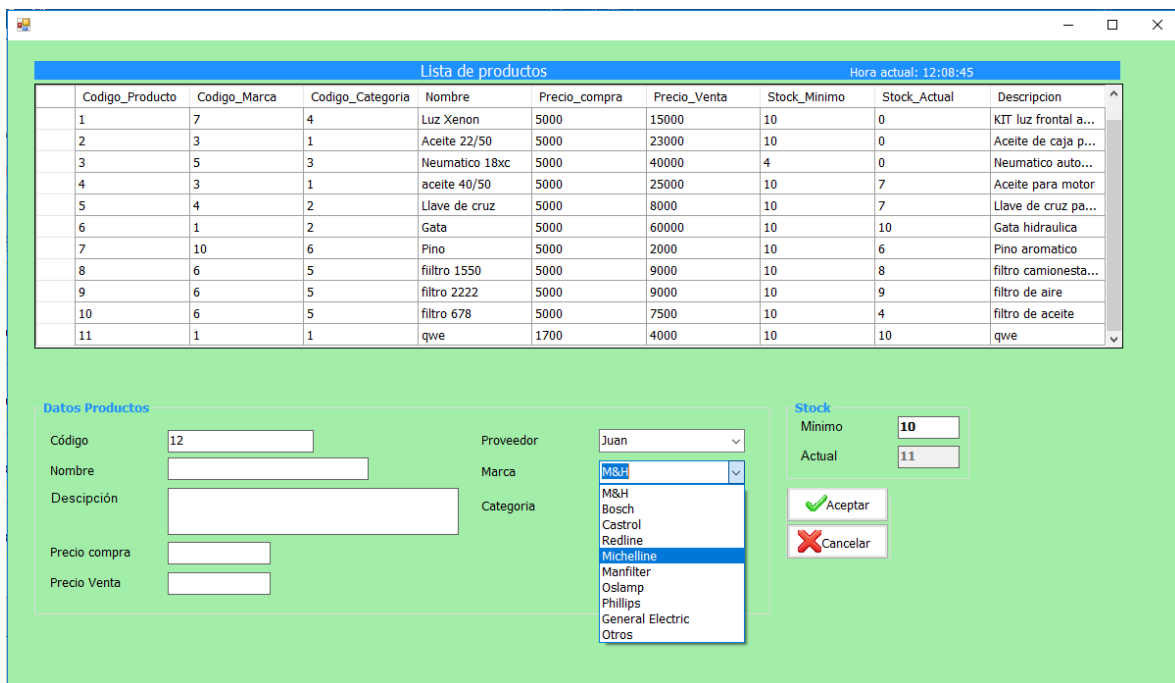


Figura 3-8, Mantenedor de productos ingresar

Referencia código fuente: Pág. 69

3.3.3 Venta

Nombre: Venta

Objetivo: Permitir al sistema realizar una venta

Diagrama de bloque:

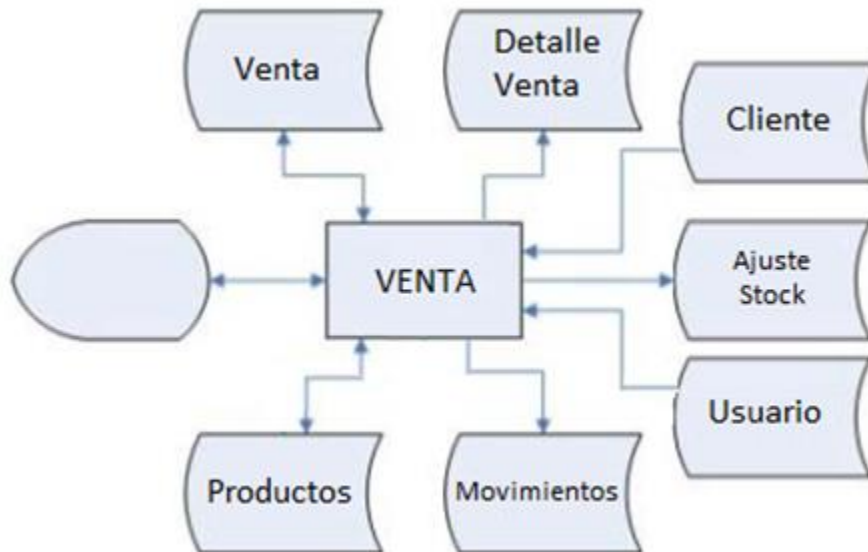


Figura 3-11, Diagrama de bloque para Venta

Reglas de proceso:

Este programa es el encargado de que el sistema pueda realizar ventas, al iniciar se despliega un formulario con los datos necesarios para realizar una venta (ver Figura 3-12),

en la parte izquierda el usuario debe ingresar el Rut del cliente, si este existe se cargaran sus datos correspondientes los datos del usuario que realiza son cargados automáticamente.

Vendedor
 Rut: 19048335
 Nombre: Emmanuel
 Apellido: Gamboa

Datos del Cliente
 Rut: Teléfono/Celular:
 Nombre: Dirección:
 Apellido:

Datos Productos
 Nombre:
 Descripción:
 Precio Unitario: 0
 Stock: 0
 Cantidad: 1
 Precio Actual: 0
 Metodo de Pago: EFECTIVO

Filtros
 Codigo: Marca:
 Nombre: Categoria:

| Lista de productos | | | | |
|--------------------|-----------------|----------------|--------------|--------------|
| | Codigo_Producto | Nombre | Precio_Venta | Stock_Actual |
| ▶ | 0 | Led 24v | 20000 | 3 |
| | 1 | Luz Xenon | 15000 | 0 |
| | 2 | Aceite 22/50 | 23000 | 0 |
| | 3 | Neumatico 18xc | 40000 | 0 |
| | 4 | aceite 40/50 | 25000 | 7 |
| | 5 | Llave de cruz | 8000 | 7 |

| Carrito de compras | | | | | |
|--------------------|-----------------|--------|-----------------|----------|--------------|
| | Codigo_Producto | Nombre | Precio_Unitario | Cantidad | Precio_Total |
| | | | | | |

Datos Venta
 Sub Total: 0
 Descuento: 10%
 Total: 0

Factura Boleta

Ingresar Venta Devolución Salir

Figura 3-12, Venta

En la parte superior derecha aparece el listado de productos disponibles para la venta, se da la opción de múltiples filtros con el fin de hacer que la lista se reduzca y encontrar más fácil lo que se busca. Al seleccionar un producto los datos son cargados (ver Figura 3-13).

Nº de venta 10 Registro de nueva venta Fecha: 22-03-2019 12:20:19

Vendedor
 Rut: 19048335
 Nombre: Emmanuel
 Apellido: Gamboa

Datos del Cliente
 Rut: 10400787
 Nombre: Pedro
 Apellido: Gutierrez
 Teléfono/Celular: 97742732
 Dirección: Eastman

Datos Productos
 Nombre: Llave de cruz
 Descripción: Llave de cruz para maletero
 Precio Unitario: 8000
 Stock: 7
 Cantidad: 2
 Precio Actual: 16000
 Metodo de Pago: EFECTIVO

Factura
 Boleta

Datos Venta
 Sub Total: 0
 Descuento: 10%
 Total: 0

Filtros
 Código: Marca:
 Nombre: Categoría:

Lista de productos

| Codigo_Producto | Nombre | Precio_Venta | Stock_Actual |
|-----------------|----------------|--------------|--------------|
| 0 | Led 24v | 20000 | 3 |
| 1 | Luz Xenon | 15000 | 0 |
| 2 | Aceite 22/50 | 23000 | 0 |
| 3 | Neumatico 18xc | 40000 | 0 |
| 4 | aceite 40/50 | 25000 | 7 |
| 5 | Llave de cruz | 8000 | 7 |

Carrito de compras

| Codigo_Producto | Nombre | Precio_Unitario | Cantidad | Precio_Total |
|-----------------|--------|-----------------|----------|--------------|
|-----------------|--------|-----------------|----------|--------------|

Figura 3-13, Venta con un producto cargado.

Una vez el producto es seleccionado el usuario debe ingresar la cantidad que el cliente desea comprar, al presionar agregar el producto y su respectiva cantidad son agregados al detalle de venta (ver figura 3-14), este paso es iterativo y de esta forma se llenan los productos que el cliente quiere comprar.

Nº de venta 10 Registro de nueva venta Fecha: 22-03-2019 12:22:19

Vendedor
Rut: 19048335
Nombre: Emmanuel
Apellido: Gamboa

Datos del Cliente
Rut: 10400787
Nombre: Pedro
Apellido: Gutierrez
Teléfono/Celular: 97742732
Dirección: Eastman

Datos Productos
Nombre: Pino
Descripción: Pino aromático
Precio Unitario: 2000
Stock: 6
Cantidad: 1
Precio Actual: 2000
Metodo de Pago: EFECTIVO

Factura
 Boleta

Datos Venta
Sub Total: 78000
Descuento: 10%
Total: 70200

Filtros
Codigo: Marca:
Nombre:

Lista de productos

| Codigo_Producto | Nombre | Precio_Venta | Stock_Actual |
|-----------------|----------------|--------------|--------------|
| 2 | Aceite 22/50 | 23000 | 0 |
| 3 | Neumatico 18xc | 40000 | 0 |
| 4 | aceite 40/50 | 25000 | 7 |
| 5 | Llave de cruz | 8000 | 7 |
| 6 | Gata | 60000 | 10 |
| 7 | Pino | 2000 | 6 |

Carrito de compras

| Codigo_Producto | Nombre | Precio_Unitario | Cantidad | Precio_Total |
|-----------------|---------------|-----------------|----------|--------------|
| 5 | Llave de cruz | 8000 | 2 | 16000 |
| 6 | Gata | 60000 | 1 | 60000 |
| 7 | Pino | 2000 | 1 | 2000 |

Figura 3-14, Venta Con detalle de venta

Una vez el detalle de venta está completo y la venta es validada, se guardarán los registros en la base de datos en las tablas venta y detalle venta, el stock de los productos vendidos será disminuido y se guardará el monto de la venta en la tabla movimientos, el sistema emite una factura (ver Figura 3-15) o boleta (ver Figura 3-16) según corresponda a la venta realizada.

Objetivo: Permite al sistema realizar compras de productos

Diagrama de bloque:

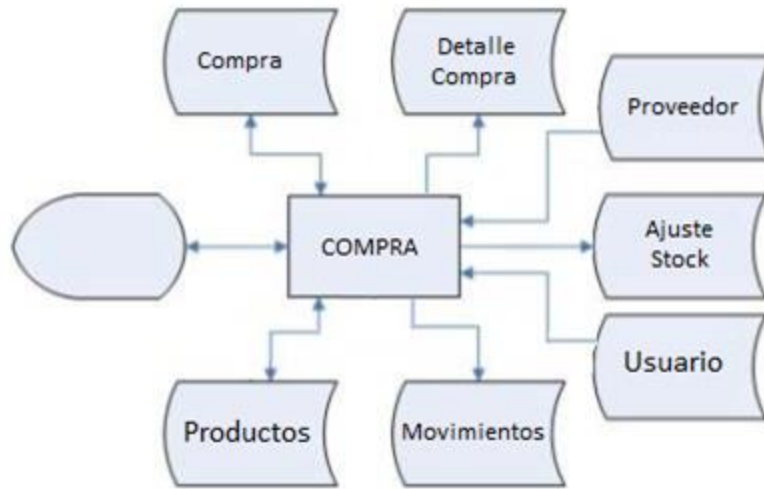


Figura 3-17, Diagrama de bloque para Compra

Reglas de proceso:

Este programa trabaja sobre los stocks de los productos, al iniciarlo se despliega un formulario donde se despliegan las ordenes de compras ya realizadas por el sistema (ver Figura 3-18).

Captura de pantalla de un formulario de compra. El formulario tiene un fondo verde claro y una barra superior con el título 'Compra'. A la izquierda hay un menú con opciones: 'Historico', 'En espera', 'Completado' y 'Cancelado'. Debajo de esto hay una tabla con el título 'Lista de Pedidos' que muestra los siguientes datos:

| | Nro_pedido | Rut_proveedor | Fecha | Estado | Valor |
|---|------------|---------------|------------|------------|-------|
| ▶ | 1 | 8784327 | 06-03-2019 | Cancelado | 50000 |
| | 2 | 8784327 | 06-03-2019 | Cancelado | 20000 |
| | 3 | 8784327 | 06-03-2019 | Completado | 20000 |
| | 4 | 12684549 | 06-03-2019 | En espera | 70000 |
| | 5 | 8784327 | 06-03-2019 | Completado | 15000 |

Debajo de la tabla hay una barra de herramientas con los botones: 'Agregar', 'Completar', 'Cancelar', 'Eliminar', 'Revisar' y 'Imprimir'. A la derecha del formulario hay un panel con el título 'Detalle pedido' que muestra un formulario con los campos: 'Nro_Pedido', 'codigo_producto' y 'cantidad'. En la parte superior derecha del panel se muestra 'Hora actual: 12:37:07'. En la parte inferior del formulario hay un formulario para el 'Orden de compra' con los campos: 'Código', 'Proveedor', 'Fecha', 'Estado' y 'Monto a pagar'. También hay un formulario para el 'Usuario' con los campos: 'Rut' (valor: 19048335), 'Nombre' (valor: Emmanuel) y 'Apellido' (valor: Gamboa). En la parte inferior derecha del formulario hay un botón 'Salir'.

Figura 3-18, Compra

Al seleccionar una de ellas se despliega el detalle de la misma (ver Figura 3-19), también se da la opción de cambiar el estado de dicha orden, al cambiar una orden a estado

completado, se actualiza el stock de los productos comprados, se guarda un registro por cada producto en la tabla de ajustes de stock indicando la cantidad que se compró y se guarda un registro del monto gastado en la tabla de movimientos.

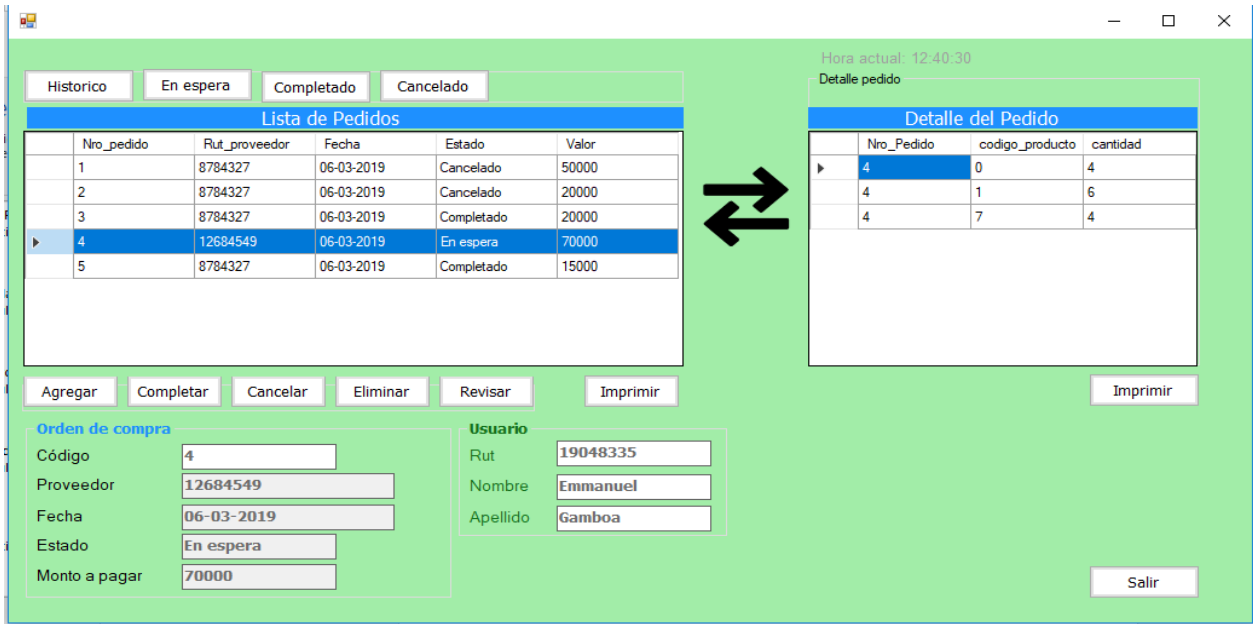


Figura 3-19, Compra con Detalle

Para ingresar una nueva compra el usuario debe presionar el botón agregar, se desplegará una nueva ventana (ver figura 3-20).

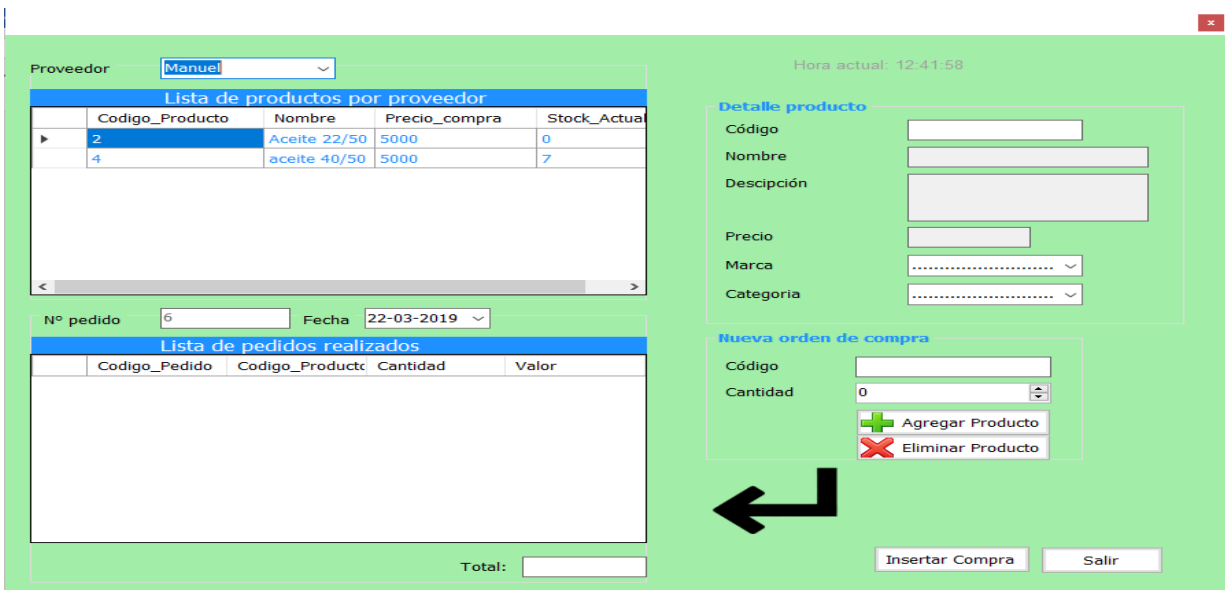


Figura 3-20, Ingresar Compra

El usuario debe seleccionar el proveedor a quien se le está comprando, una vez seleccionado se cargarán los productos que vende el proveedor, el usuario deberá ir

seleccionando el producto, especificar la cantidad y agregarlos al detalle de compra (Ver Figura 3-21).

Proveedor: Manuel

Hora actual: 12:52:10

Lista de productos por proveedor

| Codigo_Producto | Nombre | Precio_compra | Stock_Actual |
|-----------------|--------------|---------------|--------------|
| 2 | Aceite 22/50 | 5000 | 0 |
| 4 | aceite 40/50 | 5000 | 7 |

Detalle producto

Código: 4
Nombre: aceite 40/50
Descripción: Aceite para motor
Precio: 5000
Marca: Castrol
Categoría: ACEITES

Nº pedido: 6 Fecha: 22-03-2019

Lista de pedidos realizados

| Codigo_Pedido | Codigo_Producto | Cantidad | Valor |
|---------------|-----------------|----------|-------|
| 6 | 2 | 5 | 25000 |
| 6 | 4 | 10 | 50000 |

Nueva orden de compra

Código: 4
Cantidad: [Spinner]
+ Agregar Producto
- Eliminar Producto

Total: 75000

Insertar Compra Salir

Figura 3-21, Compra con Detalle de compra

Al aceptar la compra se guardarán los registros correspondientes en las tablas compra y detalle de compra en la base de datos, el estado de la nueva compra será "En espera".

Referencia código fuente: 99

3.3.5 Ajuste de Stock

Nombre: Ajustes

Objetivo: Permite al sistema ver los movimientos de stock de productos y ajustarlo si es necesario.

Diagrama de bloque:

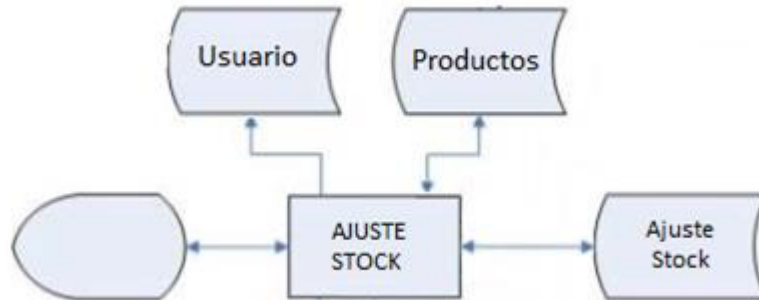


Figura 3-22, Diagrama de bloque de Ajuste de Stock

Reglas de proceso:

Este programa permite al sistema ajustar el stock de un producto, al ingresar se despliega un formulario en él están listados todos los movimientos de productos que se han hecho se da la posibilidad de filtrar la búsqueda con el fin de acotar la lista (ver Figura 3-23).

Lista de productos con bajo stock

| Codigo_Producto | Nombre | Stock_Minimo | Stock_Actual |
|-----------------|----------------|--------------|--------------|
| 0 | Led 24v | 4 | 3 |
| 1 | Luz Xenon | 10 | 0 |
| 2 | Aceite 22/50 | 10 | 0 |
| 3 | Neumatico 18xc | 4 | 0 |
| 4 | aceite 40/50 | 10 | 7 |
| 5 | Llave de cruz | 10 | 7 |
| 7 | Pino | 10 | 6 |
| 8 | filtro 1550 | 10 | 8 |
| 9 | filtro 2222 | 10 | 9 |
| 10 | filtro 678 | 10 | 4 |

Hora actual: 13:36:33

Codigo Producto:

Datos producto

Código producto:

Cantidad:

Fecha:

Tipo:

Descripcion:

Salir

Agregar Eliminar Imprimir

Lista de ajustes de stock

| Codigo_producto | Cantidad | fecha | tipo | descripcion |
|-----------------|----------|---------------------|---------|------------------------------|
| 4 | -2 | 06-03-2019 17:03:09 | salida | venta Nro: 9 |
| 5 | -1 | 06-03-2019 17:03:09 | salida | venta Nro: 9 |
| 9 | 3 | 06-03-2019 22:44:11 | entrada | entrada por orden de comp... |
| 1 | -2 | 15-03-2019 22:46:47 | salida | venta Nro: 9 |
| 2 | -1 | 15-03-2019 22:46:47 | salida | venta Nro: 9 |
| 3 | -2 | 15-03-2019 22:50:19 | salida | venta Nro: 9 |

Figura 3-23, Ajuste de Stock

El usuario con privilegios de administrador puede ajustar el stock de un producto para ellos de ingresar los datos manualmente en el formulario, al aceptar los datos se guardará un

registro en la tabla de ajuste de stock y se actualizará el stock del producto en la tabla producto (ver Figura 3-24), desde esta pantalla también se podrá imprimir un informe de productos con bajo stock (ver Figura 3-25).

| Lista de productos con bajo stock | | | | |
|-----------------------------------|-----------------|----------------|--------------|--------------|
| | Codigo_Producto | Nombre | Stock_Minimo | Stock_Actual |
| | 0 | Led 24v | 4 | 3 |
| ▶ | 1 | Luz Xenon | 10 | 0 |
| | 2 | Aceite 22/50 | 10 | 0 |
| | 3 | Neumatico 18xc | 4 | 0 |
| | 4 | aceite 40/50 | 10 | 7 |
| | 5 | Llave de cruz | 10 | 7 |
| | 7 | Pino | 10 | 6 |
| | 8 | filtro 1550 | 10 | 8 |
| | 9 | filtro 2222 | 10 | 9 |
| | 10 | filtro 678 | 10 | 4 |

Hora actual: 14:00:23

Codigo Producto

Datos producto

Código producto

Cantidad

Fecha

Tipo

Descripcion

| Lista de ajustes de stock | | | | | |
|---------------------------|-----------------|----------|---------------------|---------|------------------------------|
| | Codigo_producto | Cantidad | fecha | tipo | descripcion |
| | 4 | -2 | 06-03-2019 17:03:09 | salida | venta Nro: 9 |
| ▶ | 5 | -1 | 06-03-2019 17:03:09 | salida | venta Nro: 9 |
| | 9 | 3 | 06-03-2019 22:44:11 | entrada | entrada por orden de comp... |
| | 1 | -2 | 15-03-2019 22:46:47 | salida | venta Nro: 9 |
| | 2 | -1 | 15-03-2019 22:46:47 | salida | venta Nro: 9 |
| | 3 | -2 | 15-03-2019 22:50:19 | salida | venta Nro: 9 |

Figura 3-24, Ingresar Ajuste de Stock

INFORME DE AJUSTES DE STOCK

FECHA DE IMPRESIÓN: 22-03-2019

HORA DE IMPRESIÓN: 14:01:17

| Codigo_producto | Cantidad | Fecha | tipo | descripcion |
|-----------------|----------|---------------------|---------|---------------------------|
| 4 | -2 | 05-03-2019 17:03:09 | salida | venta No. 9 |
| 5 | -1 | 05-03-2019 17:03:09 | salida | venta No. 9 |
| 9 | 3 | 05-03-2019 22:44:11 | entrada | entrada por orden de c.c. |
| 1 | -2 | 15-03-2019 22:45:47 | salida | venta No. 9 |
| 2 | -1 | 15-03-2019 22:45:47 | salida | venta No. 9 |
| 3 | -2 | 15-03-2019 22:50:19 | salida | venta No. 9 |

Figura 3-25, Ajuste de Stock

Referencia código fuente: Pág. 106

Conclusiones

El desarrollo del presente Sistema de Información dejó diversas apreciaciones en las distintas etapas que lo componen, las cuales procedemos a mencionar a continuación.

En la etapa de análisis se pudo concluir que es la etapa más compleja de la construcción de un sistema, pues acá uno como analista se ve enfrentado a los requerimientos del cliente, y se debe revisar constantemente lo que realmente necesita, ya que no siempre es claro a expresar sus ideas, siempre aparecen nuevas ideas lo que genera requerimientos a futuro.

Comenzar a construir un sistema desde el punto de vista analítico es un proceso sumamente complejo y difícil de visualizar, debido a que en este punto comienza la interpretación del problema y el diseño de la solución informática con el fin de satisfacer las necesidades del usuario, y desde este punto se empieza a analizar la solución ideal al problema planteado, aspecto que toma bastante tiempo llevarse a cabo, y se somete a constantes revisiones y cambios.

En la etapa de diseño, la construcción de la base de datos es un proceso sumamente importante en el desarrollo de las soluciones informáticas, esta es susceptible a cambios por parte del usuario debido a nuevos requerimientos o por la generación de nuevas tablas por proceso de normalización. Se pudo apreciar que esta es la etapa más lenta por las constantes modificaciones y actualizaciones, sin duda la parte más crítica del sistema, ya que, de esta depende el almacenamiento de los datos.

En la etapa de construcción, se pudo concluir que, si bien es un proceso complejo, es relativamente más sencillo comparado con las etapas anteriores, a medida que se construye el sistema nuevas necesidades van apareciendo por lo que es necesario reevaluar las etapas anteriores, se puede concluir que una mala etapa de análisis genera una carga de trabajo adicional.

Finalmente, ante todo lo anterior expuesto se puede concluir que el desarrollo de un sistema de información es un proceso bastante complejo y este necesita de tiempo para su

óptimo desarrollo. Se requirió de bastante trabajo en equipo enfocado en el análisis ya que este punto es vital para el desarrollo de sistemas de calidad.

Bibliografía

- <https://social.msdn.microsoft.com/Forums/es-ES/home?forum=vbes>

Sito oficial de Microsoft donde se presta ayuda a programadores.

- <https://www.lawebdelprogramador.com/foros/Visual-Basic.NET/index1.html>

Sitio donde programadores comparten experiencias.

- <https://programacion.net/foros/visual-basic>

Sitio de debate para programadores.

- <https://www.connectionstrings.com/mysql>

Sitio que brinda ayuda con respecto a conexiones con los distintos motores de base de datos

- <https://mariadb.org/>

Sitio oficial de MariaDB.

Anexo: Código Fuente

1 -Formulario Inicio

```
Imports MySql.Data.MySqlClient
Public Class FrmPrincipal
    Dim BsnNegocios As New BsnNegocios
    Dim st As String

    Private Sub CerrarSesionToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
CerrarSesionToolStripMenuItem.Click
        GbLogin.Visible = True
```

```

    ToolStrip1.Enabled = False
    GbDescuento.Visible = False
    GBdinero.Visible = False
    txtRut.Text = ""
    txtContraseña.Text = ""
    lblNombre.Text = "Bienvenido: "
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    lblFechaYHora.Text = "Hora actual: " & Date.Now.ToLongTimeString
End Sub

Private Sub ProductosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles mantenedores.Click
    FrmProductos.Show()
    Me.Hide()
End Sub

Private Sub MarcaToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
IngresarNuevaCategoriaToolStripMenuItem.Click
    FrmMarca.Show()
    Me.Hide()
End Sub

Private Sub CategoriaToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
IngresarNuevoTipoToolStripMenuItem.Click
    FrmCategoria.Show()
    Me.Hide()
End Sub

Private Sub CLientesToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles InsertarConsultaToolStripMenuItem.Click
    FrmClientes.Show()
    Me.Hide()
End Sub

Private Sub UsuariosToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ModificarConsultaToolStripMenuItem.Click
    FrmUsuarios.Show()
    Me.Hide()
End Sub

Private Sub ConsultarProductos_Click(sender As Object, e As EventArgs) Handles ConsultarProductos.Click
    FrmProveedor.Show()
    Me.Hide()
End Sub

Private Sub Compra_Click(sender As Object, e As EventArgs) Handles ToolStripLabel6.Click
    FrmCompraRealizarOrden.Show()
    Me.Hide()
End Sub

Private Sub FrmPrincipal_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    GbDescuento.Visible = True
    ToolStrip1.Enabled = False
    GbDescuento.Visible = False
    GBdinero.Visible = False
    lblNombre.Enabled = False
    ToolStrip1.BackColor = Me.BackColor
    lblFechaYHora.Enabled = False
    TxTcaja.Text = BsnNegocios.getMontoDinero
End Sub

Function BuscarDescuento()

```

```

Dim conn As MySqlConnection
conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
Dim username As Boolean = True
conn.Open()
Dim sqlquery As String = "select Total_Descuento from descuentos WhereCodigo_Descuento = " & 1 & ""
Dim adapter As New MySqlDataAdapter
Dim command As New MySqlCommand
command.CommandText = sqlquery
command.Connection = conn
Dim registro As MySqlDataReader = command.ExecuteReader
If registro.Read Then
    NumericUpDown1.Value = registro("Total_Descuento")
    lbldescuento.Text = NumericUpDown1.Value & "%"
End If
conn.Close()
End Function

Private Sub ToolStripLabel5_Click(sender As Object, e As EventArgs) Handles ToolStripLabel5.Click
    FrmVenta.Show()
    Me.Hide()
End Sub

Private Sub MediosDePagoToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
MediosDePagoToolStripMenuItem.Click
    FrmMediosDePago.Show()
    Me.Hide()
End Sub

Private Sub BtnAceptar_Click(sender As Object, e As EventArgs) Handles BtnAceptar.Click
    Dim sw As Integer = 0
    If txtRut.Text Is "" Then
        MessageBox.Show("Debe ingresar un rut para iniciar sesion", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        sw = sw + 1
        If txtContraseña.Text Is "" Then
            MessageBox.Show("Debe ingresar una contraseña para iniciar sesion", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        Else
            Dim conn As MySqlConnection
            sw = sw + 1
            conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
            Dim username As Boolean = True
            conn.Open()
            Dim sqlquery As String = "select Rut_Usuario,Nombre,Apellido,Contrasena,Nivel_Acceso from usuarios Where Rut_Usuario
= " & txtRut.Text & ""
            Dim adapter As New MySqlDataAdapter
            Dim command As New MySqlCommand
            command.CommandText = sqlquery
            command.Connection = conn
            Dim registro As MySqlDataReader = command.ExecuteReader
            If registro.Read Then
                If registro("Contrasena") = txtContraseña.Text Then
                    Dim principal As New FrmPrincipal
                    TextBox1.Text = registro("Rut_Usuario")
                    TextBox2.Text = registro("Nombre")
                    TextBox3.Text = registro("Apellido")
                    TextBox4.Text = registro("Nivel_Acceso")
                    sw = sw + 1
                    GbLogin.Visible = False
                    BuscarDescuento()
                    lblNombre.Text = "Bienvenido: " + TextBox2.Text
                    If Integer.Parse(TextBox4.Text) = 1 Then
                        ToolStrip1.Enabled = True
                        ToolStripLabel6.Enabled = False
                    End If
                End If
            End If
        End If
    End If
End Sub

```

```

        ToolStripLabel2.Enabled = False
        ToolStripDropDownButton2.Enabled = False

    Else
        ToolStrip1.Enabled = True
        ToolStripLabel6.Enabled = True
        ToolStripLabel2.Enabled = True
        ToolStripDropDownButton2.Enabled = True
        GbDescuento.Visible = True
        GBdinero.Visible = True
    End If
    conn.Close()
End If
End If
conn.Close()
End If

End If
If sw = 2 Then
    MessageBox.Show("Datos ingresados incorrectamente", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
End Sub

Private Sub BtnSalir_Click(sender As Object, e As EventArgs) Handles BtnSalir.Click
    Application.Exit()
End Sub

Private Sub ModificarDatosClienteToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ModificarDatosClienteToolStripMenuItem.Click
    Application.Exit()
End Sub

Private Sub ToolStripDropDownButton1_Click(sender As Object, e As EventArgs)
    FrmAjustarStock.Show()
    Me.Hide()
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim descuento As Integer = 0
    Dim descuentos As New ClsDescuento
    descuentos.Descuento = NumericUpDown1.Value
    BsnNegocios.UpdateDescuento(descuentos)
    BuscarDescuento()
End Sub

Private Sub NumericUpDown1_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown1.ValueChanged
    lbldescuento.Text = NumericUpDown1.Value & "%"
End Sub

Private Sub ToolStripMenuItem1_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem1.Click
    FrmAjustarStock.Show()
    Me.Hide()
End Sub

Private Sub ToolStripMenuItem2_Click(sender As Object, e As EventArgs) Handles ToolStripMenuItem2.Click
    FrmAjusteDinero.Show()
    Me.Hide()
End Sub

Private Sub ToolStripLabel2_Click(sender As Object, e As EventArgs) Handles ToolStripLabel2.Click
End Sub
End Class

```

2- MANTENEDOR DE PRODUCTOS

```
Imports System.IO
Imports System.Data.OleDb
Imports MySql.Data.MySqlClient
Public Class FrmProductos
    Dim BsnNegocios As New BsnNegocios
    Dim sw As Integer
    Dim marca As String
    Dim categoria As String
    Dim proveedor As String
    Dim pagenumber As Integer = 1
    Dim mRow As Integer = 0
    Dim comodin As String
    Private Sub FrmProductos_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        GbProducto.Enabled = False
        GbStock.Enabled = False
        Me.BackColor = FrmPrincipal.BackColor
        CargarProductos()
        CargarComboBox()
        sw = 0
        Visibilidad()
    End Sub

    Function CargarProductos() ' ***** CARGAR PRODUCTOS EN EL DATASET
        DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
        Dim conn As MySqlConnection
        conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
        Dim username As Boolean = True
        conn.Open()
        Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos "
        Dim adapter As New MySqlDataAdapter
        Dim command As New MySqlCommand
        command.CommandText = sqlquery
        command.Connection = conn
        adapter.SelectCommand = command
        adapter.Fill(DataSet1.DataTableProducto)
        conn.Close()
    End Function

    Function CargarComboBox()
        CmbCategoria.DataSource = BsnNegocios.GetCategoria()
        CmbCategoria.DisplayMember = "Nombre"
        CmbCategoria.ValueMember = "Codigo_Categoria"
        CmbMarca.DataSource = BsnNegocios.GetMarca
        CmbMarca.DisplayMember = "Nombre"
        CmbMarca.ValueMember = "Codigo_Marca"
        CmbProveedor.DataSource = BsnNegocios.GetProveedor()
        CmbProveedor.DisplayMember = "Nombre"
    End Function

    Function Visibilidad()
        If sw = 0 Then
            GbBotones.Visible = True
            GbAceptarCancelar.Visible = False
            TxtBuscarCodigoProducto.Visible = False
            GBBuscar.Visible = False
            BtnImprimir.Visible = False
        ElseIf sw = 1 Then ' ***** AGREGAR
            GbBotones.Visible = False
            GbAceptarCancelar.Visible = True
        End If
    End Function
End Class
```

```

    GbProducto.Enabled = True
    GbStock.Enabled = True
Elseif sw = 2 Then ' ***** MODIFICAR
    GbBotones.Visible = False
    GbAceptarCancelar.Visible = True
    GbProducto.Enabled = True
    TxtCodProducto.Enabled = False
    GbStock.Enabled = True
    TxtStockActual.Enabled = False
    GBBuscar.Visible = True
Elseif sw = 3 Then ' ***** ELIMINAR
    GbBotones.Visible = False
    GbAceptarCancelar.Visible = True
    TxtCodProducto.Enabled = False
    GbProducto.Enabled = False
    GBBuscar.Visible = True
Elseif sw = 4 Then ' ***** CONSULTAR
    GbBotones.Visible = False
    GbAceptarCancelar.Visible = True
    GbProducto.Enabled = False
    GBBuscar.Visible = True
    TxtBuscarCodigoProducto.Visible = True
    BtnImprimir.Visible = True
Elseif sw = 5 Then ' ***** SALIR
    GbBotones.Visible = True
    GbAceptarCancelar.Visible = False
    TxtCodProducto.Enabled = True
    GbProducto.Enabled = True
    GbProducto.Enabled = False
    GbStock.Enabled = True
    GBBuscar.Visible = False
    TxtBuscarCodigoProducto.Text = ""
    TxtBuscarCodigoMarca.Text = ""
    TxtBuscarCodigoCategoria.Text = ""
    TxtPrecioCompra.Text = ""
    BtnImprimir.Visible = False
    CargarProductos()
    CargarComboBox()
End If
End Function

Function Limpiar()
    TxtCodProducto.Text = ""
    TxtNombre.Text = ""
    TxtDescripcionProducto.Text = ""
    TxtPrecio.Text = ""
    TxtStockMinimo.Text = 10
    TxtStockActual.Text = 11
End Function

Public Function IsDataGridViewEmpty(ByRef dataGridView As DataGridView) As Boolean
    Dim isEmpty As Boolean
    isEmpty = True
    For Each row As DataGridViewRow In dataGridView.Rows
        For Each cell As DataGridViewCell In row.Cells
            If Not String.IsNullOrEmpty(cell.Value) Then
                If Not String.IsNullOrEmpty(Trim(cell.Value.ToString())) Then
                    isEmpty = False
                Exit For
            End If
        End If
    Next
    Return isEmpty

```



```

        End If
    Else
        MsgBox.Show("El PRODUCTO DEBE TENER UN PRECIO DE VENTA!", "Error", MessageBoxButtons.OK,
MsgBoxIcon.Error)
    End If
    Else
        MsgBox.Show("El PRODUCTO DEBE TENER UN PRECIO DE COMPRA!", "Error", MessageBoxButtons.OK,
MsgBoxIcon.Error)
    End If
    Else
        MsgBox.Show("DEBE INGRESAR UN NOMBRE PARA EL PRODUCTO!", "Error", MessageBoxButtons.OK,
MsgBoxIcon.Error)
    End If
    End If
    Else
        MsgBox.Show("EL CODIGO DEBE SER MAYOR A 0!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    Else
        If TxtCodProducto.Text = "" Then
            MsgBox.Show("DEBE INGRESAR UN CODIGO!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    End If
    ElseIf sw = 2 Then
        If TxtCodProducto.Text IsNot "" Then
            If TxtNombre.Text IsNot "" Then
                If TxtPrecioCompra.Text IsNot "" Then
                    If TxtPrecio.Text IsNot "" Then
                        If Integer.Parse(TxtPrecio.Text) > Integer.Parse(TxtPrecioCompra.Text) Then
                            If categoria IsNot "" Then
                                If marca IsNot "" Then
                                    If TxtStockActual IsNot "" Then
                                        If TxtStockMinimo IsNot "" Then
                                            If Integer.Parse(TxtStockActual.Text) > Integer.Parse(TxtStockMinimo.Text) Then
                                                If MsgBox.Show("ESTA SEGURO DE MODIFICAR ESTE PRODUCTO?", "Error",
MsgBoxButtons.YesNo, MessageBoxIcon.Error) = DialogResult.Yes Then
                                                    Dim producto As New ClsProducto
                                                    producto.Codigo = TxtCodProducto.Text
                                                    producto.Codigo_Categoria = categoria
                                                    producto.Codigo_marca = marca
                                                    producto.Nombre = TxtNombre.Text
                                                    producto.PrecioCompra = TxtPrecioCompra.Text
                                                    producto.Precio = TxtPrecio.Text
                                                    producto.Stock_minimo = TxtStockMinimo.Text
                                                    producto.Stock_actual = TxtStockActual.Text
                                                    producto.Descripcion = TxtDescripcionProducto.Text
                                                    BsnNegocios.UpdateProducto(producto)
                                                    BsnNegocios.DeleteProductoProveedor(TxtCodProducto.Text)
                                                    BsnNegocios.InsertarProductoProveedor(TxtCodProducto.Text, proveedor)
                                                    Limpiar()
                                                    CargarProductos()
                                                    sw = 5
                                                    Visibilidad()
                                                End If
                                            Else
                                                MsgBox.Show("EL STOCK ACTUAL DEL PRODUCTO DEBE SER MAYOR QUE EL MINIMO", "Error",
MsgBoxButtons.OK, MessageBoxIcon.Error)
                                            End If
                                        Else
                                            MsgBox.Show("DEBE SELECCIONA UN STOCK MINIMO PARA EL PRODUCTO!", "Error",
MsgBoxButtons.OK, MessageBoxIcon.Error)
                                        End If
                                    Else
                                        Else

```

```

                MessageBox.Show("DEBE SELECCIONAR UN STOCK PARA EL PRODUCTO ACTUAL!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End If
        Else
            MessageBox.Show("DEBE SELECCIONAR UNA MARCA PARA EL PRODUCTO!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End If
    Else
        MessageBox.Show("DEBE SELECCIONAR UNA CATEGORIA PARA EL PRODUCTO!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End If
    Else
        MessageBox.Show("EL PRECIO DE VENTA DEL PRODUCTO DEBE SER MAYOR QUE EL PRECIO DE COMPRA", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    Else
        MessageBox.Show("EL PRODUCTO DEBE TENER UN PRECIO DE VENTA!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End If
    Else
        MessageBox.Show("EL PRODUCTO DEBE LLEVAR UN PRECIO DE COMPRA!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End If
    Else
        MessageBox.Show("DEBE INGRESAR UN NOMBRE PARA EL PRODUCTO!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End If
    Else
        MessageBox.Show("DEBE SELECCIONAR UN PRODUCTO A MODIFICAR DE LA LISTA!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)

    End If

    ElseIf sw = 3 Then
        If TxtCodProducto.Text IsNot "" Then
            If MessageBox.Show("ESTA SEGURO DE ELIMINAR ESTE PRODUCTO?", "Error", MessageBoxButtons.YesNo,
MessageBoxIcon.Error) = DialogResult.Yes Then
                Try
                    Dim producto As New ClsProducto
                    producto.Codigo = TxtCodProducto.Text
                    BsnNegocios.EliminarProducto(producto)
                    Limpiar()
                    CargarProductos()

                    Catch ex As Exception
                        MessageBox.Show("Este producto se encuentra en uso, no puede ser eliminado", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                    End Try

                End If
            Else
                MessageBox.Show("DEBE SELECCIONAR UN PRODUCTO A ELIMINAR DE LA LISTA!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End If
        End If
    End Sub

    Private Sub CmbCategoria_SelectedIndexChanged(sender As Object, e As EventArgs) Handles CmbCategoria.SelectedIndexChanged
        categoria = CmbCategoria.SelectedItem(0).ToString
    End Sub

    Private Sub CmbMarca_SelectedIndexChanged(sender As Object, e As EventArgs) Handles CmbMarca.SelectedIndexChanged
        marca = CmbMarca.SelectedItem(0).ToString

```

End Sub

```
Private Sub BtnIngresar_Click(sender As Object, e As EventArgs) Handles BtnIngresar.Click
    sw = 1
    Visibilidad()
End Sub
```

```
Private Sub BtnModificar_Click(sender As Object, e As EventArgs) Handles BtnModificar.Click
    sw = 2
    Visibilidad()
End Sub
```

```
Private Sub BtnEliminar_Click(sender As Object, e As EventArgs) Handles BtnEliminar.Click
    sw = 3
    Visibilidad()
End Sub
```

```
Private Sub BtnBuscar_Click(sender As Object, e As EventArgs) Handles BtnBuscar.Click
    sw = 4
    Visibilidad()
End Sub
```

```
Private Sub BtnCancelar_Click(sender As Object, e As EventArgs) Handles BtnCancelar.Click
    sw = 5
    Visibilidad()
    Limpiar()
End Sub
```

```
Private Sub BtnSalir_Click(sender As Object, e As EventArgs) Handles BtnSalir.Click
    FrmPrincipal.Show()
    FrmPrincipal.TxtCaja.Text = BsnNegocios.getMontoDinero
    Me.Dispose()
End Sub
```

```
Private Sub DgvProductos_RowHeaderMouseClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DgvProductos.RowHeaderMouseClick
    Dim row2 As DataGridViewRow = DgvProductos.CurrentRow
    If sw = 2 Or sw = 3 Or sw = 4 Then
        If row2.Cells(0).Value.Equals(DBNull.Value) Then
            TxtCodProducto.Text = ""
            TxtNombre.Text = ""
            TxtPrecio.Text = ""
            TxtPrecioCompra.Text = ""
            TxtStockMinimo.Text = ""
            TxtStockActual.Text = ""
            TxtDescripcionProducto.Text = ""
        Else
            TxtCodProducto.Text = row2.Cells(0).Value
            CmbMarca.SelectedValue = row2.Cells(1).Value
            CmbCategoria.SelectedValue = row2.Cells(2).Value
            CmbProveedor.SelectedItem = row2.Cells(3).Value
            TxtNombre.Text = row2.Cells(3).Value
            TxtPrecio.Text = row2.Cells(5).Value
            TxtPrecioCompra.Text = row2.Cells(4).Value
            TxtStockMinimo.Text = row2.Cells(6).Value
            TxtStockActual.Text = row2.Cells(7).Value
            TxtDescripcionProducto.Text = row2.Cells(8).Value
        End If
    End If
End Sub
```

```
Private Sub BtnBusqueda_Click(sender As Object, e As EventArgs)
```

```

If TxtBuscarCodigoProducto.Text = "" Then
    CargarProductos()
Else
    Dim producto As New ClsProducto
    producto.Codigo = TxtBuscarCodigoProducto.Text
    DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=123456; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where codigo_Producto = " & producto.Codigo & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProducto)
    conn.Close()
End If
End Sub

Private Sub TxtBuscar_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtBuscarCodigoProducto.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Function filtro(sw2 As Integer)
    DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
    If (TxtBuscarCodigoProducto.Text = "") And (TxtBuscarCodigoMarca.Text = "") And (txtBuscarNombre.Text = "") And
(TxtBuscarCodigoCategoria.Text = "") Then
        CargarProductos()
    Else
        DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
        Dim conn As MySqlConnection
        conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
        Dim username As Boolean = True
        conn.Open()
        Dim adapter As New MySqlDataAdapter
        Dim command As New MySqlCommand
        If (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text IsNot "")
And (TxtBuscarCodigoCategoria.Text IsNot "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &
" and Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & "and Nombre like '%" & txtBuscarNombre.Text & "%'"
& "and Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text & "%'"
            command.CommandText = sqlquery
        Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text IsNot
"") And (TxtBuscarCodigoCategoria.Text Is "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &
" and Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & "and Nombre like '%" & txtBuscarNombre.Text &
"%"
            command.CommandText = sqlquery
        Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &

```

```

" and Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & "and Codigo_categoria like '%" &
TxtBuscarCodigoCategoria.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &
" and Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text IsNot "")
And (TxtBuscarCodigoCategoria.Text IsNot "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &
"and Nombre like '%" & txtBuscarNombre.Text & "%'" & "and Codigo_categoria like '%" &
TxtBuscarCodigoCategoria.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'" &
"and Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where Codigo_Producto like '%" & TxtBuscarCodigoProducto.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text IsNot "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & "and Nombre like '%" & txtBuscarNombre.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text Is "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'"
command.CommandText = sqlquery
Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text Is "")
And (TxtBuscarCodigoCategoria.Text IsNot "") Then
Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
"
```

```

        Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & "and Codigo_categoria like '%" &
TxtBuscarCodigoCategoria.Text & "%'"
        command.CommandText = sqlquery
        Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text IsNot "") And
(TxtBuscarCodigoCategoria.Text Is "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Nombre like '%" & txtBuscarNombre.Text & "%'"
            command.CommandText = sqlquery
        Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text IsNot "") And
(TxtBuscarCodigoCategoria.Text IsNot "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Nombre like '%" & txtBuscarNombre.Text & "%'" & "and Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text &
"%"
            command.CommandText = sqlquery
        Elseif (TxtBuscarCodigoProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text Is "") And
(TxtBuscarCodigoCategoria.Text IsNot "") Then
            Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_actual,Descripcion
from Productos Where
Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text & "%'"
            command.CommandText = sqlquery
        End If
        command.Connection = conn
        adapter.SelectCommand = command
        adapter.Fill(DataSet1.DataTableProducto)
        conn.Close()

    End If
End Function

Private Sub TxtCodProducto_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtCodProducto.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub TxtPrecio_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtPrecio.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub TxtStockMinimo_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtStockMinimo.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub TxtStockActual_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtStockActual.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub TxtCantDefectuoso_KeyPress(sender As Object, e As KeyPressEventArgs)
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

```

```

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    lblFechaYHora.Text = "Hora actual: " & Date.Now.ToLongTimeString
End Sub

Private Sub TxtBuscarCodigoProducto_TextChanged(sender As Object, e As EventArgs) Handles
TxtBuscarCodigoProducto.TextChanged
    Dim sw2 As New Integer
    sw2 = 1
    filtro(sw2)
End Sub

Private Sub TxtBuscarCodigoMarca_TextChanged(sender As Object, e As EventArgs) Handles TxtBuscarCodigoMarca.TextChanged
    Dim sw3 As New Integer
    sw3 = 2
    filtro(sw3)
End Sub

Private Sub TxtBuscarCodigoCategoria_TextChanged(sender As Object, e As EventArgs) Handles
TxtBuscarCodigoCategoria.TextChanged
    Dim sw4 As New Integer
    sw4 = 3
    filtro(sw4)
End Sub

Private Sub TxtBuscarCodigoMarca_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtBuscarCodigoMarca.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub TxtBuscarCodigoCategoria_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TxtBuscarCodigoCategoria.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub CmbMarca_MouseClick(sender As Object, e As MouseEventArgs) Handles CmbMarca.MouseClick
    CmbMarca.DataSource = BsnNegocios.GetMarca
    CmbMarca.DisplayMember = "Nombre"
End Sub

Private Sub CmbProveedor_MouseClick(sender As Object, e As MouseEventArgs) Handles CmbProveedor.MouseClick
    CmbProveedor.DataSource = BsnNegocios.GetProveedor()
    CmbProveedor.DisplayMember = "Nombre"
End Sub
Private Sub CmbCategoria_MouseClick(sender As Object, e As MouseEventArgs) Handles CmbCategoria.MouseClick
    CmbCategoria.DataSource = BsnNegocios.GetCategoria()
    CmbCategoria.DisplayMember = "Nombre"
End Sub

Private Sub CmbProveedor_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
CmbProveedor.SelectedIndexChanged
    proveedor = CmbProveedor.SelectedItem(0).ToString
End Sub

Private Sub txtBuscarNombre_TextChanged(sender As Object, e As EventArgs) Handles txtBuscarNombre.TextChanged
    Dim sw2 As Integer = 3
    filtro(sw2)
End Sub

```

```

Function GetNombreMarca(codi As Integer)
    Dim s As String
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Nombre from marca Where Codigo_Marca = " & codi & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    Dim registro As MySqlDataReader = command.ExecuteReader
    If registro.Read Then
        s = registro("Nombre")
        Return s
    End If
    conn.Close()
End Function

Private Sub BtnImprimir_Click(sender As Object, e As EventArgs) Handles BtnImprimir.Click
    pagenumber = 1
    mRow = 0
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim newpage As Boolean = True
    Dim fmt As StringFormat = New StringFormat(StringFormatFlags.LineLimit)
    fmt.LineAlignment = StringAlignment.Center
    fmt.Trimming = StringTrimming.EllipsisCharacter
    Dim y As Int32 = e.MarginBounds.Top + 60
    Dim rc As Rectangle
    Dim x As Int32
    Dim h As Int32 = 0
    Dim row As DataGridViewRow
    If pagenumber = 1 Then
        Dim font As Font = New Font("Arial", 9)
        Dim font2 As Font = New Font("Arial", 16)
        Dim rc2 As RectangleF = New RectangleF(200, 30, 500, 30)
        e.Graphics.DrawRectangle(Pens.White, rc2.Left, rc2.Top, rc2.Width, rc2.Height)
        e.Graphics.DrawString("INFORME DE PRODUCTOS", font2, Brushes.Black, rc2, fmt)
        Dim rc3 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 100, 600, 30)
        e.Graphics.DrawRectangle(Pens.White, rc3.Left, rc3.Top, rc3.Width, rc3.Height)
        e.Graphics.DrawString("FECHA DE IMPRESIÓN: " & DateTime.Now.ToString("dd-MM-yyyy"), font, Brushes.Black, rc3, fmt)
        Dim rc4 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 125, 400, 30)
        e.Graphics.DrawRectangle(Pens.White, rc4.Left, rc4.Top, rc4.Width, rc4.Height)
        e.Graphics.DrawString("HORA DE IMPRESIÓN: " & DateTime.Now.ToString("HH:mm:ss"), font, Brushes.Black, rc4, fmt)
    End If
    row = DgvProductos.Rows(mRow)
    x = e.MarginBounds.Left - 50
    Dim i As Integer = 0
    Dim k As Integer = 0
    For Each cell As DataGridViewCell In row.Cells
        If cell.Visible Then
            If i = 0 Then k = -20
            If i = 1 Then k = -10
            If i = 2 Then k = -10
            If i = 3 Then k = 0
            If i = 4 Then k = -35
            If i = 5 Then k = -35
            If i = 6 Then k = -40
            If i = 7 Then k = -40
        End If
    Next
End Sub

```

```

If i = 8 Then k = +70
If i = 1 Then
    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
Elseif i = 2 Then
    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
Elseif i = 3 Then
    rc = New Rectangle(x - 180, y, 100 + k, cell.Size.Height)
Else
    rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
End If
e.Graphics.FillRectangle(Brushes.LightGray, rc)
e.Graphics.DrawRectangle(Pens.Black, rc)
fmt.Alignment = StringAlignment.Near
rc.Offset(5, 0)
If i = 0 Then
    e.Graphics.DrawString("Cod Producto", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 1 Then
    e.Graphics.DrawString("Marca", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 2 Then
    e.Graphics.DrawString("Categoria", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 4 Then
    e.Graphics.DrawString("P.Compra", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 5 Then
    e.Graphics.DrawString("P.Venta", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 6 Then
    e.Graphics.DrawString("S.Minimo", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 7 Then
    e.Graphics.DrawString("S.Actual", DgvProductos.Font, Brushes.Black, rc, fmt)
Else
    e.Graphics.DrawString(DgvProductos.Columns(cell.ColumnIndex).HeaderText,
        DgvProductos.Font, Brushes.Black, rc, fmt)
End If
x += rc.Width
h = Math.Max(h, rc.Height)
End If
i = i + 1
k = 0
Next
y += h
If pagenumber <= DgvProductos.RowCount Then
    Dim rc5 As RectangleF = New RectangleF(e.MarginBounds.Right - 40, e.MarginBounds.Bottom, 40, 30)
    e.Graphics.DrawRectangle(Pens.White, rc5.Left, rc5.Top, rc5.Width, rc5.Height)
    e.Graphics.DrawString("PAG " & pagenumber, Font, Brushes.Black, rc5, fmt)
    Dim thisNDX As Int32
    For thisNDX = mRow To DgvProductos.RowCount - 1
        If DgvProductos.Rows(thisNDX).IsNewRow Then Exit For
        row = DgvProductos.Rows(thisNDX)
        h = 0
        x = e.MarginBounds.Left - 50
        i = 0
        For Each cell As DataGridViewCell In row.Cells
            If cell.Visible Then
                If i = 0 Then k = -20
                If i = 1 Then k = -10
                If i = 2 Then k = -10
                If i = 3 Then k = 0
                If i = 4 Then k = -35
                If i = 5 Then k = -35
                If i = 6 Then k = -40
                If i = 7 Then k = -40
                If i = 8 Then k = +70
                If i = 1 Then
                    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
                Elseif i = 2 Then
                    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)

```

```

Elseif i = 3 Then
    rc = New Rectangle(x + -180, y, 100 + k, cell.Size.Height)
Else
    rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
End If
e.Graphics.DrawRectangle(Pens.Black, rc)
fmt.Alignment = StringAlignment.Near
rc.Offset(5, 0)
If i = 1 Then
    Dim nombre As String
    nombre = BsnNegocios.GetMarca2(cell.FormattedValue.ToString())
    e.Graphics.DrawString(nombre, DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 2 Then
    Dim nombre As String
    nombre = BsnNegocios.GetCategoria(cell.FormattedValue.ToString())
    e.Graphics.DrawString(nombre, DgvProductos.Font, Brushes.Black, rc, fmt)
Else
    e.Graphics.DrawString(cell.FormattedValue.ToString(),
        DgvProductos.Font, Brushes.Black, rc, fmt)
End If
x += rc.Width
h = Math.Max(h, rc.Height)
End If
i = i + 1
Next
y += h
mRow = thisNDX + 1
If (mRow = 36 * pagenumber) Then
    e.HasMorePages = True
    pagenumber += 1
Exit Sub
End If
Next
End If
e.HasMorePages = False
End Sub
End Class

```

3- VENTA

Imports MySql.Data.MySqlClient

Imports System.Globalization

Public Class FrmVenta

Dim BsnNegocios As New BsnNegocios

Dim marca As String

Dim categoria As String

Dim codigo As Integer

Dim valorNumericUpDown As Integer

Dim numero As String

Dim descuento As Integer

Dim dt As New DataSet1.DataTableCarritoDeVentasDataTable

Dim mediosDePago As Integer

Dim t As Integer

Dim pagenumber As Integer = 1

Dim mRow As Integer = 0

Dim carrito As Integer = 0

```

Dim seleccion As String = ""

Private Sub FrmVenta_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Cargar()
    CargarProductos()
    valorNumericUpDown = 1
    Me.BackColor = FrmPrincipal.BackColor
    ObtenerDescuento()
    CargarComboBox()
End Sub

Function CargarComboBox()
    CmbMediosDePago.DataSource = BsnNegocios.GetMediosDePago()
    CmbMediosDePago.DisplayMember = "Nombre"
End Function

Function Cargar()
    numero = Integer.Parse(BsnNegocios.GetNroVenta) + 1
    lblNroVenta.Text = lblNroVenta.Text + " " + numero
    txtRutVendedor.Text = FrmPrincipal.TextBox1.Text
    txtNombreVendedor.Text = FrmPrincipal.TextBox2.Text
    txtApellidoVendedor.Text = FrmPrincipal.TextBox3.Text
End Function

Private Sub BtnSalir_Click(sender As Object, e As EventArgs) Handles BtnSalir.Click
    FrmPrincipal.Show()
    FrmPrincipal.TxTcaja.Text = BsnNegocios.getMontoDinero
    Me.Dispose()
End Sub

Private Sub txtRutCliente_TextChanged(sender As Object, e As EventArgs) Handles txtRutCliente.TextChanged
    If txtRutCliente.Text IsNot "" Then
        FiltroCliente()
    End If
End Sub

Function FiltroCliente()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Rut_Cliente,Nombre,Apellido from clientes Where Rut_Cliente = " & txtRutCliente.Text & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    Dim registro As MySqlDataReader = command.ExecuteReader
    If registro.Read Then
        If registro("Rut_Cliente") = txtRutCliente.Text Then
            txtNombreCliente.Text = registro("Nombre")
            TxtApellidoCliente.Text = registro("Apellido")
            conn.Close()
        Else
            End If
    Else
        txtNombreCliente.Text = ""
        TxtApellidoCliente.Text = ""
    End If
    conn.Close()
End Function

Function CargarProductos()
    DataSet1.Clear()

```

```

Dim conn As MySqlConnection
conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
Dim username As Boolean = True
conn.Open()
Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio_Venta,Stock_Actual from Productos "
Dim adapter As New MySqlDataAdapter
Dim command As New MySqlCommand
command.CommandText = sqlquery
command.Connection = conn
adapter.SelectCommand = command
adapter.Fill(DataSet1.DataTableProductoVenta)
conn.Close()
End Function

Private Sub TxtCodProducto_TextChanged(sender As Object, e As EventArgs) Handles TxtCodProducto.TextChanged
    FiltroBuscar()
End Sub

Function FiltroProductoCodigoProducto()
    DataSet1.Clear()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio,Stock_Actual from Productos where Codigo_Producto =" &
    TxtCodProducto.Text & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProductoVenta)
    conn.Close()
End Function

Function FiltroProductoCodigoMarca()
    DataSet1.Clear()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio,Stock_Actual from Productos where Codigo_Marca =" & marca
    & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProductoVenta)
    conn.Close()
End Function

Function FiltroProductoCodigoCategoria()
    DataSet1.Clear()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio,Stock_Actual from Productos where Codigo_Categoria =" &
    categoria & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery

```

```

    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProductoVenta)
    conn.Close()
End Function

Private Sub TxtCodProducto_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtCodProducto.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub txtSubTotal_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtSubTotal.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Sub txtRutCliente_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtRutCliente.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub

Private Function limpiar()
    TxtNombreProducto.Text = ""
    TxtDescripcionProducto.Text = ""
    TxtPrecioProducto.Text = 0
    TxtStockProducto.Text = 0
    txtPrecio.Text = 0
    NumericUpDown1.Value = 1
    txtSubTotal.Text = 0
    txtTotal.Text = 0
    txtRutCliente.Text = ""
    txtNombreCliente.Text = ""
    TxtApellidoCliente.Text = ""
End Function

Private Sub DgvProductos_RowHeaderMouseClick(sender As Object, e As DataGridViewCellMouseEventArgs) Handles
DgvProductos.RowHeaderMouseClick
    Dim row As DataGridViewRow = DgvProductos.CurrentRow
    If row.Cells(0).Value.Equals(DBNull.Value) Then
        limpiar()
    Else
        codigo = row.Cells(0).Value
        BuscarDatosProducto()
        NumericUpDown1.Value = 0
        t = row.Cells(3).Value
    End If
End Sub

Function BuscarDatosProducto()
    Dim conn As MySqlConnection
    Dim marca2 As Integer
    Dim categoria2 As Integer
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select
Codigo_Producto,Codigo_Marca,Codigo_Categoria,Nombre,Precio_Compra,Precio_Venta,Stock_Minimo,Stock_Actual,Descripcion
from Productos where Codigo_Producto =" & codigo & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand

```

```

command.CommandText = sqlquery
command.Connection = conn
Dim registro As MySqlDataReader = command.ExecuteReader
If registro.Read Then
    TxtNombreProducto.Text = registro("Nombre")
    marca2 = registro("Codigo_Marca")
    categoria2 = registro("Codigo_Categoria")
    TxtDescripcionProducto.Text = registro("Descripcion")
    TxtPrecioProducto.Text = registro("Precio_Venta")
    txtPrecio.Text = TxtPrecioProducto.Text
    TxtStockProducto.Text = registro("Stock_Actual")
    conn.Close()
Else
    limpiar()
End If
conn.Close()
End Function

Private Sub NumericUpDown1_ValueChanged(sender As Object, e As EventArgs) Handles NumericUpDown1.ValueChanged
If TxtNombreProducto.Text IsNot "" Then
    If NumericUpDown1.Value <= Integer.Parse(TxtStockProducto.Text) Then
        If NumericUpDown1.Value > 0 Then
            txtPrecio.Text = Integer.Parse(TxtPrecioProducto.Text) * NumericUpDown1.Value
        End If
    Else
        NumericUpDown1.Value = Integer.Parse(TxtStockProducto.Text)
    End If
End If
End Sub

Private Sub BtnAgregar_Click(sender As Object, e As EventArgs) Handles BtnAgregar.Click
If carrito < 14 Then
    If txtPrecio.Text = 0 Then
        lblDescripcionProducto.Text = "Debe seleccionar un producto en la lista para" + vbCrLf + "agregar al carro"
    Else
        If t > 0 Then
            If NumericUpDown1.Value > 0 Then
                Dim i As Integer
                Dim sw As Integer = 0
                For i = 0 To DgvCarrito.RowCount - 1
                    Dim R As DataRow = dt.NewRow
                    If DgvCarrito.Rows(i).Cells(0).Value = codigo Then
                        If (DgvCarrito.Rows(i).Cells(3).Value + NumericUpDown1.Value) <= Integer.Parse(TxtStockProducto.Text) Then
                            DgvCarrito.Rows(i).Cells(3).Value = DgvCarrito.Rows(i).Cells(3).Value + NumericUpDown1.Value
                            DgvCarrito.Rows(i).Cells(4).Value = DgvCarrito.Rows(i).Cells(4).Value + Integer.Parse(txtPrecio.Text)
                            sw = 1
                        Else
                            MsgBox("LA CANTIDAD TOTAL DEL PRODUCTO EXCEDE EL STOCK ACTUAL EN EXISTENCIA")
                            sw = 1
                        End If
                    End If
                Next
            End If
            If sw = 0 Then
                If NumericUpDown1.Value <= Integer.Parse(TxtStockProducto.Text) Then
                    Dim R As DataRow = dt.NewRow
                    R("Cantidad") = NumericUpDown1.Value
                    R("Precio_Total") = txtPrecio.Text
                    Dim row As DataGridViewRow = DgvProductos.CurrentRow
                    R("Codigo_Producto") = codigo
                    R("Nombre") = TxtNombreProducto.Text
                    R("Precio_Unitario") = TxtPrecioProducto.Text
                    txtSubTotal.Text = Integer.Parse(txtSubTotal.Text) + Integer.Parse(txtPrecio.Text)
                    txtTotal.Text = Integer.Parse(txtSubTotal.Text) - Integer.Parse(txtSubTotal.Text) * (descuento / 100)
                    dt.Rows.Add(R)
                End If
            End If
        End If
    End If
End Sub

```

```

        DgvCarrito.DataSource = dt
        lblDescripcionProducto.Text = ""
        carrito = +1
    Else
        MsgBox("LA CANTIDAD INGRESADA EXCEDE EL STOCK ACTUAL DEL PRODUCTO")
    End If
End If
If txtNombreCliente.Text = "" Then
    lblAvisoRut.Text = "Ingrese un Rut" + vbCrLf + " de cliente"
End If
Else
    MsgBox("DEBE SELECCIONAR UNA CANTIDAD DEL PRODUCTO EN LA COMPRA")
End If
Else
    MsgBox("EL PRODUCTO SELECCIONADO NO TIENE STOCK")
End If
End If
Else
    MsgBox("EL CARRITO ESTA REPLETO")
End If

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim row As DataGridViewRow = DgvCarrito.CurrentRow
    Try
        If row.Cells(0).Value.Equals(DBNull.Value) Then
        Else
            txtSubTotal.Text = Integer.Parse(txtSubTotal.Text) - row.Cells(4).Value
            DgvCarrito.Rows.Remove(row)
            carrito = -1
        End If
    Catch ex As Exception
    End Try
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    lblFechaYHora.Text = "Fecha: " & Date.Now & " " & Date.Now.ToLongTimeString
End Sub
Function ObtenerDescuento()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Descuento from varios where Codigo =" & 1 & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    Dim registro As MySqlDataReader = command.ExecuteReader
    If registro.Read Then
        descuento = registro("Descuento")
        txtDescuento.Text = descuento.ToString & "%"
        conn.Close()
    End If
End Function

Private Sub BtnIngresar_Click(sender As Object, e As EventArgs) Handles BtnIngresar.Click
    If txtRutCliente.Text IsNot "" Then
        If seleccion IsNot "" Then
            If DgvCarrito.RowCount > 0 Then
                Dim venta As New ClsDetalleVenta
                Dim count As New Integer
            End If
        End If
    End If

```

```

venta.Nro_Venta = numero
venta.Rut_Usuario = txtRutVendedor.Text
venta.Rut_Cliente = txtRutCliente.Text
venta.Cod_Pago = mediosDePago
venta.Descuento = descuento
venta.Monto_Total = txtTotal.Text
venta.Devolucion = 0
If seleccion = "Factura" Then
    venta.Factura = (BsnNegocios.GetNroFactura()) + 1
    venta.Boleta = 0
Elseif seleccion = "Boleta" Then
    venta.Factura = 0
    venta.Boleta = (BsnNegocios.GetNroBoleta()) + 1
End If
BsnNegocios.insertarMontoPedido(txtRutVendedor.Text, System.DateTime.Now.ToString(("dd-MM-yyyy")), "Venta
Productos", txtTotal.Text)
BsnNegocios.InsertVenta(venta)
count = DgvCarrito.RowCount
Dim Comodin_Nro_Venta As Integer
Dim comodin_Codigo_Producto As Integer
Dim comodin_Cantidad As Integer
Dim comodin_sub_total As Integer
Dim i As Integer = 0
While i < count
    Comodin_Nro_Venta = numero
    comodin_Codigo_Producto = DgvCarrito.Rows(i).Cells(0).Value
    comodin_Cantidad = DgvCarrito.Rows(i).Cells(3).Value
    comodin_sub_total = DgvCarrito.Rows(i).Cells(4).Value
    Dim connectionString As String = "Server=localhost; User Id=root; Password=; Database=autosave"
    Dim SQLConnection As MySqlConnection = New MySqlConnection
    Dim oDt_sched As New DataTable()
    SQLConnection = New MySqlConnection()
    SQLConnection.ConnectionString = connectionString
    Dim str_carSql As String
    SQLConnection.Open()
    Dim sqlCommand As New MySqlCommand
    str_carSql = "Insert into detalle_venta (Nro_Venta,Codigo_Producto,Cantidad,Sub_Total) values (" &
Comodin_Nro_Venta & "," & comodin_Codigo_Producto & "," & comodin_Cantidad & "," & comodin_sub_total & ")"
    sqlCommand.Connection = SQLConnection
    sqlCommand.CommandText = str_carSql
    sqlCommand.ExecuteNonQuery()
    SQLConnection.Close()
    SQLConnection.Open()
    Dim sqlCommand1 As New MySqlCommand
    str_carSql = "Insert into ajuste_stock (Codigo_Producto,Cantidad,Fecha,Tipo,Descripcion) values (" &
comodin_Codigo_Producto & "," & comodin_Cantidad & "," & Date.Now.ToString & "','salida','venta Nro: " & Comodin_Nro_Venta
& "'"
    sqlCommand1.Connection = SQLConnection
    sqlCommand1.CommandText = str_carSql
    sqlCommand1.ExecuteNonQuery()
    SQLConnection.Close()
    SQLConnection.Open()
    Dim sqlCommand2 As New MySqlCommand
    str_carSql = "UPDATE productos SET Stock_Actual = (Stock_Actual - " & comodin_Cantidad & ")"
    str_carSql = str_carSql + " WHERE Codigo_Producto = " & comodin_Codigo_Producto
    sqlCommand2.Connection = SQLConnection
    sqlCommand2.CommandText = str_carSql
    sqlCommand2.ExecuteNonQuery()
    SQLConnection.Close()
    i = i + 1
End While
MessageBox.Show("Venta realizada con exito", "", MessageBoxButtons.OK, MessageBoxIcon.Information)
limpiar()
CargarProductos()
LimpiarCarrito()

```

```

        numero = numero + 1
        lblNroVenta.Text = lblNroVenta.Text + " " + numero
    Else
        MessageBox.Show("El carrito de compras se encuentra vacio", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    End If
Else
    MessageBox.Show("Debe seleccionar Factura o Boleta", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

Else
    MessageBox.Show("Debe seleccionar el rut del cliente para continuar", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error)
End If

End Sub

Private Sub CmbMediosDePago_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
CmbMediosDePago.SelectedIndexChanged
    mediosDePago = CmbMediosDePago.SelectedItem(0).ToString
End Sub

Private Sub txtNombreCliente_TextChanged(sender As Object, e As EventArgs) Handles txtNombreCliente.TextChanged
    lblAvisoRut.Text = ""
End Sub

Private Sub BtnRecargar_Click(sender As Object, e As EventArgs) Handles BtnRecargar.Click
    CargarProductos()
    TxtCodProducto.Text = ""
    TxtBuscarCodigoCategoria.Text = ""
    TxtBuscarCodigoMarca.Text = ""
    txtBuscarNombre.Text = ""
End Sub

Function LimpiarCarrito()
    DgvCarrito.DataSource = Nothing
End Function

Private Sub TxtBuscarCodigoMarca_TextChanged(sender As Object, e As EventArgs) Handles TxtBuscarCodigoMarca.TextChanged
    FiltroBuscar()
End Sub

Function FiltroBuscar()
    DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
    If (TxtCodProducto.Text = "") And (TxtBuscarCodigoMarca.Text = "") And (txtBuscarNombre.Text = "") And
(TxtBuscarCodigoCategoria.Text = "") Then
        CargarProductos()
    Else
        DataSet1.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
        Dim conn As MySqlConnection
        conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
        Dim username As Boolean = True
        conn.Open()
        Dim adapter As New MySqlDataAdapter
        Dim command As New MySqlCommand
        If (TxtCodProducto.Text IsNot "") And (TxtBuscarCodigoMarca.Text IsNot "") And (txtBuscarNombre.Text IsNot "") And
(TxtBuscarCodigoCategoria.Text IsNot "") Then
            Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio_Venta,Stock_Actual from Productos Where
Codigo_Producto like '%" & TxtCodProducto.Text & "%'" &
" and Codigo_Marca like '%" & TxtBuscarCodigoMarca.Text & "%'" & " and Nombre like '%" & txtBuscarNombre.Text & "%'"
& " and Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text & "%'"
            command.CommandText = sqlquery

```



```

        command.CommandText = sqlquery
    Elseif (TxtCodProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text IsNot "") And
(TxtBuscarCodigoCategoria.Text Is "") Then
        Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio_Venta,Stock_Actual from Productos Where
        Nombre like '%" & txtBuscarNombre.Text & "%"
        command.CommandText = sqlquery
    Elseif (TxtCodProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text IsNot "") And
(TxtBuscarCodigoCategoria.Text IsNot "") Then
        Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio_Venta,Stock_Actual from Productos Where
        Nombre like '%" & txtBuscarNombre.Text & "%'" & "and Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text &
"%'"
        command.CommandText = sqlquery
    Elseif (TxtCodProducto.Text Is "") And (TxtBuscarCodigoMarca.Text Is "") And (txtBuscarNombre.Text Is "") And
(TxtBuscarCodigoCategoria.Text IsNot "") Then
        Dim sqlquery As String = "select Codigo_Producto,Nombre,Precio_Venta,Stock_Actual from Productos Where
        Codigo_categoria like '%" & TxtBuscarCodigoCategoria.Text & "%"
        command.CommandText = sqlquery
    End If
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProductoVenta)
    conn.Close()

End If
End Function

Private Sub TxtBuscarCodigoCategoria_TextChanged(sender As Object, e As EventArgs) Handles
TxtBuscarCodigoCategoria.TextChanged
    FiltroBuscar()
End Sub

Private Sub txtBuscarNombre_TextChanged(sender As Object, e As EventArgs) Handles txtBuscarNombre.TextChanged
    FiltroBuscar()
End Sub

Private Sub BtnModificar_Click(sender As Object, e As EventArgs) Handles BtnModificar.Click
    FrmVentaDevolucion.Show()
    Me.Dispose()
End Sub

Private Sub BtnFactura_Click(sender As Object, e As EventArgs) Handles BtnFactura.Click
    pagenumber = 1
    mRow = 0
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim newpage As Boolean = True
    Dim fmt As StringFormat = New StringFormat(StringFormatFlags.LineLimit)
    fmt.LineAlignment = StringAlignment.Center
    fmt.Trimming = StringTrimming.EllipsisCharacter

    Dim y As Int32 = e.MarginBounds.Top + 60
    Dim rc As Rectangle
    Dim x As Int32
    Dim h As Int32 = 0
    Dim row As DataGridViewRow
    'If pagenumber = 1 Then
    Dim font As Font = New Font("Univers Medium", 9)
    Dim font2 As Font = New Font("Univers Medium", 16)
    Dim font3 As Font = New Font("Univers Medium", 12)
    Dim font4 As Font = New Font("Univers Medium", 14)

```

Dim font5 As Font = New Font("Univers Medium", 9, FontStyle.Bold)
Dim font6 As Font = New Font("Univers Medium", 7)
Dim font7 As Font = New Font("Univers Medium", 10, FontStyle.Bold)
Dim font8 As Font = New Font("Univers Medium", 12, FontStyle.Bold)

Dim rc1 As RectangleF = New RectangleF(e.MarginBounds.Left - 30, e.MarginBounds.Top - 50, 200, 50)
e.Graphics.DrawRectangle(Pens.White, rc1.Left, rc1.Top, rc1.Width, rc1.Height)
e.Graphics.DrawString("Autosave", font8, Brushes.Black, rc1, fmt)

Dim rc2 As RectangleF = New RectangleF(e.MarginBounds.Left - 60, e.MarginBounds.Top, 500, 30)
e.Graphics.DrawRectangle(Pens.Transparent, rc2.Left, rc2.Top, rc2.Width, rc2.Height)
e.Graphics.DrawString("VENTA AL POR MENOR DE ARTICULOS DE REPUESTO AUTOMOVILISTICOS", font, Brushes.Black, rc2, fmt)

Dim rc9 As RectangleF = New RectangleF(e.MarginBounds.Left + 50, e.MarginBounds.Top + 15, 280, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc9.Left, rc9.Top, rc9.Width, rc9.Height)
e.Graphics.DrawString("_____", font, Brushes.Black, rc9, fmt)

Dim rc69 As RectangleF = New RectangleF(e.MarginBounds.Left + 470, e.MarginBounds.Top - 80, 260, 110)
e.Graphics.DrawRectangle(Pens.Black, rc69.Left, rc69.Top, rc69.Width, rc69.Height)

Dim rc3 As RectangleF = New RectangleF(e.MarginBounds.Left + 480, e.MarginBounds.Top - 70, 260, 30)
e.Graphics.DrawRectangle(Pens.Transparent, rc3.Left, rc3.Top, rc3.Width, rc3.Height)
e.Graphics.DrawString(" RUT NUMERO.: 77.599.410-k", font5, Brushes.Black, rc3, fmt)
'vbCrLf & " FACTURA ELECTRONICA" & vbCrLf & " N°"

Dim rc67 As RectangleF = New RectangleF(e.MarginBounds.Left + 510, e.MarginBounds.Top - 40, 200, 30)
e.Graphics.DrawRectangle(Pens.Transparent, rc67.Left, rc67.Top, rc67.Width, rc67.Height)
e.Graphics.DrawString("FACTURA ELECTRONICA", font5, Brushes.Black, rc67, fmt)

Dim rc68 As RectangleF = New RectangleF(e.MarginBounds.Left + 520, e.MarginBounds.Top, 260, 30)
e.Graphics.DrawRectangle(Pens.Transparent, rc68.Left, rc68.Top, rc68.Width, rc68.Height)
e.Graphics.DrawString(" N° " & (BsnNegocios.GetNroFactura()) + 1, font5, Brushes.Black, rc68, fmt)

Dim rc24 As RectangleF = New RectangleF(e.MarginBounds.Left + 540, e.MarginBounds.Top + 40, 120, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc24.Left, rc24.Top, rc24.Width, rc24.Height)
e.Graphics.DrawString("S.I.I. - OLMUÉ", font, Brushes.Black, rc24, fmt)

Dim rc74 As RectangleF = New RectangleF(e.MarginBounds.Left - 60, e.MarginBounds.Top + 40, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc74.Left, rc74.Top, rc74.Width, rc74.Height)
e.Graphics.DrawString("RIQUELME 5609", font, Brushes.Black, rc74, fmt)

Dim rc75 As RectangleF = New RectangleF(e.MarginBounds.Left + 200, e.MarginBounds.Top + 40, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc75.Left, rc75.Top, rc75.Width, rc75.Height)
e.Graphics.DrawString("OLMUÉ - OLMUÉ", font, Brushes.Black, rc75, fmt)

Dim rc76 As RectangleF = New RectangleF(e.MarginBounds.Left - 60, e.MarginBounds.Top + 70, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc76.Left, rc76.Top, rc76.Width, rc76.Height)
e.Graphics.DrawString("FONO: 977110966", font, Brushes.Black, rc76, fmt)

Dim rc7 As RectangleF = New RectangleF(e.MarginBounds.Left + 200, e.MarginBounds.Top + 70, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc7.Left, rc7.Top, rc7.Width, rc7.Height)
e.Graphics.DrawString("EMAIL: RepuestosAutosave@gmail.com", font, Brushes.Black, rc7, fmt)

Dim rc8 As RectangleF = New RectangleF(e.MarginBounds.Left + 490, e.MarginBounds.Top + 70, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc8.Left, rc8.Top, rc8.Width, rc8.Height)
e.Graphics.DrawString("OLMUÉ, " & DateTime.Today.ToString("dd") & " DE " & (DateTime.Today.ToString("MMMM")).ToUpper
&
" DEL " & DateTime.Today.Date.ToString("yyyy"), font5, Brushes.Black, rc8, fmt)

Dim rc4 As RectangleF = New RectangleF(e.MarginBounds.Left - 70, e.MarginBounds.Top + 100, 800, 145)

```

e.Graphics.DrawRectangle(Pens.Black, rc4.Left, rc4.Top, rc4.Width, rc4.Height)
'*****
***** DENTRRO DEL 4° CUADRO
*****
Dim texto1 As String
Dim texto2 As String
Dim u As Integer = 0
Dim t, m As Integer
t = 1
While t < 6
    If t = 1 Then
        texto1 = "Señor(es)"
        texto2 = txtNombreCliente.Text & " " & TxtApellidoCliente.Text

    ElseIf t = 2 Then
        texto1 = "RUT"
        texto2 = txtRutCliente.Text
    ElseIf t = 3 Then
        texto1 = "Giro"
        texto2 = "*****"
    ElseIf t = 4 Then
        texto1 = "Dirección"
        texto2 = BsnNegocios.GetClienteDireccion(Integer.Parse(txtRutCliente.Text))
    ElseIf t = 5 Then
        texto1 = "Comuna"
        texto2 = BsnNegocios.GetClienteComuna(Integer.Parse(txtRutCliente.Text))
    End If
    Dim rc13 As RectangleF = New RectangleF(e.MarginBounds.Left - 60, e.MarginBounds.Top + 110 + u, 120, 20)
    e.Graphics.DrawRectangle(Pens.Transparent, rc13.Left, rc13.Top, rc13.Width, rc13.Height)
    e.Graphics.DrawString(texto1, font5, Brushes.Black, rc13, fmt)

    rc = New Rectangle(e.MarginBounds.Left + 35, e.MarginBounds.Top + 110 + u, 20, 20)
    e.Graphics.DrawRectangle(Pens.Transparent, rc)
    fmt.Alignment = StringAlignment.Near
    rc.Offset(5, 0)
    e.Graphics.DrawString(":", DgvCarrito.Font, Brushes.Black, rc, fmt)

    Dim rc27 As RectangleF = New RectangleF(e.MarginBounds.Left + 50, e.MarginBounds.Top + 110 + u, 280, 20)
    e.Graphics.DrawRectangle(Pens.Transparent, rc27.Left, rc27.Top, rc27.Width, rc27.Height)
    e.Graphics.DrawString(texto2, font, Brushes.Black, rc27, fmt)
    t = t + 1
    u = u + 25
End While
t = 1
u = 0

While t < 6
    If t = 1 Then
        texto1 = "Telefono"
        texto2 = "*****"
        texto2 = BsnNegocios.GetClienteTelefono(Integer.Parse(txtRutCliente.Text))
    ElseIf t = 2 Then
        texto1 = "Vencimiento"
        texto2 = DateTime.Now
    ElseIf t = 3 Then
        texto1 = "Medios de pago"
        texto2 = CmbMediosDePago.Text
    ElseIf t = 4 Then
        texto1 = "Vendedor"
        texto2 = txtNombreVendedor.Text & " " & txtApellidoVendedor.Text
    ElseIf t = 5 Then
        texto1 = "Ciudad"
        texto2 = BsnNegocios.GetClienteCiudad(Integer.Parse(txtRutCliente.Text))
    End If

```

```

Dim rc42 As RectangleF = New RectangleF(e.MarginBounds.Left + 370, e.MarginBounds.Top + 110 + u, 140, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc42.Left, rc42.Top, rc42.Width, rc42.Height)
e.Graphics.DrawString(texto1, font5, Brushes.Black, rc42, fmt)
rc = New Rectangle(e.MarginBounds.Left + 480, e.MarginBounds.Top + 110 + u, 20, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc)
fmt.Alignment = StringAlignment.Near
rc.Offset(5, 0)
e.Graphics.DrawString(":", DgvCarrito.Font, Brushes.Black, rc, fmt)
Dim rC43 As RectangleF = New RectangleF(e.MarginBounds.Left + 500, e.MarginBounds.Top + 110 + u, 300, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rC43.Left, rC43.Top, rC43.Width, rC43.Height)
e.Graphics.DrawString(texto2, font, Brushes.Black, rC43, fmt)
t = t + 1
u = u + 25
End While

*****
****
Dim rc23 As RectangleF = New RectangleF(e.MarginBounds.Left - 60, e.MarginBounds.Top + 250, 90, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc23.Left, rc23.Top, rc23.Width, rc23.Height)
e.Graphics.DrawString("REFERENCIA", font, Brushes.Black, rc23, fmt)

Dim rc56 As RectangleF = New RectangleF(e.MarginBounds.Left + 40, e.MarginBounds.Top + 250, 20, 20)
e.Graphics.DrawRectangle(Pens.Transparent, rc56.Left, rc56.Top, rc56.Width, rc56.Height)
e.Graphics.DrawString(":", font, Brushes.Black, rc56, fmt)

Dim rc5 As RectangleF = New RectangleF(e.MarginBounds.Left - 70, e.MarginBounds.Top + 290, 800, 450)
e.Graphics.DrawRectangle(Pens.Black, rc5.Left, rc5.Top, rc5.Width, rc5.Height)

Dim rc6 As RectangleF = New RectangleF(e.MarginBounds.Left + 490, e.MarginBounds.Top + 745, 240, 70)
e.Graphics.DrawRectangle(Pens.Transparent, rc6.Left, rc6.Top, rc6.Width, rc6.Height)

Dim rc38 As RectangleF = New RectangleF(e.MarginBounds.Left + 350, e.MarginBounds.Top + 820, 375, 60)
e.Graphics.DrawRectangle(Pens.Black, rc38.Left, rc38.Top, rc38.Width, rc38.Height)

Dim rc39 As RectangleF = New RectangleF(e.MarginBounds.Left + 352, e.MarginBounds.Top + 820, 130, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc39.Left, rc39.Top, rc39.Width, rc39.Height)
e.Graphics.DrawString("Observaciones:", font, Brushes.Black, rc39, fmt)

Dim rc50 As RectangleF = New RectangleF(e.MarginBounds.Left + 352, e.MarginBounds.Top + 880, 380, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc50.Left, rc50.Top, rc50.Width, rc50.Height)
e.Graphics.DrawString("Cancelado por : _____", font, Brushes.Black, rc50, fmt)

Dim rc51 As RectangleF = New RectangleF(e.MarginBounds.Left - 70, e.MarginBounds.Top + 910, 800, 60)
e.Graphics.DrawRectangle(Pens.Black, rc51.Left, rc51.Top, rc51.Width, rc51.Height)

Dim rc52 As RectangleF = New RectangleF(e.MarginBounds.Left - 65, e.MarginBounds.Top + 910, 290, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc52.Left, rc52.Top, rc52.Width, rc52.Height)
e.Graphics.DrawString("Nombre : _____", font, Brushes.Black, rc52, fmt)

Dim rc53 As RectangleF = New RectangleF(e.MarginBounds.Left + 220, e.MarginBounds.Top + 910, 380, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc53.Left, rc53.Top, rc53.Width, rc53.Height)
e.Graphics.DrawString("R.U.T : _____", font, Brushes.Black, rc53, fmt)

Dim rc54 As RectangleF = New RectangleF(e.MarginBounds.Left - 65, e.MarginBounds.Top + 925, 290, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc54.Left, rc54.Top, rc54.Width, rc54.Height)
e.Graphics.DrawString("Recinto : _____", font, Brushes.Black, rc54, fmt)

Dim rc55 As RectangleF = New RectangleF(e.MarginBounds.Left + 220, e.MarginBounds.Top + 925, 380, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc55.Left, rc55.Top, rc55.Width, rc55.Height)
e.Graphics.DrawString("Fecha : _____", font, Brushes.Black, rc55, fmt)

```

```

Dim rc57 As RectangleF = New RectangleF(e.MarginBounds.Left + 490, e.MarginBounds.Top + 925, 300, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc57.Left, rc57.Top, rc57.Width, rc57.Height)
e.Graphics.DrawString("Firma : _____", font, Brushes.Black, rc57, fmt)

```

```

Dim rc58 As RectangleF = New RectangleF(e.MarginBounds.Left - 65, e.MarginBounds.Top + 930, 900, 60)
e.Graphics.DrawRectangle(Pens.Transparent, rc58.Left, rc58.Top, rc58.Width, rc58.Height)
e.Graphics.DrawString("El acuso de recibo que se declara en este acto de acuerdo a lo dispuesto en la letra b) del Art 4°, y la letra
c) del Art 5° de la Ley 19.953, acredita que la entrega " & vbCrLf &
"de mercaderias o servicio(s) prestado(s) ha(n) sido recibido(s)", font6, Brushes.Black, rc58, fmt)

```

```

y = e.MarginBounds.Top + 290
'End If
x = e.MarginBounds.Left - 70
Dim i As Integer = 0
Dim k As Integer = 0
Dim encabezado As String
For j = 0 To 4
    If i = 0 Then
        k = -65
        encabezado = "Codigo"
        rc = New Rectangle(x, y, 100 + k, 30)
        e.Graphics.FillRectangle(Brushes.DarkGray, rc)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        e.Graphics.DrawString("Item", Font, Brushes.White, rc, fmt)
        x += rc.Width
        k = -20
    ElseIf i = 1 Then
        k = +200
        encabezado = "Nombre Producto"
    ElseIf i = 2 Then
        k = +20
        encabezado = "Precio Unitario"
    ElseIf i = 3 Then
        k = -10
        encabezado = "Cantidad"
    ElseIf i = 4 Then
        k = -20
        rc = New Rectangle(x, y, 100 + k, 30)
        e.Graphics.FillRectangle(Brushes.DarkGray, rc)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        e.Graphics.DrawString("Descuento", Font, Brushes.White, rc, fmt)
        x += rc.Width
        k = -5
        encabezado = "Precio Total"
    End If
    rc = New Rectangle(x, y, 100 + k, 30)
    e.Graphics.FillRectangle(Brushes.DarkGray, rc)
    e.Graphics.DrawRectangle(Pens.Black, rc)
    fmt.Alignment = StringAlignment.Near
    rc.Offset(5, 0)
    e.Graphics.DrawString(encabezado, Font, Brushes.White, rc, fmt)
    x += rc.Width
    i = i + 1
    k = 0
Next

```

```

y += h
Dim rc16 As RectangleF = New RectangleF(e.MarginBounds.Left - 35, e.MarginBounds.Top + 290, 80, 450)
e.Graphics.DrawRectangle(Pens.Black, rc16.Left, rc16.Top, rc16.Width, rc16.Height)
Dim rc17 As RectangleF = New RectangleF(e.MarginBounds.Left + 345, e.MarginBounds.Top + 290, 120, 450)
e.Graphics.DrawRectangle(Pens.Black, rc17.Left, rc17.Top, rc17.Width, rc17.Height)
Dim rc18 As RectangleF = New RectangleF(e.MarginBounds.Left + 465, e.MarginBounds.Top + 290, 90, 450)

```

```

e.Graphics.DrawRectangle(Pens.Black, rc18.Left, rc18.Top, rc18.Width, rc18.Height)
Dim rc19 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 290, 80, 450)
e.Graphics.DrawRectangle(Pens.Black, rc19.Left, rc19.Top, rc19.Width, rc19.Height)
Dim total As Integer = 0
Dim rc32 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 745, 110, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc32.Left, rc32.Top, rc32.Width, rc32.Height)
e.Graphics.DrawString("Neto", font, Brushes.Black, rc32, fmt)
Dim rc82 As RectangleF = New RectangleF(e.MarginBounds.Left + 630, e.MarginBounds.Top + 745, 110, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc82.Left, rc82.Top, rc82.Width, rc82.Height)
e.Graphics.DrawString("¢", font, Brushes.Black, rc82, fmt)
Dim rc40 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 750, 310, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc40.Left, rc40.Top, rc40.Width, rc40.Height)
e.Graphics.DrawString("_____ ", font, Brushes.Black, rc40, fmt)
Dim rc33 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 768, 110, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc33.Left, rc33.Top, rc33.Width, rc33.Height)
e.Graphics.DrawString("IVA 19%", font, Brushes.Black, rc33, fmt)
Dim rc83 As RectangleF = New RectangleF(e.MarginBounds.Left + 630, e.MarginBounds.Top + 768, 110, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc83.Left, rc83.Top, rc83.Width, rc83.Height)
e.Graphics.DrawString("$", font, Brushes.Black, rc83, fmt)
Dim rc41 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 774, 310, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc41.Left, rc41.Top, rc41.Width, rc41.Height)
e.Graphics.DrawString("_____ ", font, Brushes.Black, rc41, fmt)
Dim rc34 As RectangleF = New RectangleF(e.MarginBounds.Left + 555, e.MarginBounds.Top + 791, 110, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc34.Left, rc34.Top, rc34.Width, rc34.Height)
e.Graphics.DrawString("Total", font, Brushes.Black, rc34, fmt)
Dim r84 As RectangleF = New RectangleF(e.MarginBounds.Left + 630, e.MarginBounds.Top + 791, 110, 24)
e.Graphics.DrawRectangle(Pens.Transparent, r84.Left, r84.Top, r84.Width, r84.Height)
e.Graphics.DrawString("¢", font, Brushes.Black, r84, fmt)

```

```

Dim rc35 As RectangleF = New RectangleF(e.MarginBounds.Left + 650, e.MarginBounds.Top + 745, 130, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc35.Left, rc35.Top, rc35.Width, rc35.Height)
Dim rc36 As RectangleF = New RectangleF(e.MarginBounds.Left + 650, e.MarginBounds.Top + 768, 130, 23)
e.Graphics.DrawRectangle(Pens.Transparent, rc36.Left, rc36.Top, rc36.Width, rc36.Height)
Dim rc37 As RectangleF = New RectangleF(e.MarginBounds.Left + 650, e.MarginBounds.Top + 791, 130, 24)
e.Graphics.DrawRectangle(Pens.Transparent, rc37.Left, rc37.Top, rc37.Width, rc37.Height)

```

y = e.MarginBounds.Top + 320

Try

```

Dim thisNDX As Int32
Dim j As Integer = 0
For thisNDX = mRow To DgvCarrito.RowCount - 1
    If DgvCarrito.Rows(thisNDX).IsNewRow Then Exit For
    If (mRow = (15 * pagenummer) - 1) Then
        e.Graphics.DrawString(total, font, Brushes.Black, rc37, fmt)
        e.Graphics.DrawString((total * 81) \ 100, font, Brushes.Black, rc35, fmt)
        e.Graphics.DrawString(total * 19 \ 100, font, Brushes.Black, rc36, fmt)
        e.HasMorePages = True
        pagenummer += 1
    Exit Sub
End If
row = DgvCarrito.Rows(thisNDX)
h = 0
x = e.MarginBounds.Left - 70
i = 0
For Each cell As DataGridViewCell In row.Cells
    If cell.Visible Then
        If i = 0 Then
            k = -65
            rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
            e.Graphics.DrawRectangle(Pens.Black, rc)
            fmt.Alignment = StringAlignment.Near
            rc.Offset(5, 0)
            e.Graphics.DrawString(j + 1, DgvCarrito.Font, Brushes.Black, rc, fmt)
            x += rc.Width

```

```

        k = -20
    End If

    If i = 1 Then k = +200
    If i = 2 Then k = +20
    If i = 3 Then k = -10
    If i = 4 Then
        k = -20

        e.Graphics.DrawString("", font, Brushes.Black, rc36, fmt)
        e.Graphics.DrawString("", font, Brushes.Black, rc37, fmt)
        e.Graphics.DrawString("", font, Brushes.Black, rc35, fmt)

        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        e.Graphics.DrawString(txtDescuento.Text, DgvCarrito.Font, Brushes.Black, rc, fmt)
        x += rc.Width
        k = -5
        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        Dim divisor As String = txtDescuento.Text.Substring(0, txtDescuento.Text.IndexOf("%"))
        e.Graphics.DrawString(Integer.Parse(cell.FormattedValue) * (100 - divisor) \ 100,
            DgvCarrito.Font, Brushes.Black, rc, fmt)
        total = total + Integer.Parse(cell.FormattedValue) * (100 - divisor) \ 100

        'txtTotal.Text = total
    Else
        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        e.Graphics.DrawString(cell.FormattedValue.ToString(),
            DgvCarrito.Font, Brushes.Black, rc, fmt)
    End If
    x += rc.Width
    h = Math.Max(h, rc.Height)
    i = i + 1
End If
Next
y += h
mRow = thisNDX + 1

j = j + 1
Next
e.Graphics.DrawString(total, font, Brushes.Black, rc37, fmt)
e.Graphics.DrawString((total * 81) \ 100, font, Brushes.Black, rc35, fmt)
e.Graphics.DrawString(total * 19 \ 100, font, Brushes.Black, rc36, fmt)
Catch ex As Exception

End Try
End Sub

Private Sub CheckBoxFactura_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBoxFactura.CheckedChanged
    seleccion = "Factura"
    CheckBoxBoleta.CheckState = False
End Sub

```

```

Private Sub CheckBoxBoleta_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBoxBoleta.CheckedChanged
    seleccion = "Boleta"
    CheckBoxFactura.CheckState = False
End Sub

```

```

Private Sub PrintDocument2_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument2.PrintPage

```

```

    Dim newpage As Boolean = True
    Dim fmt As StringFormat = New StringFormat(StringFormatFlags.LineLimit)
    fmt.LineAlignment = StringAlignment.Center
    fmt.Trimming = StringTrimming.EllipsisCharacter
    Dim y As Int32 = e.MarginBounds.Top + 60
    Dim rc As Rectangle

```

```

    Dim x As Int32
    Dim h As Int32 = 0
    Dim row As DataGridViewRow
    'If pagenumber = 1 Then
    Dim font As Font = New Font("Arial", 9)
    Dim font2 As Font = New Font("Arial", 16)

```

```

    Dim rc2 As RectangleF = New RectangleF(180, 100, 500, 900)
    e.Graphics.DrawRectangle(Pens.Black, rc2.Left, rc2.Top, rc2.Width, rc2.Height)
    e.Graphics.DrawString(" ", font2, Brushes.Black, rc2, fmt)

```

```

    Dim rc3 As RectangleF = New RectangleF(180, 110, 200, 20)
    e.Graphics.DrawRectangle(Pens.Black, rc3.Left, rc3.Top, rc3.Width, rc3.Height)
    e.Graphics.DrawString("Autosave", font2, Brushes.Black, rc3, fmt)

```

```

'End If
row = DgvProductos.Rows(mRow)
x = e.MarginBounds.Left - 50
Dim i As Integer = 0
Dim k As Integer = 0
For Each cell As DataGridViewCell In row.Cells

```

```

    If cell.Visible Then
        If i = 0 Then k = -20
        If i = 1 Then k = -10
        If i = 2 Then k = -10
        If i = 3 Then k = 0
        If i = 4 Then k = -35
        If i = 5 Then k = -35
        If i = 6 Then k = -40
        If i = 7 Then k = -40
        If i = 8 Then k = +70
        If i = 1 Then
            rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
        ElseIf i = 2 Then
            rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
        ElseIf i = 3 Then
            rc = New Rectangle(x + -180, y, 100 + k, cell.Size.Height)
        Else
            rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        End If

```

```

        e.Graphics.FillRectangle(Brushes.LightGray, rc)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        If i = 0 Then
            e.Graphics.DrawString("Cod Producto", DgvProductos.Font, Brushes.Black, rc, fmt)
        ElseIf i = 1 Then
            e.Graphics.DrawString("Marca", DgvProductos.Font, Brushes.Black, rc, fmt)
        ElseIf i = 2 Then
            e.Graphics.DrawString("Categoria", DgvProductos.Font, Brushes.Black, rc, fmt)

```

```

Elseif i = 4 Then
    e.Graphics.DrawString("P.Compra", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 5 Then
    e.Graphics.DrawString("P.Venta", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 6 Then
    e.Graphics.DrawString("S.Minimo", DgvProductos.Font, Brushes.Black, rc, fmt)
Elseif i = 7 Then
    e.Graphics.DrawString("S.Actual", DgvProductos.Font, Brushes.Black, rc, fmt)
Else
    e.Graphics.DrawString(DgvProductos.Columns(cell.ColumnIndex).HeaderText,
        DgvProductos.Font, Brushes.Black, rc, fmt)
End If
x += rc.Width
h = Math.Max(h, rc.Height)
End If
i = i + 1
k = 0
Next
y += h
If pagenumber <= DgvProductos.RowCount Then
    Dim rc5 As RectangleF = New RectangleF(e.MarginBounds.Right - 40, e.MarginBounds.Bottom, 40, 30)
    e.Graphics.DrawRectangle(Pens.White, rc5.Left, rc5.Top, rc5.Width, rc5.Height)
    e.Graphics.DrawString("PAG " & pagenumber, Font, Brushes.Black, rc5, fmt)
    Dim thisNDX As Int32
    For thisNDX = mRow To DgvProductos.RowCount - 1
        If DgvProductos.Rows(thisNDX).IsNewRow Then Exit For
        row = DgvProductos.Rows(thisNDX)
        h = 0
        x = e.MarginBounds.Left - 50
        i = 0
        For Each cell As DataGridViewCell In row.Cells
            If cell.Visible Then
                If i = 0 Then k = -20
                If i = 1 Then k = -10
                If i = 2 Then k = -10
                If i = 3 Then k = 0
                If i = 4 Then k = -35
                If i = 5 Then k = -35
                If i = 6 Then k = -40
                If i = 7 Then k = -40
                If i = 8 Then k = +70
                If i = 1 Then
                    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
                Elseif i = 2 Then
                    rc = New Rectangle(x + 100, y, 100 + k, cell.Size.Height)
                Elseif i = 3 Then
                    rc = New Rectangle(x + -180, y, 100 + k, cell.Size.Height)
                Else
                    rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
                End If
                e.Graphics.DrawRectangle(Pens.Black, rc)
                fmt.Alignment = StringAlignment.Near
                rc.Offset(5, 0)
                If i = 1 Then
                    'Dim nombre As String
                    'nombre = BsnNegocios.GetMarca2(cell.FormattedValue.ToString())
                    'e.Graphics.DrawString(nombre, DgvProductos.Font, Brushes.Black, rc, fmt)
                Elseif i = 2 Then
                    'Dim nombre As String
                    'nombre = BsnNegocios.GetCategoria(cell.FormattedValue.ToString())
                    'e.Graphics.DrawString(nombre, DgvProductos.Font, Brushes.Black, rc, fmt)
                Else
                    'e.Graphics.DrawString(cell.FormattedValue.ToString(),
                    'DgvProductos.Font, Brushes.Black, rc, fmt)
                End If
            End If
        Next
    Next
End If

```

```

        x += rc.Width
        h = Math.Max(h, rc.Height)
    End If
    i = i + 1
Next
y += h
mRow = thisNDX + 1
If (mRow = 36 * pagenumber) Then
    e.HasMorePages = True
    pagenumber += 1
    Exit Sub
End If
Next
End If
e.HasMorePages = False
End Sub
Private Sub BtnBoleta_Click(sender As Object, e As EventArgs) Handles BtnBoleta.Click
    pagenumber = 1
    mRow = 0
    PrintPreviewDialog2.Document = PrintDocument2
    PrintPreviewDialog2.ShowDialog()
End Sub
End Class
4- COMPRA

```

```

Imports System.IO
Imports System.Data.OleDb
Imports MySql.Data.MySqlClient
Public Class FrmCompraRealizarOrden
    Dim BsnNegocios As New BsnNegocios
    Dim sw As Integer
    Dim pagenumber As Integer = 1
    Dim mRow As Integer = 0
    Dim imprimir As Integer = 0

    Private Sub FrmOrdenCompra_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        CargarPedidos()
        Me.BackColor = FrmPrincipal.BackColor
        txtRutVendedor.Text = FrmPrincipal.TextBox1.Text
        txtNombreVendedor.Text = FrmPrincipal.TextBox2.Text
        txtApellidoVendedor.Text = FrmPrincipal.TextBox3.Text
    End Sub
    Function CargarPedidos()
        Dim conn As MySqlConnection
        conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
        Dim username As Boolean = True
        conn.Open()
        Dim sqlquery As String = "select Nro_Pedido,Rut_Proveedor,Fecha,Valor,Estado from pedido"
        Dim adapter As New MySqlDataAdapter
        Dim command As New MySqlCommand
        Dim dataset As New DataSet
        command.CommandText = sqlquery
        command.Connection = conn
        adapter.SelectCommand = command
        adapter.Fill(DataSet1.DataTableOrdenDeCompra)
        conn.Close()
    End Function
    Function CargarPedidoFiltrado(filtro As String)
        Dim conn As MySqlConnection
        conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
        Dim username As Boolean = True
        conn.Open()
        Dim sqlquery As String = "select Nro_Pedido,Rut_Proveedor,Fecha,Valor,Estado from pedido where Estado='" & filtro & "'"
    End Function

```

```

Dim adapter As New MySqlDataAdapter
Dim command As New MySqlCommand
Dim dataset As New DataSet
command.CommandText = sqlquery
command.Connection = conn
adapter.SelectCommand = command
adapter.Fill(DataSet1.DataTableOrdenDeCompra)
conn.Close()

```

End Function

Function CargarDetallePedidos(filtro As Integer)

```

Dim conn As MySqlConnection
conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
Dim username As Boolean = True
conn.Open()
Dim sqlquery As String = "select Nro_pedido,Codigo_Producto,Cantidad from detalle_pedido where Nro_Pedido=" & filtro & ""
Dim adapter As New MySqlDataAdapter
Dim command As New MySqlCommand
Dim dataset As New DataSet
command.CommandText = sqlquery
command.Connection = conn
adapter.SelectCommand = command
adapter.Fill(DataSet1.DataTableDetallePedido)
conn.Close()

```

End Function

Private Sub BtnFiltroHistorico_Click(sender As Object, e As EventArgs) Handles BtnFiltroHistorico.Click

```

DataSet1.Clear()
CargarPedidos()

```

End Sub

Private Sub BtnFiltroEspera_Click(sender As Object, e As EventArgs) Handles BtnFiltroEspera.Click

```

DataSet1.Clear()
CargarPedidoFiltrado("En espera")

```

End Sub

Private Sub BtnFiltroCompletado_Click(sender As Object, e As EventArgs) Handles BtnFiltroCompletado.Click

```

DataSet1.Clear()
CargarPedidoFiltrado("Completado")

```

End Sub

Private Sub BtnFiltroCancelado_Click(sender As Object, e As EventArgs) Handles BtnFiltroCancelado.Click

```

DataSet1.Clear()
CargarPedidoFiltrado("Cancelado")

```

End Sub

Private Sub DgvProductos_RowHeaderMouseClick(sender As Object, e As DataGridViewCellEventArgs) Handles DgvProductos.RowHeaderMouseClick

```

Dim row2 As DataGridViewRow = DgvProductos.CurrentRow
sw = 2

```

```

If sw = 2 Or sw = 3 Or sw = 4 Then

```

```

    If row2.Cells(0).Value.Equals(DBNull.Value) Then

```

```

        TxtCodpedido.Text = ""

```

```

        TxtProveedor.Text = ""

```

```

        Txtfecha.Text = ""

```

```

        TxTestado.Text = ""

```

```

        Txtfecha.Text = ""

```

```

    Else

```

```

        TxtCodpedido.Text = row2.Cells(0).Value

```

```

        TxtProveedor.Text = row2.Cells(1).Value

```

```

        Txtfecha.Text = row2.Cells(2).Value

```

```

        TxTestado.Text = row2.Cells(3).Value

```

```

        txtmonto.Text = row2.Cells(4).Value

```

```

        DataSet1.DataTableDetallePedido.Clear()
        CargarDetallePedidos(row2.Cells(0).Value)
    End If
End If
End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    FrmPrincipal.Show()
    FrmPrincipal.TxTcaja.Text = BsnNegocios.getMontoDinero
    Me.Dispose()
End Sub

Private Sub BtnAgregar_Click(sender As Object, e As EventArgs) Handles BtnAgregar.Click
    FrmCompraAgregar.Show()
    Me.Dispose()
End Sub

Private Sub BtnCompletar_Click(sender As Object, e As EventArgs) Handles BtnCompletar.Click
    If TxTestado.Text.Equals("En espera") = True Then
        If MessageBox.Show("El stock de productos sera ajustado", "¿Estas seguro de completar este pedido?",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.Yes Then
            Dim Query As String
            Dim con As MySqlConnection = New MySqlConnection("Data Source=localhost;Database=autosave;User
            ID=root;Password=;")
            con.Open()
            Query = "UPDATE pedido SET Estado = 'Completado'"
            Query = Query + " WHERE Nro_Pedido = " & TxtCodpedido.Text
            Dim cmd As MySqlCommand = New MySqlCommand(Query, con)
            Dim i As Integer = cmd.ExecuteNonQuery()
            con.Close()
            For i = 0 To DgvDetallePedido.RowCount - 1
                Dim val1 As String = DgvDetallePedido.Rows(i).Cells(1).Value
                Dim val2 As String = DgvDetallePedido.Rows(i).Cells(2).Value
                Dim val3 As String = FechaTxT.Value.ToString
                Dim val4 As String = TxtCodpedido.Text

                con.Open()
                Query = "UPDATE productos SET Stock_Actual = (Stock_Actual + " & val2 & ")"
                Query = Query + " WHERE codigo_Producto = " & val1
                Dim cmd1 As MySqlCommand = New MySqlCommand(Query, con)
                cmd1.ExecuteNonQuery()
                con.Close()
                con.Open()
                Query = "insert into ajuste_stock (Codigo_Producto,Cantidad,Fecha,Tipo,Descripcion) values (" & val1 & ", " & val2 & ", " &
                val3 & ", 'entrada', 'entrada por orden de compra: " & val4 & "')"
                Dim cmd2 As MySqlCommand = New MySqlCommand(Query, con)
                cmd2.ExecuteNonQuery()
                con.Close()
            Next
            MessageBox.Show("Este pedido ha sido completado y productos actualizados", "", MessageBoxButtons.OK,
            MessageBoxIcon.Information)
            DataSet1.Clear()
            CargarPedidos()
            Dim v_tipo As String = "Pago orden de compra: " & TxtCodpedido.Text

            BsnNegocios.insertarMontoPedido(txtRutVendedor.Text, Txtfecha.Text, v_tipo, "-" & txtmonto.Text)

        End If

    Else
        If TxTestado.Text.Equals("Completado") = True Then
            MessageBox.Show("Este pedido ya ha sido completado", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    End If

```

```

    If TxTestado.Text.Equals("Completado") = True Then
        MessageBox.Show("Este pedido ya ha sido cancelado", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End If
End Sub

Private Sub BtnCancelar_Click(sender As Object, e As EventArgs) Handles BtnCancelar.Click
    If TxTestado.Text.Equals("En espera") = True Then

        If MessageBox.Show("¿Esta seguro de cancelar este pedido?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
DialogResult.Yes Then
            Dim Query As String
            Dim con As MySqlConnection = New MySqlConnection("Data Source=localhost;Database=autosave;User
ID=root;Password=;")
            con.Open()
            Query = "UPDATE pedido SET Estado = 'Cancelado'"
            Query = Query + " WHERE Nro_Pedido = " & TxtCodpedido.Text
            Dim cmd As MySqlCommand = New MySqlCommand(Query, con)
            Dim i As Integer = cmd.ExecuteNonQuery()
            con.Close()
            DataSet1.Clear()
            CargarPedidos()
            MessageBox.Show("Este pedido ha sido cancelado", "", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    Else
        If TxTestado.Text.Equals("Cancelado") = True Then
            MessageBox.Show("Este pedido ya ha sido completado", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
        If TxTestado.Text.Equals("Completado") = True Then
            MessageBox.Show("Este pedido no se puede cancelar, pues ya ha sido completado", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)

        End If
    End If
End Sub

Private Sub BtnEliminar_Click(sender As Object, e As EventArgs) Handles BtnEliminar.Click
    If TxtCodpedido.Text.Equals("") = False Then
        If MessageBox.Show("Estas seguro de eliminar este pedido, sera borrado del registro historico", "", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = DialogResult.Yes Then
            Dim val As String = TxtCodpedido.Text
            Dim Query As String
            Dim con As MySqlConnection = New MySqlConnection("Data Source=localhost;Database=autosave;User
ID=root;Password=;")
            con.Open()
            Query = "delete from pedido where Nro_Pedido = " & val & ""
            Dim cmd As MySqlCommand = New MySqlCommand(Query, con)
            Dim i As Integer = cmd.ExecuteNonQuery()
            con.Close()
            Dim orden As New FrmCompraRealizarOrden
            orden.Show()
            Me.Dispose()

        End If
    Else
        MessageBox.Show("Primero seleccione un registro", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    LblFechaYHora.Text = "Hora actual: " & Date.Now.ToLongTimeString
End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs)
    FrmCompraStockBajo.Show()
    Me.Dispose()
End Sub

Private Sub BtnRevisarBajoStock_Click(sender As Object, e As EventArgs) Handles BtnRevisarBajoStock.Click
    FrmCompraStockBajo.Show()
    Me.Hide()
End Sub

Private Sub BtnImprimir_Click(sender As Object, e As EventArgs) Handles BtnImprimir.Click
    pagenumber = 1
    mRow = 0
    imprimir = 1
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub BtnImprimir2_Click(sender As Object, e As EventArgs) Handles BtnImprimir2.Click
    pagenumber = 1
    mRow = 0
    imprimir = 2
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()
End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    Dim newpage As Boolean = True
    Dim fmt As StringFormat = New StringFormat(StringFormatFlags.LineLimit)
    fmt.LineAlignment = StringAlignment.Center
    fmt.Trimming = StringTrimming.EllipsisCharacter
    Dim y As Int32 = e.MarginBounds.Top + 60
    Dim rc As Rectangle
    Dim x As Int32
    Dim h As Int32 = 0
    Dim row As DataGridViewRow
    If pagenumber = 1 Then
        Dim font As Font = New Font("Arial", 9)
        Dim font2 As Font = New Font("Arial", 16)
        Dim rc2 As RectangleF = New RectangleF(200, 30, 300, 30)
        e.Graphics.DrawRectangle(Pens.White, rc2.Left, rc2.Top, rc2.Width, rc2.Height)
        e.Graphics.DrawString("DETALLES DEL PEDIDO", font2, Brushes.Black, rc2, fmt)
        Dim rc3 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 100, 400, 30)
        e.Graphics.DrawRectangle(Pens.White, rc3.Left, rc3.Top, rc3.Width, rc3.Height)
        e.Graphics.DrawString("FECHA DE IMPRESIÓN: " & DateTime.Now.ToString("dd-MM-yyyy"), font, Brushes.Black, rc3, fmt)
        Dim rc4 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 125, 400, 30)
        e.Graphics.DrawRectangle(Pens.White, rc4.Left, rc4.Top, rc4.Width, rc4.Height)
        e.Graphics.DrawString("HORA DE IMPRESIÓN: " & DateTime.Now.ToString("HH:mm:ss"), font, Brushes.Black, rc4, fmt)
        If imprimir = 1 Then ' pedido

        End If
    End If
    If imprimir = 1 Then
        row = DgvProductos.Rows(mRow)
    Elseif imprimir = 2 Then
        row = DgvDetallePedido.Rows(mRow)
    End If

    x = e.MarginBounds.Left - 50
    Dim i As Integer = 0
    Dim k As Integer = 0
    For Each cell As DataGridViewCell In row.Cells

```

```

If cell.Visible Then
  If i = 0 Then k = +10
  If i = 1 Then k = +10
  If i = 2 Then k = +10
  If i = 3 Then k = +10
  If i = 4 Then k = +10
  rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
  e.Graphics.FillRectangle(Brushes.LightGray, rc)
  e.Graphics.DrawRectangle(Pens.Black, rc)
  fmt.Alignment = StringAlignment.Near
  rc.Offset(5, 0)
  If imprimir = 1 Then
    e.Graphics.DrawString(DgvProductos.Columns(cell.ColumnIndex).HeaderText,
      DgvProductos.Font, Brushes.Black, rc, fmt)
  Elseif imprimir = 2 Then

    e.Graphics.DrawString(DgvDetallePedido.Columns(cell.ColumnIndex).HeaderText,
      DgvProductos.Font, Brushes.Black, rc, fmt)

  End If
  x += rc.Width
  If (i = 1) And (imprimir = 2) Then
    rc = New Rectangle(x, y, 150 + k, cell.Size.Height)
    e.Graphics.FillRectangle(Brushes.LightGray, rc)
    e.Graphics.DrawRectangle(Pens.Black, rc)
    fmt.Alignment = StringAlignment.Near
    rc.Offset(5, 0)
    e.Graphics.DrawString("Nombre Producto", DgvDetallePedido.Font, Brushes.Black, rc, fmt)
    x += rc.Width
  End If
  If (i = 2) And (imprimir = 2) Then
    rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
    e.Graphics.FillRectangle(Brushes.LightGray, rc)
    e.Graphics.DrawRectangle(Pens.Black, rc)
    fmt.Alignment = StringAlignment.Near
    rc.Offset(5, 0)
    e.Graphics.DrawString("Precio Productos", DgvDetallePedido.Font, Brushes.Black, rc, fmt)
    x += rc.Width
  End If
  h = Math.Max(h, rc.Height)
End If
i = i + 1
k = 0
Next
y += h
If (pagenumber <= DgvProductos.RowCount And imprimir = 1) Or (pagenumber <= DgvDetallePedido.RowCount And imprimir =
2) Then
  Dim rc5 As RectangleF = New RectangleF(e.MarginBounds.Right - 40, e.MarginBounds.Bottom, 40, 30)
  e.Graphics.DrawRectangle(Pens.White, rc5.Left, rc5.Top, rc5.Width, rc5.Height)
  e.Graphics.DrawString("PAG " & pagenumber, Font, Brushes.Black, rc5, fmt)
  Dim thisNDX As Int32
  Dim num As Integer
  If imprimir = 1 Then
    num = DgvProductos.RowCount - 1
  Elseif imprimir = 2 Then
    num = DgvDetallePedido.RowCount - 1
  End If
  For thisNDX = mRow To num
    If imprimir = 1 Then
      If DgvProductos.Rows(thisNDX).IsNewRow Then Exit For
      row = DgvProductos.Rows(thisNDX)
    Elseif imprimir = 2 Then
      If DgvDetallePedido.Rows(thisNDX).IsNewRow Then Exit For
      row = DgvDetallePedido.Rows(thisNDX)
    End If
  End If

```

```

h = 0
x = e.MarginBounds.Left - 50
i = 0
Dim sum1 As Integer
Dim codigo As Integer
For Each cell As DataGridViewCell In row.Cells
    If cell.Visible Then
        If i = 0 Then k = +10
        If i = 1 Then k = +10
        If i = 2 Then k = +10
        If i = 3 Then k = +10
        If i = 4 Then k = +10
        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        If (i = 1) And (imprimir = 1) Then
            If (cell.FormattedValue.ToString() = "10") Then
                If imprimir = 1 Then
                    e.Graphics.DrawString("K",
                        DgvProductos.Font, Brushes.Black, rc, fmt)
                Elseif imprimir = 2 Then
                    e.Graphics.DrawString("K",
                        DgvDetallePedido.Font, Brushes.Black, rc, fmt)
                End If
            End If
        Else
            If imprimir = 1 Then
                e.Graphics.DrawString(cell.FormattedValue.ToString(),
                    DgvProductos.Font, Brushes.Black, rc, fmt)
            Elseif imprimir = 2 Then
                e.Graphics.DrawString(cell.FormattedValue.ToString(),
                    DgvDetallePedido.Font, Brushes.Black, rc, fmt)
            End If
        End If
    Else
        If imprimir = 1 Then
            e.Graphics.DrawString(cell.FormattedValue.ToString(),
                DgvProductos.Font, Brushes.Black, rc, fmt)
        Elseif imprimir = 2 Then
            e.Graphics.DrawString(cell.FormattedValue.ToString(),
                DgvDetallePedido.Font, Brushes.Black, rc, fmt)
        End If
    End If
    x += rc.Width
    If (i = 1) And (imprimir = 2) Then
        codigo = cell.FormattedValue.ToString()
        rc = New Rectangle(x, y, 150 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        Dim nombre As String
        nombre = BsnNegocios.GetNombreProducto(codigo)
        e.Graphics.DrawString(nombre, DgvDetallePedido.Font, Brushes.Black, rc, fmt)
        x += rc.Width
    End If
    If (i = 2) And (imprimir = 2) Then
        sum1 = cell.FormattedValue.ToString()
        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        Dim Valor As Integer
        Valor = BsnNegocios.GetValorProducto(codigo)

```

```

        e.Graphics.DrawString(Valor * sum1.ToString(),
            DgvDetallePedido.Font, Brushes.Black, rc, fmt)

    End If
    h = Math.Max(h, rc.Height)
End If
i = i + 1
Next
mRow = thisNdx + 1
If (mRow = 36 * pagenumber) Then
    e.HasMorePages = True
    pagenumber += 1
Exit Sub
End If
Next
End If
Dim rc6 As RectangleF = New RectangleF(e.MarginBounds.Right - 300, y + 5, 200, 30)
e.Graphics.DrawRectangle(Pens.White, rc6.Left, rc6.Top, rc6.Width, rc6.Height)
e.Graphics.DrawString("PRECIO TOTAL: " & txtmonto.Text, Font, Brushes.Black, rc6, fmt)
Dim rc7 As RectangleF = New RectangleF(e.MarginBounds.Right - 300, y + 35, 200, 1)
e.Graphics.DrawRectangle(Pens.Black, rc7.Left, rc7.Top, rc7.Width, rc7.Height)
e.HasMorePages = False
End Sub

Private Sub GroupBox3_Enter(sender As Object, e As EventArgs) Handles GroupBox3.Enter

End Sub
End Class

```

5- AJUSTE DE STOCK

```
Imports MySql.Data.MySqlClient
```

```
Public Class FrmAjustarStock
```

```
    Dim sw As Integer = 0
```

```
    Dim BsnNegocios As New BsnNegocios
```

```
    Dim pagenumber As Integer = 1
```

```
    Dim mRow As Integer = 0
```

```
    Private Sub FrmAjustarStock_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        CargarAjustes()
```

```
        Me.BackColor = FrmPrincipal.BackColor
```

```
        CargarProductosBajoStock()
```

```
    End Sub
```

```
*****
****
```

```
Function CargarProductosBajoStock() ' ***** CARGAR PRODUCTOS EN EL DATASET
```

```
    DataSet1.DataTableProductoBajoStock.Clear() ' ***** LIMPIAR EL DATASET DE LOS DATOS QUE CONTENGA
```

```
    Dim conn As MySqlConnection
```

```
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
```

```
    Dim username As Boolean = True
```

```
    conn.Open()
```

```

    Dim sqlquery As String = "select Codigo_Producto,Nombre,Stock_Minimo,Stock_actual from Productos where Stock_Actual <
Stock_Minimo"
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableProductoBajoStock)
    conn.Close()
End Function

Function CargarAjustes()
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Codigo_Producto,Cantidad,Fecha,Tipo,Descripcion from ajuste_stock"
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    Dim dataset As New DataSet
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableAjusteStock)
    conn.Close()
End Function

Function CargarAjustesFiltrados(filtro As String)
    Dim conn As MySqlConnection
    conn = New MySqlConnection("server=localhost; Port=3306; user id=root; password=; database=autosave")
    Dim username As Boolean = True
    conn.Open()
    Dim sqlquery As String = "select Codigo_Producto,Cantidad,Fecha,Tipo,Descripcion from ajuste_stock where Codigo_Producto="
    & filtro & ""
    Dim adapter As New MySqlDataAdapter
    Dim command As New MySqlCommand
    Dim dataset As New DataSet
    command.CommandText = sqlquery
    command.Connection = conn
    adapter.SelectCommand = command
    adapter.Fill(DataSet1.DataTableAjusteStock)
    conn.Close()
End Function

Private Sub Txtfiltro_TextChanged(sender As Object, e As EventArgs) Handles Txtfiltro.TextChanged
    Try
        DataSet1.Clear()
        CargarAjustesFiltrados(Txtfiltro.Text)
    Catch ex As Exception

    End Try
    If Txtfiltro.Text.Equals("") = True Then
        CargarAjustes()
    End If
End Sub

Private Sub BtnSalir_Click(sender As Object, e As EventArgs) Handles BtnSalir.Click
    FrmPrincipal.Show()
    FrmPrincipal.TxTcaja.Text = BsnNegocios.getMontoDinero
    Me.Dispose()
End Sub

Private Sub DgvProductos_RowHeaderMouseClick(sender As Object, e As DataGridViewCellMouseEventArgs) Handles
DgvProductos.RowHeaderMouseClick
    Dim row2 As DataGridViewRow = DgvProductos.CurrentRow

```

```

If sw = 0 Or sw = 2 Then
    If row2.Cells(0).Value.Equals(DBNull.Value) Then
        TxtCodProducto.Text = ""
        TxtValor.Text = ""
        Txtfecha.Text = ""
        TxTtipo.Text = ""
        TxtDescripcion.Text = ""
    Else
        TxtCodProducto.Text = row2.Cells(0).Value
        TxtValor.Text = row2.Cells(1).Value
        Txtfecha.Text = row2.Cells(2).Value
        TxTtipo.Text = row2.Cells(3).Value
        TxtDescripcion.Text = row2.Cells(4).Value
    End If
End If

End Sub

Private Sub BtnAceptar_Click(sender As Object, e As EventArgs) Handles BtnAceptar.Click
    If sw = 1 Then
        If TxtCodProducto.Text.Equals("") = False Then
            If NumericUpDown1.Text.Equals("") = False Then
                If TxTtipo.Text.Equals("") = False Then
                    If TxtDescripcion.Text.Equals("") = False Then
                        If MessageBox.Show("¿Esta seguro de realizar este ajuste?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.Yes Then
                            Try
                                insertarAjuste()
                                MessageBox.Show("Ajuste realizado correctamente", "", MessageBoxButtons.OK, MessageBoxIcon.Information)
                                DataSet1.Clear()
                                CargarAjustes()
                                GrupoAjuste.Enabled = False
                                TxtCodProducto.Text = ""
                                NumericUpDown1.Visible = False
                                DateTimePicker1.Visible = False
                                TxtValor.Visible = True
                                Txtfecha.Visible = True
                                Txtfecha.Text = ""
                                TxTtipo.Text = ""
                                TxtValor.Text = ""
                                TxtDescripcion.Text = ""
                                BtnAceptar.Visible = False
                                BtnCancelar.Visible = False
                                BtnAgregar.Visible = True
                                BtnEliminar.Visible = True
                                sw = 0
                            Catch ex As Exception
                                MessageBox.Show("Codigo de producto invalido", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
                            End Try
                        End If
                    End If
                Else
                    MessageBox.Show("Primero Ingrese un tipo", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
                End If
            Else
                MessageBox.Show("Primero Ingrese un valor distinto de 0", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End If
        Else
            MessageBox.Show("Primero Ingrese un codigo de producto", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    End If
    If sw = 2 Then
        If TxtCodProducto.Text.Equals("") = False Then

```

```

    eliminarAjuste()
    DataSet1.Clear()
    CargarAjustes()
    MessageBox.Show("Registro eliminado con exito", "", MessageBoxButtons.OK, MessageBoxIcon.Information)
    GrupoAjuste.Enabled = False
    TxtCodProducto.Text = ""
    NumericUpDown1.Value = 0
    NumericUpDown1.Visible = False
    DateTimePicker1.Visible = False
    TxtValor.Visible = True
    Txtfecha.Visible = True
    TxtValor.Text = ""
    Txtfecha.Text = ""
    TxTtipo.Text = ""
    TxtDescripcion.Text = ""
    BtnAceptar.Visible = False
    BtnCancelar.Visible = False
    BtnAgregar.Visible = True
    BtnEliminar.Visible = True
    sw = 0
Else
    MessageBox.Show("Primero seleccione un registro", "", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
End If
End Sub
Function insertarAjuste()
    Dim Query As String
    Dim con As MySqlConnection = New MySqlConnection("Data Source=localhost;Database=autosave;User ID=root;Password=;")
    con.Open()
    Query = "UPDATE productos SET Stock_Actual = (Stock_Actual + " & NumericUpDown1.Value.ToString & ")"
    Query = Query + " WHERE Codigo_Producto = " & TxtCodProducto.Text & ""
    Dim cmd1 As MySqlCommand = New MySqlCommand(Query, con)
    cmd1.ExecuteNonQuery()
    con.Close()
    con.Open()
    Query = "insert into ajuste_stock (Codigo_Producto,Cantidad,Fecha,Tipo,Descripcion) values (" & TxtCodProducto.Text & "," &
NumericUpDown1.Text & "," & DateTimePicker1.Value.ToString & "," & TxTtipo.Text & "," & TxtDescripcion.Text & """)
    Dim cmd As MySqlCommand = New MySqlCommand(Query, con)
    cmd.ExecuteNonQuery()
    con.Close()
End Function
Function eliminarAjuste()
    Dim Query As String
    Dim con As MySqlConnection = New MySqlConnection("Data Source=localhost;Database=autosave;User ID=root;Password=;")
    con.Open()
    Query = "delete from ajuste_stock where Codigo_Producto =" & TxtCodProducto.Text & " and Fecha =" & Txtfecha.Text & ""
    Dim cmd1 As MySqlCommand = New MySqlCommand(Query, con)
    cmd1.ExecuteNonQuery()
    con.Close()
End Function

Private Sub TxtDescripcion_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtDescripcion.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Char.IsDigit(e.KeyChar)
    End If
End Sub
Private Sub TxTtipo_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxTtipo.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Char.IsDigit(e.KeyChar)
    End If
End Sub
Private Sub TxTvalor_KeyPress(sender As Object, e As KeyPressEventArgs)

    If e.KeyChar <> ControlChars.Back Then

```

```

        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub
Private Sub TxtCodProducto_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TxtCodProducto.KeyPress
    If e.KeyChar <> ControlChars.Back Then
        e.Handled = Not Char.IsDigit(e.KeyChar)
    End If
End Sub
Private Sub BtnAgregar_Click(sender As Object, e As EventArgs) Handles BtnAgregar.Click
    GrupoAjuste.Enabled = True
    TxtCodProducto.Text = ""
    NumericUpDown1.Text = ""
    DateTimePicker1.Visible = True
    NumericUpDown1.Visible = True
    TxtValor.Visible = False
    Txtfecha.Visible = False
    TxTtipo.Text = ""
    TxtDescripcion.Text = ""
    BtnAceptar.Visible = True
    BtnCancelar.Visible = True
    BtnAgregar.Visible = False
    BtnEliminar.Visible = False
    sw = 1
End Sub

Private Sub BtnEliminar_Click(sender As Object, e As EventArgs) Handles BtnEliminar.Click
    DateTimePicker1.Visible = False
    NumericUpDown1.Visible = False
    TxtValor.Visible = True
    Txtfecha.Visible = True
    BtnAceptar.Visible = True
    BtnCancelar.Visible = True
    BtnAceptar.Enabled = True
    BtnCancelar.Enabled = True
    BtnAgregar.Visible = False
    BtnEliminar.Visible = False
    sw = 2
End Sub

Private Sub BtnCancelar_Click(sender As Object, e As EventArgs) Handles BtnCancelar.Click
    GrupoAjuste.Enabled = False
    TxtCodProducto.Text = ""
    Txtfecha.Text = ""
    TxTtipo.Text = ""
    NumericUpDown1.Visible = False
    TxtValor.Visible = True
    TxtDescripcion.Text = ""
    BtnAceptar.Visible = False
    BtnCancelar.Visible = False
    BtnAgregar.Visible = True
    BtnEliminar.Visible = True
    sw = 0
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    LblFechaYHora.Text = "Hora actual: " & Date.Now.ToLongTimeString
End Sub

Private Sub BtnImprimir_Click(sender As Object, e As EventArgs) Handles BtnImprimir.Click
    pagenumber = 1
    mRow = 0
    PrintPreviewDialog1.Document = PrintDocument1
    PrintPreviewDialog1.ShowDialog()

```

End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

```
Dim newpage As Boolean = True
Dim fmt As StringFormat = New StringFormat(StringFormatFlags.LineLimit)
fmt.LineAlignment = StringAlignment.Center
fmt.Trimming = StringTrimming.EllipsisCharacter
Dim y As Int32 = e.MarginBounds.Top + 60
Dim rc As Rectangle
Dim x As Int32
Dim h As Int32 = 0
Dim row As DataGridViewRow
If pagenumber = 1 Then
    Dim font As Font = New Font("Arial", 9)
    Dim font2 As Font = New Font("Arial", 16)
    Dim rc2 As RectangleF = New RectangleF(200, 30, 600, 30)
    e.Graphics.DrawRectangle(Pens.White, rc2.Left, rc2.Top, rc2.Width, rc2.Height)
    e.Graphics.DrawString("INFORME DE AJUSTES DE STOCK", font2, Brushes.Black, rc2, fmt)
    Dim rc3 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 100, 400, 30)
    e.Graphics.DrawRectangle(Pens.White, rc3.Left, rc3.Top, rc3.Width, rc3.Height)
    e.Graphics.DrawString("FECHA DE IMPRESIÓN: " & DateTime.Now.ToString("dd-MM-yyyy"), font, Brushes.Black, rc3, fmt)
    Dim rc4 As RectangleF = New RectangleF(e.MarginBounds.Left - 50, 125, 400, 30)
    e.Graphics.DrawRectangle(Pens.White, rc4.Left, rc4.Top, rc4.Width, rc4.Height)
    e.Graphics.DrawString("HORA DE IMPRESIÓN: " & DateTime.Now.ToString("HH:mm:ss"), font, Brushes.Black, rc4, fmt)
End If
row = DgvProductos.Rows(mRow)
x = e.MarginBounds.Left - 50
Dim i As Integer = 0
Dim k As Integer = 0
For Each cell As DataGridViewCell In row.Cells
    If cell.Visible Then
        If i = 0 Then k = +10
        If i = 1 Then k = -10
        If i = 2 Then k = +40
        If i = 3 Then k = 0
        If i = 4 Then k = +40
        rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
        e.Graphics.FillRectangle(Brushes.LightGray, rc)
        e.Graphics.DrawRectangle(Pens.Black, rc)
        fmt.Alignment = StringAlignment.Near
        rc.Offset(5, 0)
        e.Graphics.DrawString(DgvProductos.Columns(cell.ColumnIndex).HeaderText,
            DgvProductos.Font, Brushes.Black, rc, fmt)
        x += rc.Width
        h = Math.Max(h, rc.Height)
    End If
    i = i + 1
    k = 0
Next
y += h
If pagenumber <= DgvProductos.RowCount Then
    Dim rc5 As RectangleF = New RectangleF(e.MarginBounds.Right - 40, e.MarginBounds.Bottom, 40, 30)
    e.Graphics.DrawRectangle(Pens.White, rc5.Left, rc5.Top, rc5.Width, rc5.Height)
    e.Graphics.DrawString("PAG " & pagenumber, Font, Brushes.Black, rc5, fmt)
    Dim thisNDX As Int32
    For thisNDX = mRow To DgvProductos.RowCount - 1
        If DgvProductos.Rows(thisNDX).IsNewRow Then Exit For
        row = DgvProductos.Rows(thisNDX)
        h = 0
        x = e.MarginBounds.Left - 50
        i = 0
        For Each cell As DataGridViewCell In row.Cells
            If cell.Visible Then
                If i = 0 Then k = +10
```

```

    If i = 1 Then k = -10
    If i = 2 Then k = +40
    If i = 3 Then k = 0
    If i = 4 Then k = +40
    rc = New Rectangle(x, y, 100 + k, cell.Size.Height)
    e.Graphics.DrawRectangle(Pens.Black, rc)
    fmt.Alignment = StringAlignment.Near
    rc.Offset(5, 0)
    'If i = 1 Then
    ' Dim nombre As String
    ' nombre = BsnNegocios.GetMarca2(cell.FormattedValue.ToString())
    ' e.Graphics.DrawString(nombre, DgvProductos.Font, Brushes.Black, rc, fmt)
    ' End If

    e.Graphics.DrawString(cell.FormattedValue.ToString(),
        DgvProductos.Font, Brushes.Black, rc, fmt)
    x += rc.Width
    h = Math.Max(h, rc.Height)
End If
i = i + 1
Next
y += h
mRow = thisNDX + 1
If (mRow = 36 * pagenumber) Then
    e.HasMorePages = True
    pagenumber += 1
Exit Sub
End If
Next
End If
e.HasMorePages = False
End Sub

```

End Class