UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

SEDE VIÑA DEL MAR – JOSE MIGUEL CARRERA

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE INVENTARIO DE REPUESTOS PARA TALLER MECÁNICO

Trabajo de Titulación para optar al

Título de Técnico Universitario en

INFORMÁTICA.

Alumnos :

Diego Andrés Álvarez Jiménez (Q.E.P.D)

Vanessa Lissette Toledo Albina

Profesor Guía :

Sr. Ricardo Cahe Cabach

2022

DEDICATORIA

- Les dedico este trabajo a mis abuelos C.H.T.R (Q.E.P.D 2009) y a F.M.M(Q.E.P.D 2019) , gracias por estar a mi lado . Siempre los tengo en mente y en mis pensamientos.

- También este trabajo de título está dedicado en memoria de **DIEGO ANDRÉS ÁLVAREZ JIMÉNEZ** , quien fue compañero y amigo de la carrera de Técnico Universitario en Informática .

  Muchas gracias por haber sido un muy bien compañero , pues este trabajo de título también es tuyo y sin ti no se hubiera podido realizar , también está aquí tu esfuerzo y en todo a lo largo de la carrera .Agradezco como compañera tu buen compañerismo , generosidad , amabilidad . Nunca te olvidaremos.

  DESCANSA EN PAZ COMPAÑERO.

RESUMEN

KEYWORDS : TALLER MECÁNICO – SISTEMA DE ADMINISTRACIÓN Y CONTROL DE INVENTARIO – SOCIEDAD MECÁNICA SERVAS LIMITADA

El presente documento menciona la existencia de una sociedad llamada mecánica servas limitada compuesta por Mauricio Vasquez y Patricio Vasquez , el cual mantienen un taller mecánico ubicado en Quillota y realizan las labores de reparación , mantención de vehículos que ingresan al taller. En consecuencia , para efectuar dichas reparaciones se deben conocer los repuestos a colocar y el stock , sin embargo , si no existe un stock suficiente se debe comprar repuestos a diferentes proveedores especializados en el rubro de la venta de accesorios o repuestos de vehículos . Es por ello que , uno de los principales problemas es que se desconoce el stock de los productos , por lo cual se decidió crear e implementar un sistema de administración y control de inventario para brindar un apoyo en el control de stock de los productos almacenados . Adicionalmente para una implementación futura , también se proporciona la opcionalidad de tener un control de posibles deudores . Este sistema se realiza en base a herramientas de desarrollo de software , en donde el entorno de desarrollo se utiliza visual studio acompañado del lenguaje visual Basic , en conjunto con el motor de base de datos mysql .

Este trabajo presenta dos capítulos , el primero de ellos consiste en describir los aspectos relevantes del análisis , abarcando la descripción , situación actual del taller mecánico , problemas detectados en el funcionamiento del taller , objetivos del sistema propuesto , beneficios , funcionalidades del sistema , información que manejará el sistema , entidades de datos y condiciones de diseño . Mientras que en el capítulo 2 , consiste en describir los aspectos relevantes del diseño , abarcando la descripción de recursos computacionales , descripción de archivos , diagrama de menús , listado de totalidad de programas del sistema y diseño de pantallas generales .

Adicionando a lo anterior , este trabajo contiene imágenes y diagramas que explican el funcionamiento del taller mecánico , como también los procesos de construcción del modelo de datos y sistema .

El presente documento finaliza con las conclusiones del trabajo realizado y la bibliografía recopilando la información obtenida para el desarrollo y ejecución .

ÍNDICE

RESUMEN

INTRODUCCIÓN ……1

CAPÍTULO 1 : ASPECTOS RELEVANTES DEL ANÁLISIS.....4

1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN 5

1.2. SITUACIÓN ACTUAL 6

1.3. PROBLEMAS GENERALES DETECTADOS 12

1.4. DESCRIPCIÓN DEL SISTEMA PROPUESTO 13

1.4.1. Objetivo General 13

1.4.2. Objetivos Específicos 14

1.4.3. Beneficios del sistema 14

1.4.4. Diagrama de flujo administrativo 16

1.4.4.1. Diagrama del proceso de reparación de vehículo 16

1.4.5. Funcionalidades del sistema 18

1.5. DESCRIPCIÓN DE CÁLCULOS Y FÓRMULAS EMPLEADAS EN EL SISTEMA 23

1.6. INFORMACIÓN QUE MANEJARÁ EL SISTEMA 25

1.6.1. Entradas 25

1.6.2. Salidas 25

1.7. ENTIDADES DE DATOS 26

1.7.1. Modelo de datos 28

1.8. CONDICIONES DE DISEÑO 29

CAPITULO 2 : ASPECTOS RELEVANTES DEL DISEÑO 31

2.1. DESCRIPCIÓN DE RECURSOS COMPUTACIONALES 32

2.1.1. Configuración del sistema 32

2.1.2. Hardware usado 32

2.1.3. Software utilizado.....33

2.2. DESCRIPCIÓN DE ARCHIVOS 34

2.2.1. Descripción de tipos de datos soportados por el motor de base de datos....34

2.2.2. Descripción detallada de las tablas.....34

2.2.2.1. Usuarios.....35

2.2.2.2. Productos.....35

2.2.2.3. Proveedores.....36

2.2.2.4. Compra de Productos 37

2.2.2.5. Detalle de Compra.....38

2.2.2.6. Retiro de Productos.....39

2.2.2.7. Ajuste de Productos.....39

2.2.2.8. Vehículos.....40

2.2.2.9. Clientes.....41

2.2.2.10. Deuda.....42

2.2.2.11. Pago.....43

2.3. DIAGRAMA DE MENÚS 44

2.4. LISTADO DE TOTALIDAD DE PROGRAMAS DEL SISTEMA 45

2.5. DISEÑO DE PANTALLAS PRINCIPALES 47

2.5.1. Login 47

2.5.2. Menú principal 47

2.5.2.1. Submenú de usuarios 48

2.5.2.2 Submenú de Inventario 49

2.5.2.3 Submenú de Gestión de cobranza 50

2.5.3. Proveedores 50

2.5.4. Productos 51

2.5.5. Detalle de Compras 52

2.5.6. Retiro de Productos 52

2.5.7. Ajuste de Productos.....53

2.5.8. Cliente.....54

2.5.9. Vehículos.....54

2.5.10. Boleta / Factura.....55

2.5.11. Pagos.....56

2.5.12 Ejemplos de algunas pantallas para control de inventario 56

2.5.12.1 Ingreso de producto 56

2.5.12.2 Ingreso de aumento de stock 57

2.5.12.3 Stock de producto actualizado en su incremento.....57

2.5.12.4 Ingreso descuento de stock.....58

2.5.12.5 Stock actualizado en su descuento 58

CONCLUSIONES.....60

BIBLIOGRAFÍA 62

Índice de Figuras.

Figura 1-1. Detalle de la orden de trabajo emanada por la rectificadora.....8

Figura 1-2. Detalle de la orden de trabajo del taller a entregar al cliente 9

Figura 1-3. Ejemplo de Factura de compras de repuesto a proveedor Auto-Stock.....10

Figura 1-4. Ejemplo de Factura de compras de repuesto a proveedor Saez y Saez.....11

Figura 1-5. Diagrama de flujo del proceso de reparación de vehículo parte 1 16

Figura 1-6. Diagrama de flujo del proceso de reparación de vehículo parte 2.....17

Figura 1-7. Diagrama de flujo del proceso de reparación de vehículo parte 3.....18

Figura 1-8. Cálculo por pago en transferencia 23

Figura 1-9. Cálculo por pago en cuotas.....24

Figura 1-10. Cálculo monto que lleva pagado el cliente 24

Figura 1-11. Modelo de datos 28

Figura 2-1. Diagrama de Menú 44

Figura 2-2. Login del sistema 47

Figura 2-3. Menú Principal del sistema 48

Figura 2-4. Menú Principal del sistema con submenú de Usuarios..... 48

Figura 2-5. Menú Principal del sistema con submenú de Inventario..... 49

Figura 2-6. Menú Principal del sistema con submenú de Gestión de Cobranza..... 50

Figura 2-7. Mantenedor de Proveedores 50

Figura 2-8. Mantenedor de Productos 51

Figura 2-9. Mantenedor Detalle de compra 52

Figura 2-10. Mantenedor Retiro de productos 52

Figura 2-11 Mantenedor Ajuste de productos 53

Figura 2-12. Módulo de Clientes 54

Figura 2-13. Módulo de Vehículos 54

Figura 2-14. Módulo Boleta / Factura 55

Figura 2-15. Módulo de Pagos 56

Figura 2-16. Ingreso de producto con código 7 56

Figura 2-17. Ingreso de aumento de stock del código 7 57

Figura 2-18. Stock de producto con código 7 actualizado 57

Figura 2-19 Ingreso descuento de stock código 7.....58

Figura 2-20. Stock con código 7 actualizado en su descuento 58

Índice de tablas.

Tabla 2-2. Tabla de Usuarios.....34

Tabla 2-3. Tabla de Productos.....35

Tabla 2-4. Tabla de Proveedores.....36

Tabla 2-5. Tabla de Compra de Productos.....37

Tabla 2-6. Tabla Detalle de Compra.....37

Tabla 2-7. Tabla de Retiro de Productos.....38

Tabla 2-8. Tabla de Ajuste de Productos.....39

Tabla 2-9. Tabla de Vehículos.....40

Tabla 2-10. Tabla de Clientes.....41

Tabla 2-11. Tabla de Deuda.....42

Tabla 2-12. Tabla de Pago.....43

INTRODUCCIÓN

En chile existe una gran cantidad de personas que posee un vehículo estos pueden ser un auto , camión , camionetas , motos , van , etc. Sin embargo tener un vehículo constituye una gran responsabilidad , pues a medida que pase el tiempo el vehículo tiende a ser ocupado con una alta frecuencia . A causa de ello las piezas que contienen los vehículos tienden a deteriorarse, es decir, sufren daños o fallas que afectan al funcionamiento del vehículo y posteriormente este queda inutilizable por un tiempo mientras no se realice una mantención y/o reparación adecuada, los conductores experimentan estos sucesos a medida que se alarga la vida útil del vehículo y muchos de ellos no saben cómo repararlos cuando se ven enfrentados a estas situaciones , pues aquí aparecen los taller mecánicos.

Pero ¿ Que es un taller mecánico?, son establecimientos industriales en el cual se trabaja tanto con las manos como con herramientas en donde se realizan reparaciones y mantención a los vehículos para que nuevamente estén en buen estado y vuelvan a estar apto para su funcionamiento , de esta manera las personas acuden a estos talleres debido a que no saben cómo reparar sus vehículos y prefieren llevarlos a estos establecimientos , en el cual será reparado por un profesional o técnicos especializados en el área de mecánica . Como resultado se entrega el vehículo en perfecto estado de acuerdo con la reparación realizada con el fin de no solo alargar la vida útil del vehículo si no también mejorar su rendimiento . Sin embargo , cuando se realice una reparación y mantención los talleres mecánicos buscan que las personas se sientan seguras y confíen en sus servicios , puesto que toda reparación y mantención busca que el cliente y sus acompañantes no sufran daños o accidentes .

Hoy en día los talleres mecánicos son de gran utilidad para las personas que poseen un vehículo, puesto que estos establecimientos tienen ciertas relevancias , como por ejemplo poseer la implementación de herramientas , repuestos y máquinas que sean aptas para realizar el servicio de reparación y mantención de vehículos , personal capacitado , espacio amplio para recibir cierta cantidad de vehículos a reparar , etc.

En los talleres se experimenta un alto flujo de repuestos , ingreso de clientes y herramientas , puesto que algunos de ellos poseen sistemas para tener una buena administración y gestión pero no muchos lo poseen dado a que tienen una cierta habitualidad de gestionar un taller en base de planillas Excel , impresión de hojas que contiene las facturas o detalles de trabajos a realizar para luego ser almacenadas en carpetas e inclusive que la administración y gestión de compra de repuestos sean escritas en un cuaderno o recurran a mirar las facturas de los repuestos comprados para saber qué es lo que tienen en stock . Con lo anterior , nace la idea de crear un sistema que apoye la administración y gestión de cómo llevar un taller mecánico , mediante la implementación de un sistema de administración y control de inventario , con el objetivo de llevar una mejor organización , administración y gestión de los repuestos.

CAPÍTULO 1 : ASPECTOS RELEVANTES DEL ANÁLISIS

1. ASPECTOS RELEVANTES DEL ANÁLISIS.

1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN.

La sociedad Mecánica Servas limitada está conformada por el señor Mauricio Vasquez Barraza y su hermano Patricio Vasquez Barraza. Ambos tienen un taller mecánico llamado lubricantes y mecánica servas ubicados en La palma calle los Mardones – Quillota , pues en él solo trabajan los que conforman esta sociedad y su objetivo es brindar los diferentes servicios de mantención y reparación de los vehículos a los clientes con el fin de entregar el vehículo en las mejores condiciones posible.

En detalle , el taller que administran cuenta con un amplio espacio para almacenar los diferentes vehículos que vayan ingresando (Ver Figura 1-1) . Además cuentan con una oficina (Ver Figura 1-2) , en la cual se almacenan los repuestos que dispone el taller . En ella también se almacenan diferentes carpetas que tienen como contenido las diferentes facturas de la compra de repuestos a diferentes proveedores , como también contienen las órdenes de detalle de trabajo realizados a los clientes por alguna reparación de un vehículo junto con los valores de los repuestos que se colocaron como también la mano de obra y el precio total del servicio realizado .

El taller cuenta con diferentes repuestos , algunos de ellos como ser las bujías, filtros , retenes , flexibles y mangueras , aceites , pernos de culata , radiador , materiales de limpieza y armado , líquidos , cables de bujías , anillos , válvulas ,radiador , entre otros.

Es preciso señalar que , este taller implementa un funcionamiento en el cual realizan un diagnóstico , luego se determina las reparaciones , cuáles son los repuestos a colocar y posteriormente digitar una orden de detalle de trabajo en Excel , con la cual se estipulan los repuestos cobrados y el valor de la mano de obra a pagar.

1.2. SITUACIÓN ACTUAL.

Pues tal como se mencionó anteriormente , el taller mecánico lubricantes y mecánica servas tiene como dueño al señor mauricio y trabaja con su hermano patricio , en el cual han llegado muchos clientes buscando alguna reparación de sus vehículos en donde el señor mauricio o su hermano les brinda una mayor atención y cobertura de como reparar sus vehículos . En efecto es posible que sus clientes llegan a este taller y buscan un diagnóstico que le puedan dar sobre los problemas que tenga el vehículo , con esto el señor mauricio y/o su hermano entrega el diagnóstico y se presentan dos soluciones. La primera de ellas es que si el sr mauricio previo a realizar un diagnóstico ve que existe un problema en un motor en donde dificulta el funcionamiento del motor , él determina sacar el motor o partes del motor y enviarlo a una rectificadora . Una vez llevado se le hacen las mediciones pertinentes con los instrumentos de precisión , de esta manera finalizando la revisión se determina que si hay partes del motor que no están funcionando bien y que deben ser rectificadas , se deben traen nuevos acoples , metales , infinidades de cosas que se ocupan en un motor . Finalmente, en la rectificadora se especifican que repuestos se van a ocupar para reparar el motor u otro problema dado , esto se deja estipulado en una orden que emite la rectificadora hacia el cliente o hacia el señor mauricio, depende de quién desea realizar esta gestión y luego se realiza la reparación en el taller junto con los repuestos (Ver Figura 1-1)

La segunda solución existe si el diagnóstico que entrega el sr mauricio y/o su hermano determinen que no esté ninguna parte mecánica del motor comprometida en el funcionamiento de él como, por ejemplo : sistema de refrigeración , sistema de encendido, suspensión , entre otros . En síntesis, es todo lo que no tenga relación directa con el mecanismo del funcionamiento del motor, solo en este caso no se debe ir a una rectificadora y de forma inmediata se ve la reparación con los repuestos a colocar en el taller.

Sin embargo, una vez que se le brinda el diagnóstico ,el sr mauricio debe acordar con el cliente la mano de obra que ese va a cobrar , ésta siempre varía dependiendo si el cliente desea que el sr mauricio sea quien vaya a comprar los repuestos , dado a que él en su estimación de cobro de mano de obra decide añadir el costo e inversión de tiempo y dinero de la gestión al cual el cliente le otorgó para la compra de los repuestos siempre y cuando en el local no tengan en stock dichos repuestos para tal diagnóstico .

En definitiva, cuando el señor mauricio termina su trabajo de reparación le cobra al cliente y le entrega el detalle de la orden de trabajo realizado junto con los montos a cobrar por repuesto colocado , como también la mano de obra y el valor total del servicio (Ver figura 1-2) siempre y cuando él esté en el taller mecánico si no su hermano el sr patricio quien es jefe del taller sea la persona quien realice el cobro al cliente y éste último puede terminar en que forma de pago desea realizar , que puede ser a través de efectivo o con trasferencia o a través de cuotas pactadas . Actualmente el señor mauricio no trabaja con el servicio de Transbank dado a que ha presentado ciertos inconvenientes y prefirió cobrar de las formas mencionadas con anterioridad a sus clientes. Una vez cobrado , mauricio envía a sus clientes por medio de WhatsApp o correo electrónico dicho detalle de trabajo con el valor total que pagó o debe pagar si el cliente deseó realizarlo por transferencia o repactar cuotas si el cliente no tiene disponibilidad de pago. Como también entrega de forma física los documentos de la rectificadora y detalle de trabajo al cliente.

En la actualidad no presentan un inventario ya que en muchas ocasiones han tenido problemas en encontrar , buscar un repuesto y siempre el sr mauricio debe ir mirando las facturas de compras a proveedores para saber que repuestos están en el local y cuanta fue la cantidad que compró (Ver Figura 1-3 , 1-4), fácilmente puede existir el extravío , olvido o robo de algún repuesto . En particular nos menciona que un buen control de repuestos que estén dentro de un inventario ayudaría , facilitaría la gestión y administración de la compra de repuestos y cuantos de ellos se tienen en existencia , tanto como entradas y salidas de repuesto por cliente que venga a realizar el trabajo a este taller.

RECTIFICACIONES
ACONCAGUA LTDA.

Fecha: 05 MAYO 2021
Tarf.:0004 Vend.:

Direccion :Avda Adrian Nº 283 Artificio La Calera
R.U.T. :76.116.350-7 Telefonos :033 2225118 - 2221999

ORDEN DE TRABAJO 245456 A

Cliente :VASQUEZ MAURICIO
Propietario :
Direccion :SITIO 53 LAS PATAGUAS

Ciud.:QUILLOTA
C.Mec:0015
R.U.T: [redacted]

Fono :62448502

0000 TOYOTA HILUX 3.0 1KD 2013

Codigo	Cant.	Detalle	V.Unit.	Total
CILINDROS	01	4 CIL ENCAMIZAR	21.850	87.400
	04	4 CIL FABRICAR CAMISAS	17.825	71.300
CIGUEÑAL	11	9 CIG RECTIFICAR	6.325	56.925
BIELA	23	4 BIE ESCUADRAR	2.990	11.960
	25	4 BIE ARMAR PISTON	2.641	10.566
	26	3 BIE ESCAREAR BUJES	4.436	13.310
BLOCK	29	5 BLO ALINEAR BANCADA	1.725	8.625
	30	5 BLO PROBAR BANCADAS	1.955	9.775
	34	1 BLO LAVADO	9.085	9.085
CULATA	40	CUL RC.ASIENTOS		
OTROS	66			

REPUESTOS	MEDIDA	VALOR
Piston	STD USADOS.	
Anillos	STD ✓	
Camisas		
Met.Biela	0.25 ✓	
Met.Bancada	0.25 ✓	
Met.Axial	0.10 ✓	
Met.Leva		
Valv.Adms.		
Valv.Escp.		
Guia Valv.		
Buje Biela	4 ✓	
Retenes Val.		
Jgo.Empq.Mt		
BIELA MOTOR	2 ✓	
Otros		
Otros		
Otros		

OBSERVACION: $

NETO $ 278.946 :: TOTAL REPUESTOS $
I.V.A. $ 53.000 ::
TOTAL $ 331.946 :: TOTAL GENERAL $ 331.946

RECEPCIONADO POR

:193,500
:BLOCK 1 C/TAPAS
:CIGUEÑAL 1
:LEVA
:BIELAS 4 C/PISTON
:CULATA
:OTROS
:OTROS

RECTIFICACIONES ACONCAGUA LTDA.
23 AGO 2021
PAGADO

NOTA 1: Despues de 60 dias ,no se responde por daños o perdidas de piezas.
NOTA 2: Este Presupuesto queda sujeto a modificaciones , mientras se esta ejecutando los trabajos de rectificacion. ★ :

Figura 1-1. Detalle de la orden de trabajo emanada por la rectificadora.

Fuente : Sr. Mauricio Vasquez.

Lubricantes & Mecánica
Servas

ORDEN DE TRABAJO

Fecha de Recepcion:
Vehiculo: SUZUKI MARUTI
Nombre:
Patente:
Kilometraje:

Fecha de Entrega:
Telefono:

DETALLE DE TRABAJOS A REALIZAR

Aceite Motor	
Aceite Transmision	
Aceite de Direccion	
Liquido de Frenos	
Liquido Refrigerante	
Liquido Limpiaparabrisas	
Filtro de Aceite	
Filtro de Combustible	
Filtro de Aire	
Filtro de Polen	
Retenes	
Juego Empaquetadura motor	
Flexible y manguera alternador	
Mangueras-Abrasaderas	
Pernos de culata	
Bujias	$ 9.000
Radiador	
TOTAL	$ 9.000

Bomba de Agua -Termostato	
Kit de Distribucion	
guias de valvulas	
bujes de bielas	
piñon cigüeñal	
piñon compensador	
Tapa distribuidor	$ 42.300
Cables de bujias	$ 14.800
chavetas cigüeñal	
pernos reguladores de valvulas	
perno cigüeñal	
Cilindros de Frenos	
Pistones	
Anillos	
Metales	
Valvulas	
eje de levas	
TOTAL	$ 57.100

OTROS	
Materiales limpieza y armado	
Rectificado Volante	
Rectificado Discos-Tambores	
Rectificadora	
Arregio Inyectores	
Total	$ -

Mano de Obra y gestion repuest.	$ 55.000
Total	$ 55.000

TOTAL A PAGAR	$ 121.100

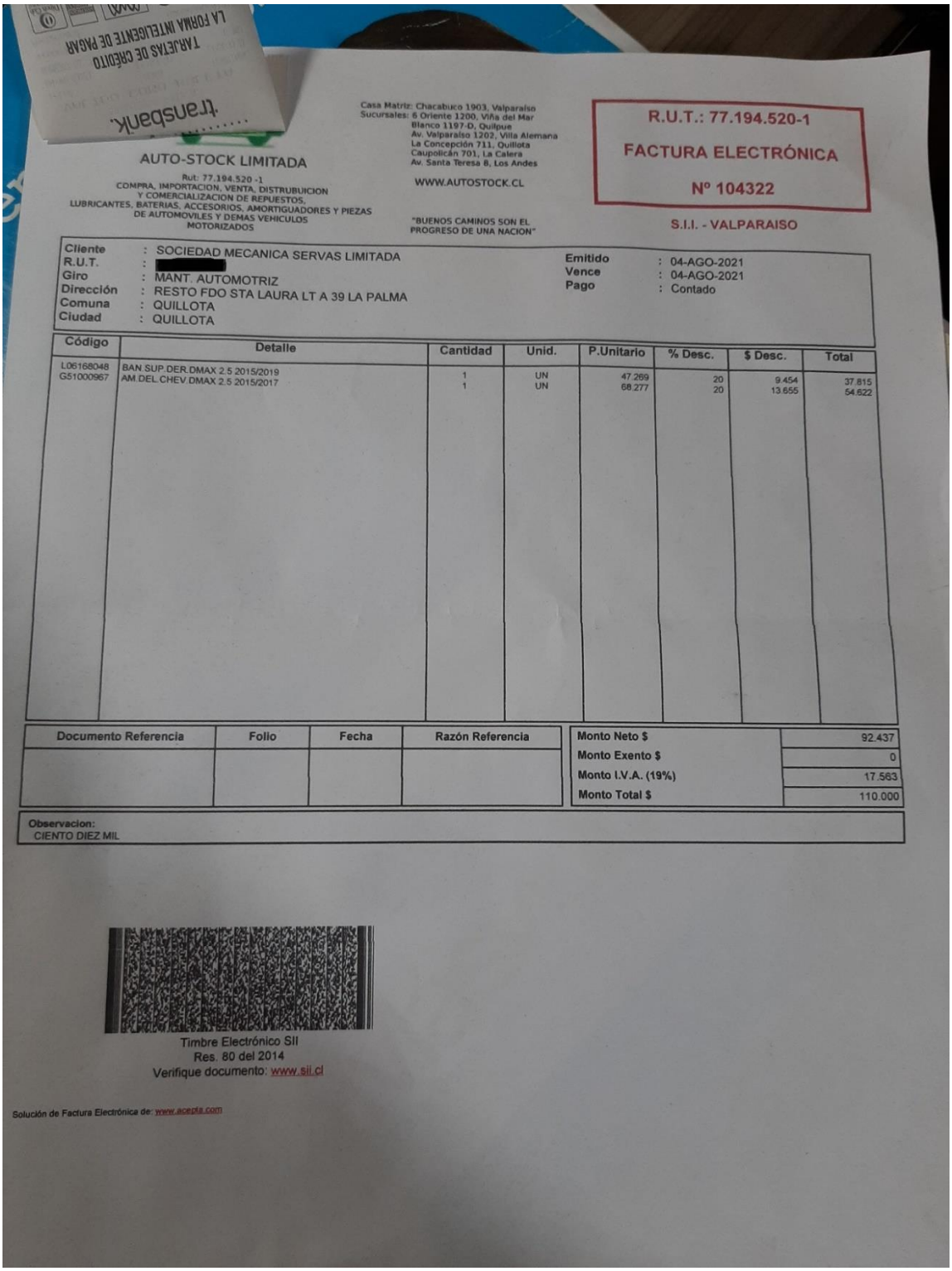
ABONOS	
ABONOS	
ABONOS	
ABONOS	
TOTAL ABONOS	$ -

SALDO A PAGAR	$ 121.100

... Santa Laura - La Palma - Quillota Fono: 9-62448502

Figura 1-2. Detalle de la orden de trabajo del taller a entregar al cliente.

Fuente : Sr. Mauricio Vasquez.



AUTO-STOCK LIMITADA

Rut: 77.194.520 -1
COMPRA, IMPORTACION, VENTA, DISTRUBUICION Y COMERCIALIZACION DE REPUESTOS, LUBRICANTES, BATERIAS, ACCESORIOS, AMORTIGUADORES Y PIEZAS DE AUTOMOVILES Y DEMAS VEHICULOS MOTORIZADOS

Casa Matriz: Chacabuco 1903, Valparaíso
Sucursales: 6 Oriente 1200, Viña del Mar
Blanco 1197-D, Quilpue
Av. Valparaíso 1202, Villa Alemana
La Concepción 711, Quillota
Caupolicán 701, La Calera
Av. Santa Teresa 8, Los Andes

WWW.AUTOSTOCK.CL

"BUENOS CAMINOS SON EL PROGRESO DE UNA NACION"

R.U.T.: 77.194.520-1
FACTURA ELECTRÓNICA
Nº 104322
S.I.I. - VALPARAISO

Cliente : SOCIEDAD MECANICA SERVAS LIMITADA
R.U.T. : [redacted]
Giro : MANT. AUTOMOTRIZ
Dirección : RESTO FDO STA LAURA LT A 39 LA PALMA
Comuna : QUILLOTA
Ciudad : QUILLOTA

Emitido : 04-AGO-2021
Vence : 04-AGO-2021
Pago : Contado

Código	Detalle	Cantidad	Unid.	P.Unitario	% Desc.	$ Desc.	Total
L06168048	BAN.SUP.DER.DMAX 2.5 2015/2019	1	UN	47.269	20	9.454	37.815
G51000967	AM.DEL.CHEV.DMAX 2.5 2015/2017	1	UN	68.277	20	13.655	54.622

Documento Referencia	Folio	Fecha	Razón Referencia

Monto Neto $	92.437
Monto Exento $	0
Monto I.V.A. (19%)	17.563
Monto Total $	110.000

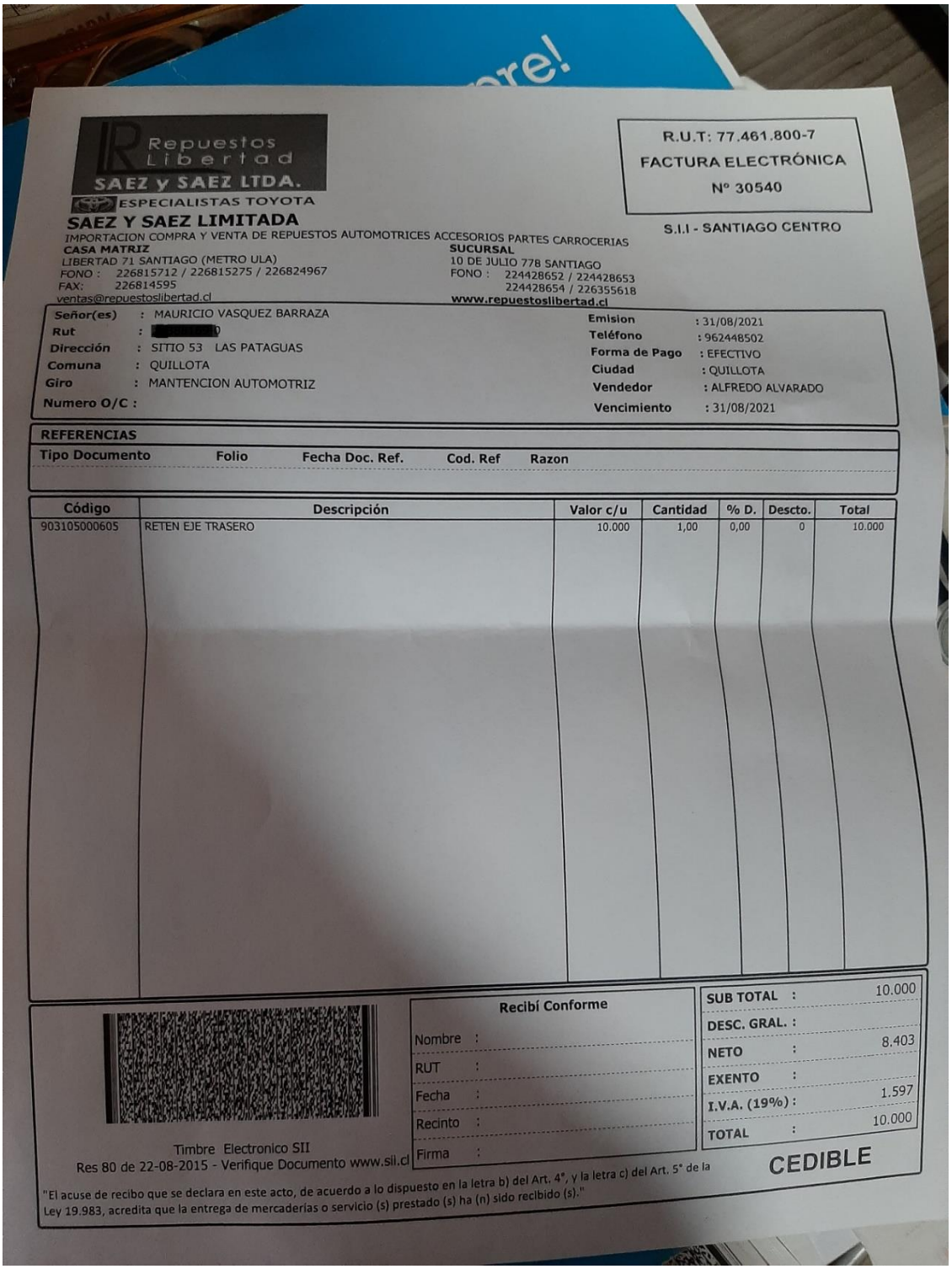
Observacion:
CIENTO DIEZ MIL

Timbre Electrónico SII
Res. 80 del 2014
Verifique documento: www.sii.cl

Solución de Factura Electrónica de: www.acepta.com

Figura 1-3. Ejemplo de Factura de compras de repuestos a proveedor Auto-Stock.

Fuente : Sr. Mauricio Vasquez.



Repuestos Libertad
SAEZ y SAEZ LTDA.
ESPECIALISTAS TOYOTA

R.U.T: 77.461.800-7
FACTURA ELECTRÓNICA
Nº 30540

S.I.I - SANTIAGO CENTRO

SAEZ Y SAEZ LIMITADA
IMPORTACION COMPRA Y VENTA DE REPUESTOS AUTOMOTRICES ACCESORIOS PARTES CARROCERIAS

CASA MATRIZ
LIBERTAD 71 SANTIAGO (METRO ULA)
FONO : 226815712 / 226815275 / 226824967
FAX: 226814595
ventas@repuestoslibertad.cl

SUCURSAL
10 DE JULIO 778 SANTIAGO
FONO : 224428652 / 224428653
224428654 / 226355618
www.repuestoslibertad.cl

Señor(es)	: MAURICIO VASQUEZ BARRAZA	Emision	: 31/08/2021
Rut	: [redacted]	Teléfono	: 962448502
Dirección	: SITIO 53 LAS PATAGUAS	Forma de Pago	: EFECTIVO
Comuna	: QUILLOTA	Ciudad	: QUILLOTA
Giro	: MANTENCION AUTOMOTRIZ	Vendedor	: ALFREDO ALVARADO
Numero O/C :		Vencimiento	: 31/08/2021

REFERENCIAS

Tipo Documento	Folio	Fecha Doc. Ref.	Cod. Ref	Razon

Código	Descripción	Valor c/u	Cantidad	% D.	Descto.	Total
903105000605	RETEN EJE TRASERO	10.000	1,00	0,00	0	10.000

Timbre Electronico SII
Res 80 de 22-08-2015 - Verifique Documento www.sii.cl

Recibí Conforme
Nombre :
RUT :
Fecha :
Recinto :
Firma :

SUB TOTAL :	10.000
DESC. GRAL. :	
NETO :	8.403
EXENTO :	
I.V.A. (19%) :	1.597
TOTAL :	10.000

CEDIBLE

"El acuse 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 19.983, acredita que la entrega de mercaderías o servicio (s) prestado (s) ha (n) sido recibido (s)."

Figura 1-4. Ejemplo de Factura de compra de repuesto a proveedor Saez y Saez .

Fuente : Sr. Mauricio Vasquez.

1.3. PROBLEMAS GENERALES DETECTADOS.

Generalmente el señor mauricio va a comprar repuestos a diferentes proveedores , los cuales al momento de llegar al taller son dejados en su oficina . Lamentablemente existen ocasiones en donde él ha tenido que ver todos los repuestos que estén en esa oficina para contarlos manualmente o tomar el detalle de las facturas que tiene en carpetas archivadas y saber qué fue lo que compro y en que cantidades . En sí , no existe un inventario de repuestos que pueda ayudar a conocer el stock de productos almacenados.

Por otro lado pueden haber ocasiones donde el señor mauricio ha tenido que efectuar el medio de pago tanto por transferencia como pactar cuotas con el cliente , de este modo para ambas debe existir una implementación segura y futura que debiera tener los datos de los clientes que estén morosos , quizás no anotados en un cuaderno o archivados en carpeta o que estén en una planilla Excel dentro del computador , sino más bien que estén dentro de un sistema que le permita administrar y gestionar el pago de los clientes morosos de esta forma queda almacenado en un sistema y que no sufra un extravío u olvido de los datos del cliente junto con su valor a pagar y que esto signifique una resta de ganancia importante para la sociedad mecánica servas limitada . Si bien , la implementación de un control de deudores permitirá tener un control futuro de posibles deudores , puesto que la sociedad debe estar preparada para enfrentar estos tipos de problemas y solucionarlos de una mejor manera.

En resumen los problemas detectados en situaciones actuales son :

1. No existe un inventario de repuestos
2. Toda la información de facturas de compras se almacena en carpetas manuales , cómo también las ordenes de detalle de trabajo digitadas en Excel que almacena los datos del vehículo y cliente , lo cual dificulta el manejo y búsqueda de la información
3. Para una implementación futura no tiene un control de deudores , teniendo en cuenta que actualmente se realizan los pagos por transferencia o en efectivo significaría que manualmente debe registrar , anotar o buscar un pago realizado por los clientes.

1.4. DESCRIPCIÓN DEL SISTEMA PROPUESTO.

Para afrontar las situaciones detectadas en el taller mecánico se presenta una solución que es un sistema que permita resolver los problemas detectados de una manera óptima, segura con una buena administración , gestión y organización del taller . Este se denomina sistema de administración y control de inventario de repuestos para taller mecánico .

El sistema tendrá dos funcionalidades , la primera de ellas consiste en tener un control de inventario el cual tendrá secciones de productos , proveedores , compras de los productos para manejar las entradas , detalle de las compra de productos y retiro de productos para manejar las salidas , es decir , manejar el stock de cada producto de acuerdo con sus entradas y salidas. Esto permite tener una buena administración y gestión de los repuestos que hay en el taller ya que deben mantener siempre un stock para realizar los servicios que ofrecen. Mientras que en la sección de proveedores se maneje todo dato relacionado al proveedor , de esta manera se busca saber que proveedor suministra tal repuesto que se ha comprado.

La segunda de ellas consiste en tener un control de deudores , puesto que esto ayudaría a mantener al día a los futuros clientes deudores ya que debe existir una preocupación , orden en saber quiénes tienen una deuda con el pago , ya que el taller mecánico podría eventualmente tener una descapitalización al no llevar un control y asimismo evitar problemas futuros en el cual la sociedad pueda verse afectada.

1.4.1. Objetivo General.

El objetivo del proyecto es desarrollar un sistema de administración y control de inventario de repuestos para el taller mecánico lubricantes y mecánica servas , de esta manera se brindará un apoyo a la administración , gestión y orden de ir controlando los repuestos como también controlar a los futuros deudores que hayan recibido el servicio realizado en el taller.

1.4.2. Objetivos Específicos.

- Creación , administración y gestión de las cuentas de usuario.
- Creación y administración de los datos de los clientes
- Creación y administración de los datos de vehículos de los clientes.
- Creación y administración de los datos de la deuda que mantiene un cliente con situación financiera de poseer una deuda.
- Creación y administración de los datos de pago que realizó el cliente deudor.
- Creación y administración de los datos de los productos.
- Creación y administración de los datos de los proveedores.
- Creación y administración de los datos de compra de los repuestos.
- Creación y administración de los datos de detalle de compra.
- Creación , administración y gestión de los datos de los retiros de los productos.
- Creación y administración de los ajustes realizados a los productos.
- Creación y administración de los reportes a generar.
- Administración y control de los pagos realizados por los clientes deudores.

1.4.3. Beneficios del sistema.

Por un lado , el tener un control de inventario para este taller es una herramienta útil y una de las mejores formas de poder administrar todo lo que tenga el taller con el fin de evitar pérdidas, robos, extravió de cualquier suministro que ingrese al local . De esta forma los beneficios que aporta son :

A. Facilita la organización y administración de los productos comprados
B. Facilita el manejo de la información exacta que será útil para el aprovisionamiento de productos
C. Vigilancia de los productos almacenados , es decir los usuarios podrán saber la situación particular de cada producto.

D. Ayuda a tener un control del stock de los productos almacenados
E. Ayuda a saber con qué proveedor se efectúa la compra de repuestos.
F. Ahorro de tiempo para la búsqueda de información.

Por otro lado , el tener un control de deudores como beneficio principal permite tener una buena gestión y administración de cobranza , pero existen otros beneficios como por ejemplo :

A. Facilita tener un control y gestión de los pagos realizados.
B. Disminuye la perdida de información que conllevan a la confusión y desconocimiento sobre la deuda que acredita el cliente
C. Ahorro de tiempo y recordatorios , ya que le permite conocer la deuda y los clientes morosos
D. Permite segmentar clientes , es decir, se podrá ver con facilidad quien tiene deuda
E. Facilita la toma de decisiones y mejoras para evitar la falta de liquidez , descapitalización y que las entradas en efectivo no se vean perjudicadas
F. Ayuda a monitorear cada situación de deuda de los clientes
G. Mayor eficacia para controlar a los deudores y gestionar una cobranza controlada
H. Mejora la rentabilidad y estado financiero ya que deben recuperar liquidez

1.4.4. DIAGRAMA DE FLUJO ADMINISTRATIVO.

1.4.4.1. Diagrama de flujo del proceso de reparación de vehículo.

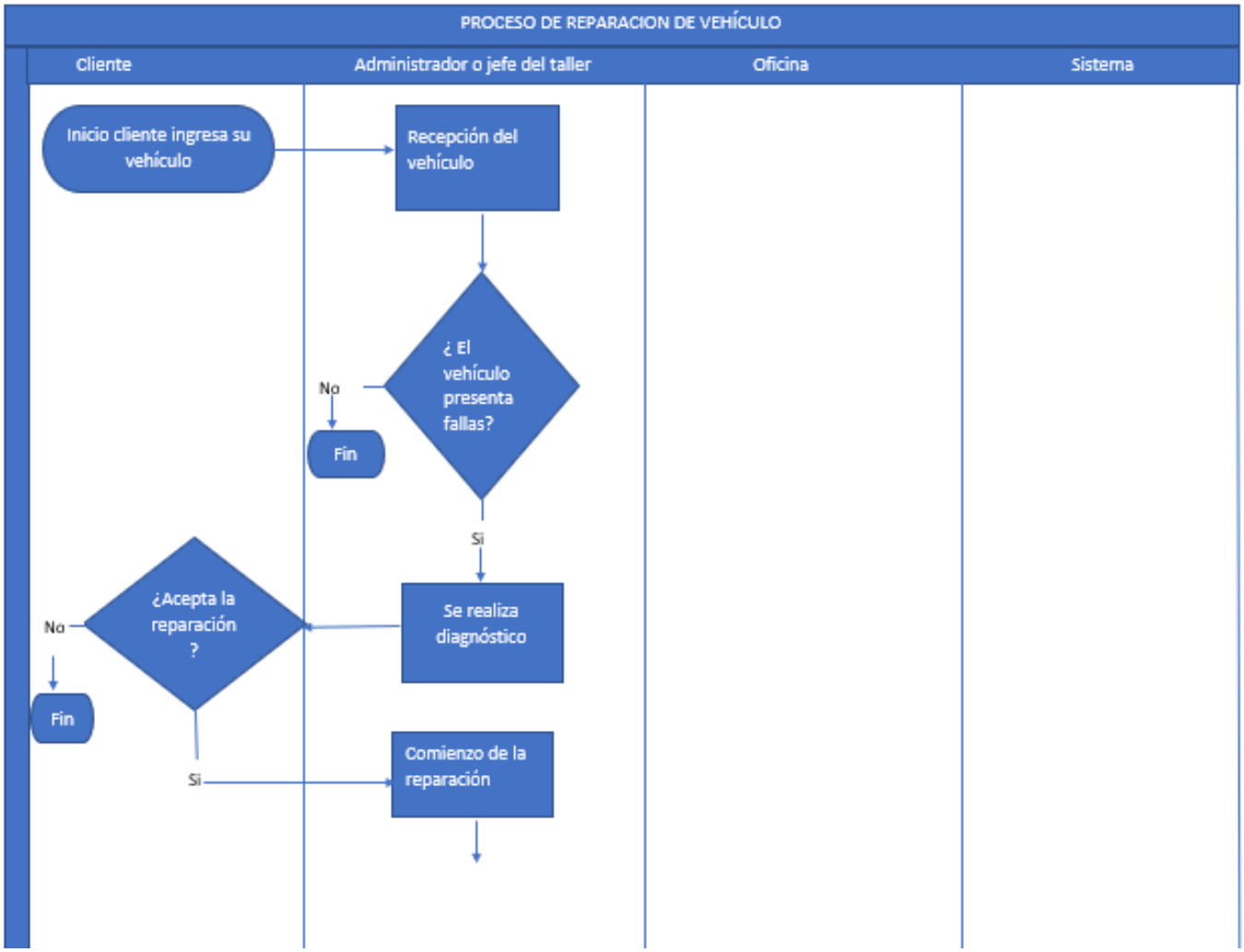


Figura 1-5. Diagrama de flujo del proceso de reparación de vehículo parte 1.

Fuente : Elaboración propia.

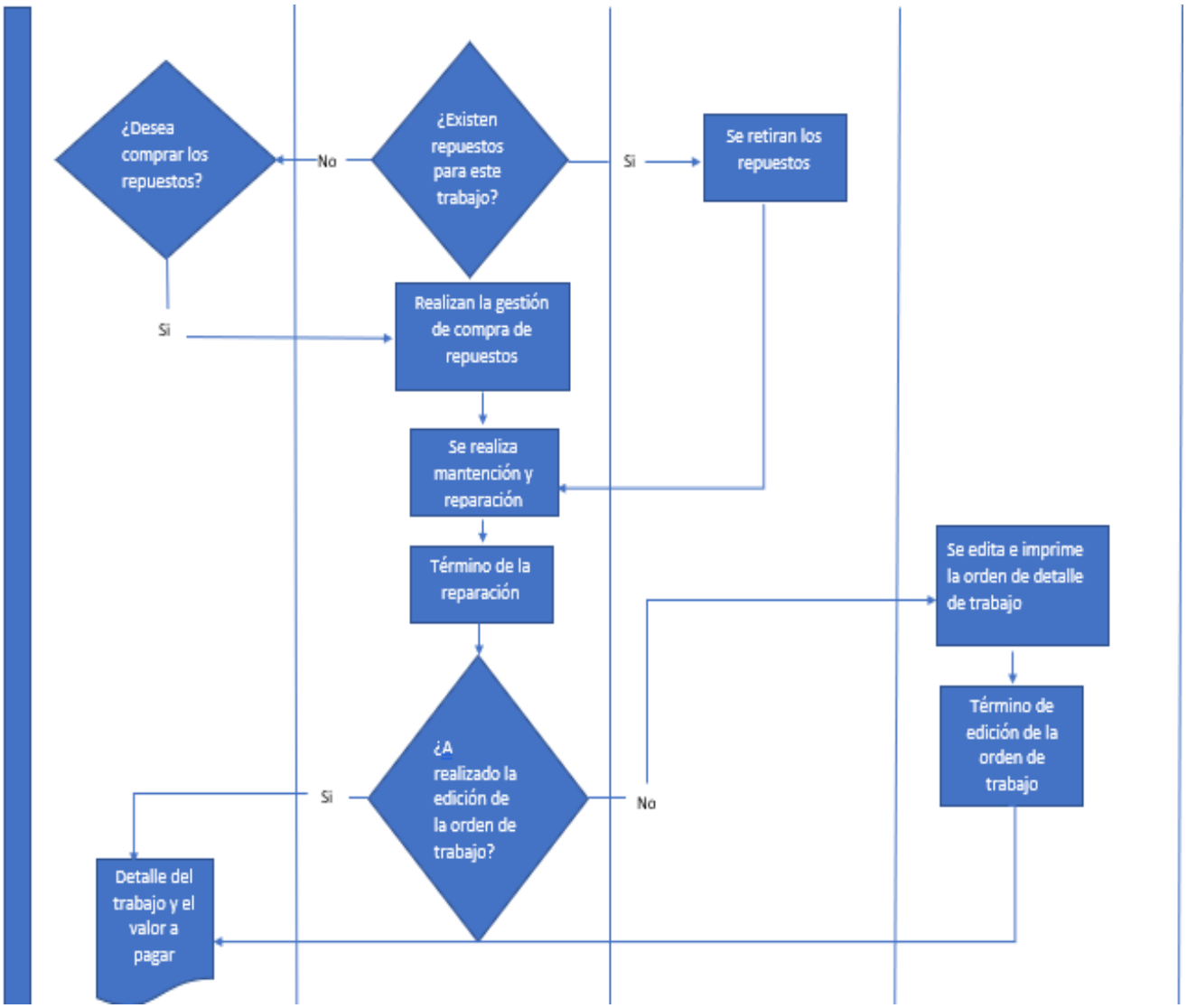


Figura 1-6. Diagrama de flujo del proceso de reparación de vehículo parte 2.

Fuente : Elaboración propia.

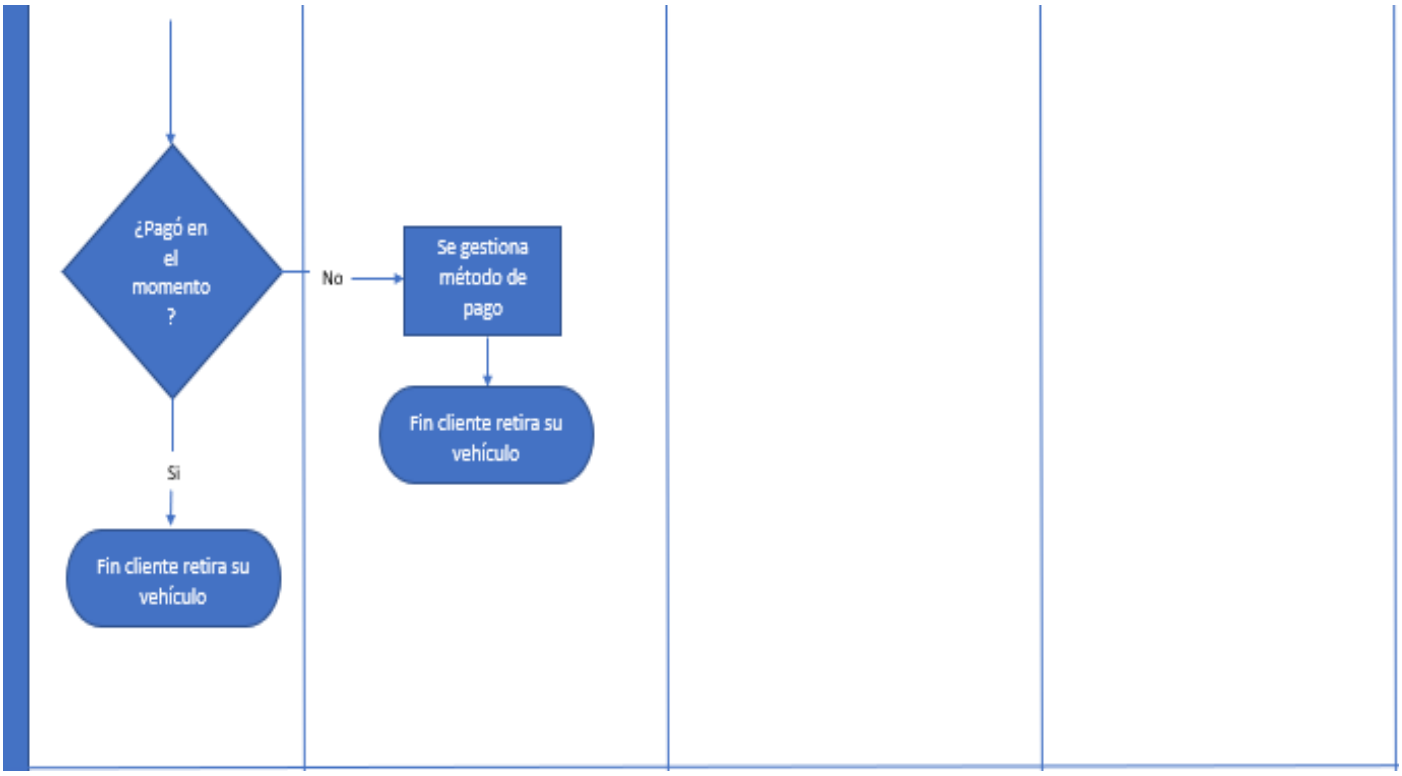


Figura 1-7. Diagrama de flujo del proceso de reparación de vehículo parte 3.

Fuente : Elaboración propia.

1.4.5. Funcionalidades del sistema.

- El sistema debe permitir la creación y administración del mantenedor de cuentas de usuarios , puesto que almacenará todas las cuentas de los usuarios que deban ingresar al sistema . Para ello tendrá las funcionalidades de agregar , modificar , eliminar y consultar un usuario.

  El sistema al contar con este mantenedor , se pedirá el ingreso de nombre de usuario , correo, contraseña y nombre del rol.

- El sistema contará con un inicio de sesión de las diferentes cuentas de usuario . Los usuarios tendrán acceso a las secciones del inventario como también al control de deudores. Se pide el ingreso de nombre de usuario , contraseña .

- El sistema debe permitir la creación y administración de un inventario . Este proceso consiste en tener mantenedores de productos y proveedores . Además contiene las compras de productos , detalle de compra de productos y retiro de productos .
  Para estas secciones todos los usuarios registrados tendrán acceso .
  Las funcionalidades son las siguientes :

  A. Para el mantenedor de productos :
     - Ingresar al sistema todo tipo de repuestos comprados y los que ya están en las oficinas del taller junto con las características del producto , esta sección estará en detalle en las entradas del sistema para esclarecer los datos a pedir.
     - Agregar , modificar y eliminar productos.
     - Implementar la funcionalidad de filtro ya que hace énfasis a realizar una consulta más expedita y encontrar de forma oportuna el repuesto.

  B. Para el mantenedor de proveedores :
     - Ingresar al sistema los datos de todo proveedor al cual se le han comprado los repuestos .
     - Agregar , modificar y eliminar proveedores.
     - Consultar por algún proveedor al cual se le ha comprado algún repuesto.

C. Para compra de productos :

- Ingresar al sistema los datos de los números de facturas junto con la fecha de compra . En efecto , no será necesario ingresar el número de factura real puesto que el usuario determinará un código de factura
- Agregar , modificar , eliminar una compra.
- Consultar por algún número de factura o fecha de compra.

D. Para detalle de compra :

- Ingresar al sistema todo los datos relacionados a los detalles de una compra realizada asociado al código de una factura.
- Agregar , modificar y eliminar detalle de compra.
- Consultar por algún detalle .

E. Para retiro de productos :

- Ingresar al sistema todos los datos de los retiros de un producto .
- Agregar , modificar y eliminar retiro de productos.
- Consultar por algún retiro de producto.

- El sistema deberá permitir la creación y administración del mantenedor de ajustes de un producto con respecto a stock ya sea por robo , extravíos , etc. . Para ello se deberá agregar , modificar y eliminar un ajuste, como también realizar consultas.

- El sistema debe generar alertas o ventanas emergente cuando haya un déficit de stock .

- El sistema debe permitir la creación y administración de un control de deudores , este proceso consiste en gestionar y administrar la cobranza , ya que en la sección de clientes se almacenarán todos los clientes y de acuerdo con el estado financiero del cliente acreditando ser deudor , de esta manera se controlará a todo cliente deudor.

Para este control el cual contará con los siguientes módulos y solo tendrán acceso los usuarios de jefe de taller y administrador :

A . Para el módulo de Clientes :

- Ingresar todos los datos de los clientes.
- Agregar y modificar los datos de los clientes.
- Consultar por algún cliente.

B . Para el módulo de Vehículos :

- Ingresar todos los datos de los vehículos de los clientes.
- Agregar , modificar un vehículo.
- Consultar por algún vehículo.

C . Para el módulo de Deuda :

- Ingresar todos los datos de una deuda que mantiene un cliente de acuerdo con su situación financiera de deudor.
- Agregar y modificar una deuda .
- Consultar por alguna deuda.
- Cotizar una deuda generada e implementación de opción calcular y limpiar. Esto no quedará ingresado en el sistema , pues es solo una cotización .
- Generar un pdf con la deuda generada por cliente para ser enviada.

D . Para la módulo de Pago :

- Ingresar los datos de un pago realizado por el cliente deudor.
- Agregar , actualizar un pago.
- Consultar o buscar un pago realizado.
- Se deberá implementar la funcionalidad de calcular el monto que lleva pagado el cliente , para esto solo será efectivo ingresar el Rut del cliente y el valor del trabajo realizado (deuda total) , posteriormente el sistema establecerá por

pantalla el monto de la deuda que lleva pagado . Estos datos no ingresarán al sistema puesto que cumple una función que no debe ser registrada .

- El sistema deberá tener una sección de historial de deuda y/o pago , para ello solo se ingresará el Rut del cliente , patente del vehículo , entre otros. Este proceso no quedará registrado en el sistema , puesto que también tendrá la opción de limpiar para consultar por otro historial de un cliente en específico.

- El sistema deberá generar informes o reportes en las diferentes funcionalidades del sistema del cual se detallan en la sección salidas del sistema.

1.5. DESCRIPCIÓN DE CÁLCULOS Y FÓRMULAS EMPLEADAS EN EL SISTEMA.

- Teniendo en cuenta que en el control de deudores , específicamente en el módulo de deuda se tendrá una función de cotización de pago que tendrá como opciones de cuotas y transferencia , si bien en un futuro algún cliente que desee repactar el valor total del servicio entregado es posible que desee conocer los valores que quedarían por pagar o el futuro monto a pagar en caso de una transferencia , es por esto que el sistema hará lo siguientes cálculos :

1) Si el cliente desea realizar una parte del pago por transferencia el sistema deberá realizar el siguiente calculo :

Le queda por pagar = Valor total del trabajo realizado − Monto que desee pagar el cliente

Figura 1-8. Cálculo por pago en transferencia.

Fuente : Elaboración propia.

En donde :

- Le queda por pagar : Es el resultado del cálculo generado.
- Valor total del trabajo realizado : Se ingresa el valor total del servicio entregado.
- Monto que desee pagar el cliente : Se ingresa el valor que el cliente quiera pagar .

2) Si el cliente desea pactar en cuotas el pago el sistema deberá realizar el siguiente calculo:

$$\text{Monto por cuota} = \frac{\text{Valor total del trabajo realizado}}{\text{Cantidad de cuotas}}$$

Figura 1-9. Cálculo por pago en cuotas.

Fuente : Elaboración propia.

En donde :

- Monto por cuota : Es el resultado del cálculo generado a pagar por cada cuota generada
- Valor total del trabajo realizado : Se ingresa el valor total del servicio entregado.
- Cantidad de cuotas : Se ingresa la cantidad de cuotas que el cliente desee.

- En el módulo de pago , el sistema deberá realizar el siguiente cálculo :

$$\text{Lleva pagado} = \sum_{monto\ pagado\ =1}^{n} \text{monto pagado} + \text{monto pagado} + \text{monto pagado} + \text{monto pagado} + \cdots + \text{n}$$

Figura 1-10. Cálculo monto que lleva pagado el cliente.

Fuente : Elaboración propia.

En donde :

- Lleva pagado : Es el resultado de las sumatoria de todos los montos pagados por el cliente.
- Monto pagado : Es el monto que ha pagado el cliente .
- n = Cantidad de veces de un monto pagado.

1.6. INFORMACIÓN QUE MANEJARÁ EL SISTEMA.

1.6.1. Entradas.

- Datos de usuario.
- Datos de productos.
- Datos de proveedores.
- Datos de clientes.
- Datos de deuda que tiene un cliente deudor.
- Datos de vehículos
- Datos de pago.
- Datos de la compra del producto.
- Datos de detalle de la compra del producto.
- Datos de retiro de productos.
- Datos de ajuste del producto.

1.6.2. Salidas.

- Reporte en pdf del monto a pagar junto con otros datos del cliente.
- Informes de stock
- Informe de proveedores .
- Si existe una cantidad inferior a 10 repuestos de diferente categoría se alertará al dueño mediante una alerta emergente sobre el stock que se encuentra actualmente dicho producto
- Informe de clientes que repactaron alguna deuda , con el valor a pagar o clientes morosos que no hayan realizado transferencia .
- Resultado de la consulta de la cotización de pago de cuotas o transferencia.
- Resultado de la consulta de lo que lleva pagado el cliente

1.7. ENTIDADES DE DATOS.

- **Usuarios (PK = nombre_usuario -> Caracteres alfanuméricos)** : Registrará a los diferentes usuarios que tendrá el sistema .
- **Clientes (PK = rut_cliente -> Caracteres alfanuméricos)** : Registrará a todos los clientes que hayan recibido el servicio ofrecido por el taller .

  Dado a que el sistema busca controlar a deudores , en este caso la entidad tendrá una situación financiera de cada cliente que permite ubicar al cliente en un estado ya sea de deudor , no posee deuda (pago completo) . Así mismo guardará y almacenará toda la cartera de clientes pero teniendo énfasis a controlar la situación financiera de un cliente deudor.
- **Deudas (PK = codigo_deuda -> Número correlativo)** : Registrará la deuda que mantiene todo cliente que se encuentre en situación financiera de deudor .
- **Pagos (PK = codigo_pago ->Número correlativo) :** Registrará los pagos realizados por los clientes en donde su situación financiera sea deudor .
- **Productos (PK = codigo_producto -> Número correlativo)** : Registrará los productos que fueron comprados a diferentes proveedores .
- **Proveedores (PK = rut_proveedor -> Caracteres)** : Registrará a los diferentes proveedores a los cuales se les compra los productos , además contendrá el nombre , ubicación , teléfono del proveedor.
- **Compra_Productos (PK = codigo_factura -> Número correlativo)** : Registrará el número único de una factura este será un código , el cual se identifica con un número correlativo que proporcionará el usuario , pues no se debe ingresar el número real de la factura.

  Con este número se podrá ver los datos de una compra realizada a un proveedor , en el cual su detalle estará especificado en la entidad Detalle_Compra.
- **Detalle_Compras (codigo_factura + codigo_producto)** : Registrará el detalle de la compra realizada asociado a un código de factura. En síntesis , es un desglose de lo comprado.

- **Retiro_Productos (PK = codigo_retiro_producto -> Número correlativo)** : Registrará las salidas del producto , en este caso el retiro se origina por el servicio entregado a cada cliente ya que se descuenta cada producto retirado según la reparación que se deba realizar .
- **Ajuste_Productos (PK = codigo_ajuste -> Número correlativo)** : Registrará los ajustes realizados a un producto con respecto a su stock producido por robo , extravíos , etc.
- **Vehiculos (PK = patente_vehiculo -> Caracteres alfanuméricos) :** Registrará los vehículos de los clientes.

1.7.1. Modelo de datos.

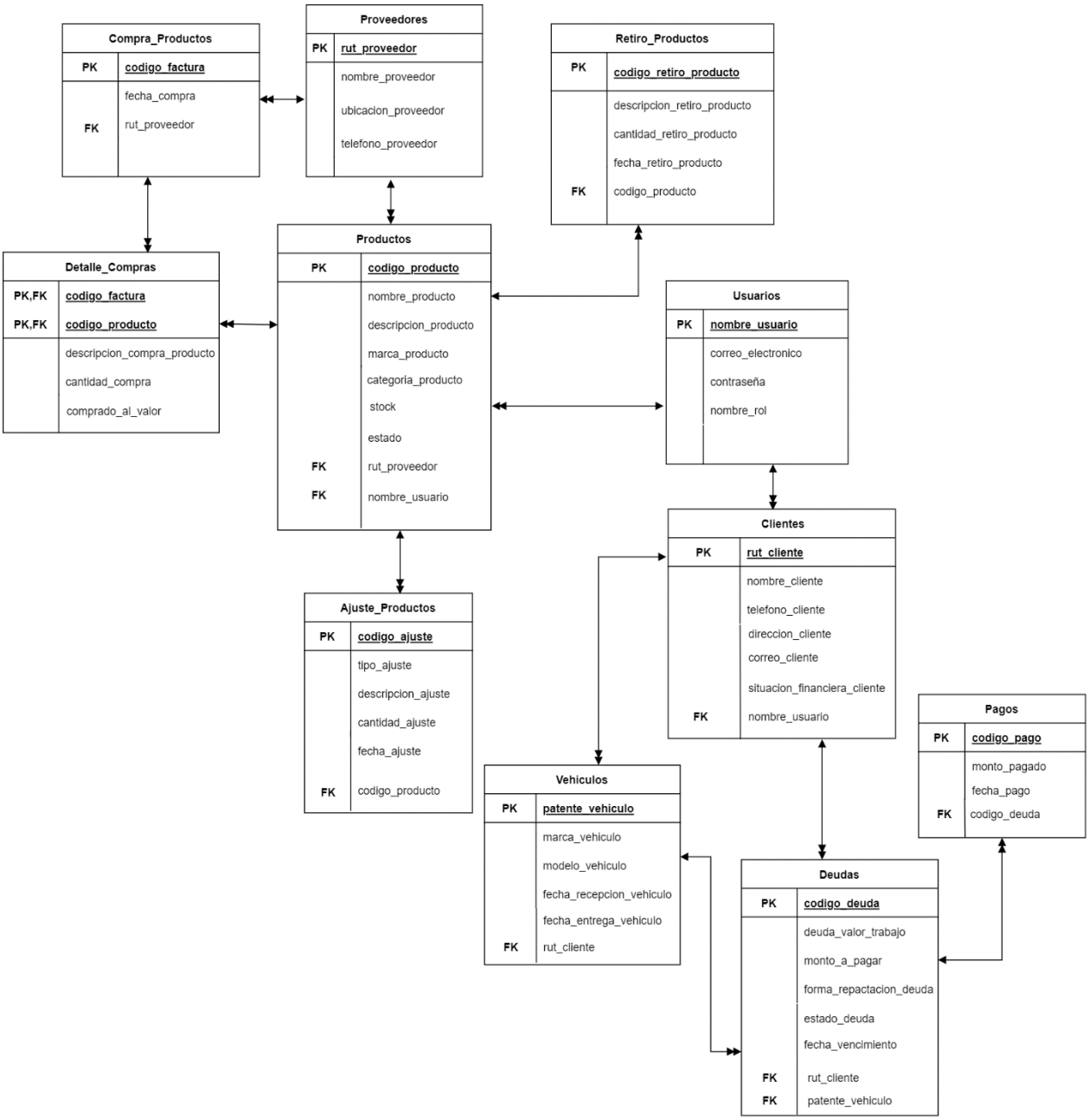


Figura 1-11. Modelo de datos.

Fuente : Elaboración propia

1.8. CONDICIONES DE DISEÑO.

El taller mecánico solo cuenta con un notebook fijo , pues el sistema estará orientado a un solo notebook .

En efecto , en el sistema a desarrollar se utilizará un lenguaje de programación llamado Visual Basic a través del programa Visual Studio , junto con ello se utilizará una base de datos MYSQL y el software XAMPP para habilitar los servicios de MYSQL.

CAPÍTULO 2 : ASPECTOS RELEVANTES DEL DISEÑO.

2. ASPECTOS RELEVANTES DEL DISEÑO.

2.1. DESCRIPCIÓN DE RECURSOS COMPUTACIONALES.

En este capítulo se detallará el medio ambiente computacional en que se realizará el sistema propuesto . Así mismo , se describirá su configuración como ser el hardware usado , software .

2.1.1. Configuración del sistema

La configuración del sistema se adhiere a los recursos computaciones con los que se cuente . En este caso , el taller tiene un notebook el cual le pertenece al sr mauricio y es llevado al taller para trabajar en él en su oficina.

Con respecto a lo anterior , se presentan las características de hardware para el desarrollo del sistema e implementación del sistema.

2.1.2. Hardware usado.

- Para el desarrollo del sistema :
  - Notebook : Lenovo Ideapad 520-15IKB
  - Tipo de procesador : EL procesador del equipo es un Intel Core i5-8250U 8º generación Dual-Core de 1.60GHz hasta 3,4 GHz
  - Memoria RAM : La memoria del equipo es de una tarjeta de 8GB del tipo DDR4
  - Almacenamiento : El disco duro del equipo tiene un almacenamiento de 2 TB

- Para la implementación futura del sistema propuesto se tiene un notebook stand alone , ya que es el único equipo con él que cuentan para trabajar. Éste tiene un procesador Intel Celeron B800 1.50 GHZ , además cuenta con una memoria RAM de 4gb con un tipo de sistema operativo de 32 bits con almacenamiento 500 GB.

2.1.3. Software utilizado.

El notebook que se usará para desarrollar el sistema tiene un sistema operativo Windows 10 , mientras que el notebook que se usará para implementar el sistema tiene un sistema operativo Windows 7 Ultimate.

A continuación , se presenta la herramienta de desarrollo de software :

- Entorno de desarrollo : El entorno de desarrollo es Visual Studio puede ser la versión de 2017 o 2019 . En efecto , este es el mejor IDE completo para desarrolladores de .Net en Windows , el cual contiene Windows Forms (.NET Framework) . Este cuenta con un diseñador visual de arrastrar y soltar para crear fácilmente aplicaciones de Windows Forms .
- Lenguaje : El lenguaje de programación para desarrollar este sistema es Visual Basic . Sin embargo , se necesita conectar con el gestor de base de datos y para ello se debe instalar un conector especial de MYSQL .Este se denomina mysql-connector-net.
- Motor de base de datos : Se ocupará MYSQL , en el cual para habilitar sus servicios se utilizará XAMPP. Para habilitarlo se activa la conexión a la base de datos por el servidor instalado en XAMPP de MYSQL.
- DBMS : El sistema de gestión de base de datos que se utilizará es phpMyAdmin . Este viene incluido en el software XAMPP.

2.2. DESCRIPCIÓN DE ARCHIVOS.

2.2.1. Descripción de tipos de datos soportados por el motor de base de datos

El motor de base de datos MYSQL soporta diferentes tipos de datos . De esta manera , en la documentación oficial de MYSQL se estipula que se tiene diferentes tipos de datos como ser numéricos (TINYINT , SMALLINT , INT , DOUBLE , FLOAT , REAL ,DECIMAL,MEDIUMINT,BIGINT) , fecha y hora (DATE , TIME , DATETIME , TIMESTAMP , YEAR), de cadena(CHAR, VARCHAR , BINARY, VARBINARY , BLOB , TEXT , ENUM ,SET), especiales enfocados a geometría (GEOMETRY, POINT , POLYGON , ENTRE OTROS)

2.2.2. Descripción detallada de tablas.

2.2.2.1. Usuarios.

- Nombre lógico : Usuarios
- Nombre físico : Usuarios
- Descripción : Datos de los usuarios que harán uso del sistema
- Clave primaria : nombre_usuario
- Longitud de registro : 268

Tabla 2-1. Tabla de Usuarios

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
nombre_usuario	Varchar	20	Nombre para identificar a un usuario
correo_electronico	Varchar	70	Correo electrónico que tenga un usuario
contraseña	Varchar	128	Caracteres alfanuméricos que tendrá una cuenta de usuario para acceder al sistema

nombre_rol	Varchar	50	Número correlativo para identificar el rol de un usuario

Fuente : Elaboración propia.

2.2.2.2. <u>Productos.</u>

- Nombre lógico : Productos
- Nombre físico : Productos
- Descripción : Datos de todos los productos que se manejan en el taller mecánico para reparar los vehículos
- Clave primaria : codigo_producto
- Clave(s) foránea(s) : rut_proveedor (hacia la tabla Proveedores) , nombre_usuario(hacia la tabla Usuarios)
- Longitud de registro : 206

Tabla 2-2. Tabla de productos

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_producto	Int	6	Número correlativo para identificar a un producto
nombre_producto	Varchar	50	Nombre de un producto
descripcion_producto	Varchar	70	Descripción de un producto
marca_producto	Varchar	25	Marca de un producto
categoria_producto	Varchar	20	Nombre de una categoría en que se encuentra un producto
stock	SmallInt	4	Número que indica la cantidad actual de un producto
estado	Varchar	1	Indica el estado en que se encuentra un producto ejemplo : V (Vigente) , N(No vigente)
rut_proveedor	Varchar	10	Identificador de un proveedor

nombre_usuario	Varchar	20	Nombre para identificar a un usuario

Fuente : Elaboración propia.

2.2.2.3. Proveedores.

- Nombre lógico : Proveedores
- Nombre físico : Proveedores
- Descripción : Datos sobre los proveedores , a los cuales se les han comprado los productos.
- Clave primaria : rut_proveedor
- Clave(s) foránea(s) : No posee clave foráneas.
- Longitud de registro : 132

Tabla 2-3. Tabla de Proveedores

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
rut_proveedor	Varchar	10	Identificador de un proveedor
nombre_proveedor	Varchar	60	Nombre del proveedor
ubicacion_proveedor	Varchar	50	Dirección del proveedor
telefono_proveedor	Varchar	12	Teléfono del proveedor

Fuente : Elaboración propia.

2.2.2.4. Compra de Productos.

- Nombre lógico : Compra de Productos
- Nombre físico : Compra_Productos
- Descripción : Contiene los datos de todos los números de facturas de una compra . Dicha factura es originada por la compra de productos a un proveedor.
- Clave primaria : codigo_factura
- Clave(s) foránea(s) : rut_proveedor (hacia la tabla Proveedores)

- Longitud de registro : 26

Tabla 2-4. Tabla de Compra de Productos

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_factura	MediumInt	6	Número correlativo para identificar una factura (código único)
fecha_compra	Date	10	Fecha de emisión de la factura de compra
rut_proveedor	Varchar	10	Identificador de un proveedor.

Fuente : Elaboración propia.

2.2.2.5.Detalle de Compra.

- Nombre lógico : Detalle de compra
- Nombre físico : Detalle_Compras
- Descripción : Datos de los detalles de una compra de productos asociado a un código de factura
- Clave primaria : codigo_factura + codigo_producto
- Clave(s) foránea(s) : codigo_factura (hacia la tabla Compra_Productos) , codigo_producto (hacia la tabla Productos)
- Longitud de registro : 73

Tabla 2-5. Tabla Detalle de Compra.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_factura	MediumInt	6	Número correlativo para identificar una compra
codigo_producto	Int	6	Número correlativo para identificar a un producto

descripcion_compra_producto	Varchar	50	Descripción del producto comprado
cantidad_compra	SmallInt	4	Cantidades de un producto comprado
comprado_al_valor	Int	7	Valor del producto

Fuente : Elaboración propia.

2.2.2.6. Retiro de Productos.

- Nombre lógico : Retiro de Productos
- Nombre físico : Retiro_Productos
- Descripción : Datos de los retiros de productos que son ocupados para una reparación
- Clave primaria : codigo_retiro_producto
- Clave(s) foránea(s) : codigo_producto (hacia la tabla Productos)
- Longitud de registro : 76

Tabla 2-6. Tabla de Retiro de Productos

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_retiro_producto	Int	6	Número correlativo para identificar el retiro de un producto
descripcion_retiro_producto	Varchar	50	Descripción del retiro de productos
cantidad_retiro_producto	SmallInt	4	Cantidades retiradas de un producto
fecha_retiro_producto	Date	10	Fecha de retiro del producto
codigo_producto	Int	6	Número correlativo para identificar a un producto

Fuente : Elaboración propia.

2.2.2.7. Ajuste de Productos

- Nombre lógico : Ajuste de Productos
- Nombre físico : Ajuste_Productos
- Descripción : Datos referentes al ajuste de un producto respecto al stock ya sea por pérdida , robos , en mal estado , entre otros.
- Clave primaria : codigo_ajuste
- Clave(s) foránea(s) : codigo_producto (hacia la tabla Productos)
- Longitud de registro : 77

Tabla 2-7. Tabla Ajustes de Productos.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_ajuste	Int	6	Número correlativo para identificar el ajuste de un producto
tipo_ajuste	Byte	1	Indica el tipo de ajuste de un producto : 0(incremento producto) , 1(pérdida) , 2(consumo) ,3(robo) , 4(en mal estado) ,5(otros).
descripcion_ajuste	Varchar	50	Descripción del ajuste de un producto
cantidad_ajuste	SmallInt	4	Cantidad ajustada del producto
fecha_ajuste	Date	10	Fecha en que se realizó el ajuste del producto.
codigo_producto	Int	6	Número correlativo para identificar a un producto

Fuente : Elaboración propia.

2.2.2.8. Vehículos.

- Nombre lógico : Vehículos

- Nombre físico : Vehiculos
- Descripción : Datos de todos los vehículos de los clientes.
- Clave primaria : patente_vehiculo
- Clave(s) foránea(s) : rut_cliente (hacia la tabla Clientes)
- Longitud de registro : 76

Tabla 2-8. Tabla de Vehículos.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
patente_vehiculo	Varchar	6	Identificador de un vehículo
marca_vehiculo	Varchar	20	Marca del vehículo
modelo_vehiculo	Varchar	20	Serie o nombre del modelo que produce una marca de un vehículo
fecha_recepcion_vehiculo	Date	10	Fecha de recepción del vehículo del cliente
fecha_entrega_vehiculo	Date	10	Fecha de entrega del vehículo al cliente
rut_cliente	Varchar	10	Identificador de un cliente

Fuente : Elaboración propia.

2.2.2.9. **<u>Clientes.</u>**

- Nombre lógico : Clientes
- Nombre físico : Clientes
- Descripción : Contiene todos los datos personales de los clientes que recibieron los servicios entregados por el taller , pues en esta tabla se controla la situación financiera del cliente .

- Clave primaria : rut_cliente
- Clave(s) foránea(s) : nombre_usuario (hacia la tabla Usuarios)
- Longitud de registro : 223

Tabla 2-9. Tabla de Clientes.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
rut_cliente	Varchar	10	Identificador del cliente
nombre_cliente	Varchar	60	Nombre completo del cliente
telefono_cliente	Varchar	12	Teléfono del cliente
dirección_cliente	Varchar	60	Dirección del cliente
correo_cliente	Varchar	60	Correo del cliente
situacion_financiera_cliente	Varchar	1	Es la situación financiera que mantiene cada cliente con el taller como ser : D(Deudor) , P(pago completo)
nombre_usuario	Varchar	20	Nombre para identificar a un usuario

Fuente: Elaboración propia.

2.2.2.10. <u>Deudas.</u>

- Nombre lógico : Deudas
- Nombre físico : Deudas
- Descripción : Datos referentes al detalle de la deuda que posee un cliente en situación financiera deudor.
- Clave primaria : codigo_deuda

- Clave(s) foránea(s) : rut_cliente (hacia la tabla Clientes), patente_vehiculo(hacia la tabla Vehiculos)
- Longitud de registro : 72

Tabla 2-10. Tabla de Deudas.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_deuda	Int	10	Número correlativo para identificar una deuda
deuda_valor_trabajo	Int	7	Valor total del servicio realizado
monto_a_pagar	Int	7	Valor de la deuda a pagar
forma_repactacion_deuda	Varchar	20	Indica la forma de repactar una deuda : cuotas , transferencia , efectivo . Solo en el caso de ser cuotas se especifica la cantidad de cuotas que se debe colocar al lado ej : forma de repacto : 6 cuotas.
Estado_deuda	Varchar	2	Indica el estado de la deuda : PP(por pagar) , PG(pagado) , AT(atrasado).
Fecha_vencimiento	Date	10	Fecha de vencimiento de la deuda
rut_cliente	Varchar	10	Identificador del cliente
patente_vehiculo	Varchar	6	Identificador de un vehículo

Fuente : Elaboración propia.

2.2.2.11. **<u>Pagos.</u>**

- Nombre lógico : Pagos
- Nombre físico : Pagos
- Descripción : Datos referentes a los pagos realizados por los clientes deudores
- Clave primaria : codigo_pago

- Clave(s) foránea(s) : codigo_deuda (hacia la tabla Deuda).
- Longitud de registro : 37

Tabla 2.11. Tabla de Pago.

DESCRIPCIÓN DE REGISTRO			
NOMBRE	**TIPO DE DATO**	**LONGITUD**	**DESCRIPCIÓN**
codigo_pago	Int	10	Número correlativo para identificar el pago de una deuda
monto_pagado	Int	7	Valor que ha pagado el cliente con situación financiera deudor
fecha_pago	Date	10	Fecha en el que se realizó el pago
codigo_deuda	Int	10	Número correlativo para identificar una deuda

Fuente : Elaboración propia.

2.3. DIAGRAMA DE MENÚS

Si bien , cabe señalar que el sistema abarca un control de inventario y deudores , sus funcionalidades correspondientes como por ejemplo los mantenedores (productos , proveedores , compra de productos , detalle de compra , retiro de productos y ajustes del producto) y los módulos (clientes , vehículos , deudas y pagos) , se distribuirán en diferentes secciones del menú que a continuación se muestran en el diagrama.

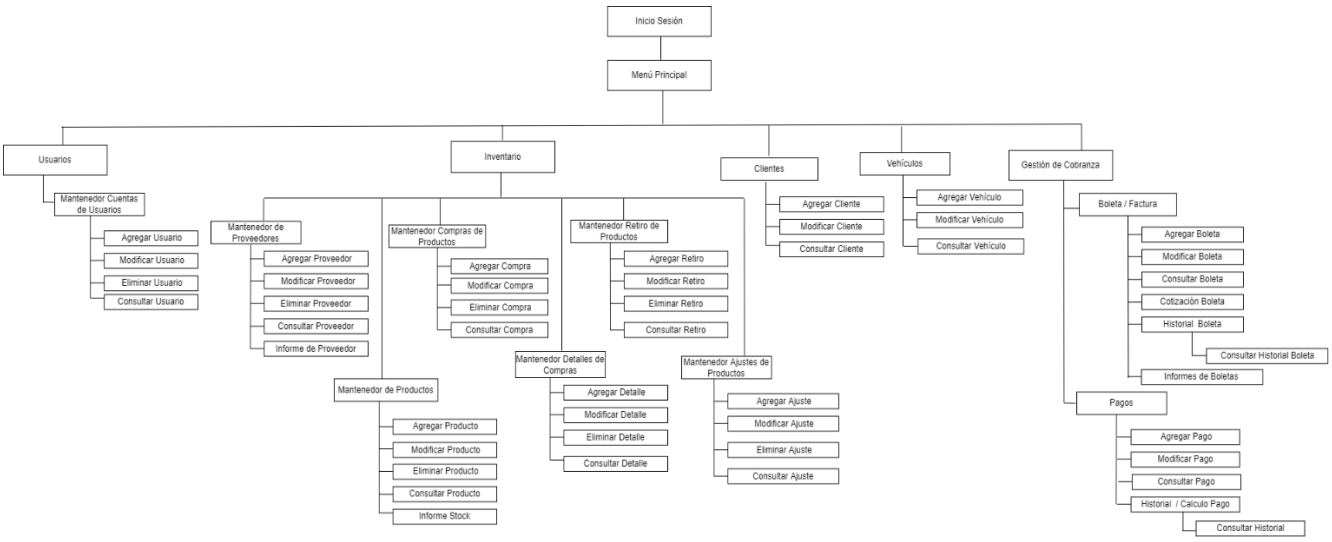


Figura 2-1. Diagrama de Menús

Fuente: Elaboración Propia

2.4. LISTADO DE TOTALIDAD DE PROGRAMAS DEL SISTEMA.

Nombre	*Objetivo*
Login	Permite el ingreso al sistema
Mantenedor Cuentas de usuarios	Permite agregar , modificar , eliminar y consultar las cuentas de usuario pertenecientes al sistema
Mantenedor de Proveedores	Permite agregar , modificar , eliminar y consultar los proveedores
Mantenedor de Productos	Permite agregar , modificar , eliminar y consultar los diferentes productos que mantienen o ingresan al taller mecánico.
Mantenedor Compras de Productos	Permite registrar las compras de los productos dando la posibilidad de agregar , modificar , eliminar y consultar dicha compra.
Mantenedor Detalle de Compras	Permite registrar el detalle de la compra de los productos dando la posibilidad de agregar , modificar , eliminar y consultar un detalle.
Mantenedor Retiro de Productos	Permite registrar y efectuar el retiro de un producto que se requiera para la reparación de los vehículos
Mantenedor Ajustes de Productos	Permite registrar el ajuste de un producto en cuento a pérdidas , extravíos , etc. Asimismo, permite agregar , modificar , eliminar y consultar un ajuste
Clientes	Permite conocer y registrar los diferentes clientes deudores que realizaron uso del servicio del taller . Además , posibilita agregar, modificar y consultar.

Vehículos	Permite Conocer y registrar los diferentes vehículos de los deudores que fueron reparados . Además , posibilita agregar , modificar y consultar un vehículo
Boleta / Factura	Permite registrar , conocer y administrar las cuentas pendientes y emitidas de las cuotas del servicio que el cliente deudor acredita. Además posibilita , agregar , modificar y consultar , ver historial y cotizar.
Pagos	Permite conocer y administrar los pagos realizados por los clientes deudores . Además posibilita agregar , modificar , consultar , ver historial y calcular un pago.

2.5 DISEÑO DE PANTALLAS PRINCIPALES.

A continuación , se presentan las pantallas principales del sistema con una breve descripción

2.5.1 Login

- Pantalla correspondiente al inicio de sesión donde se permite ingresar las credenciales de acceso al sistema debidamente validado.

Figura 2-2. Login del sistema.

Fuente : Elaboración Propia.

2.5.2 Menú Principal

- Pantalla correspondiente al menú principal del sistema , donde se encuentran sus respectivas secciones con submenús y alerta de déficit de stock de productos.

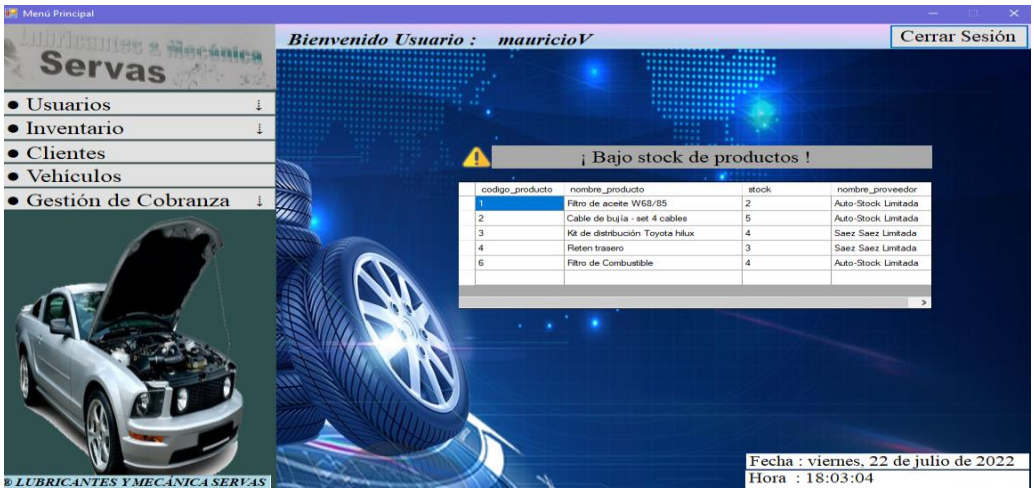


Figura 2-3. Menú Principal del sistema.

Fuente : Elaboración Propia.

2.5.2.1 Submenú de usuarios

- Pantalla corresponde al submenú de usuarios que contiene el mantenedor de cuentas de usuarios .

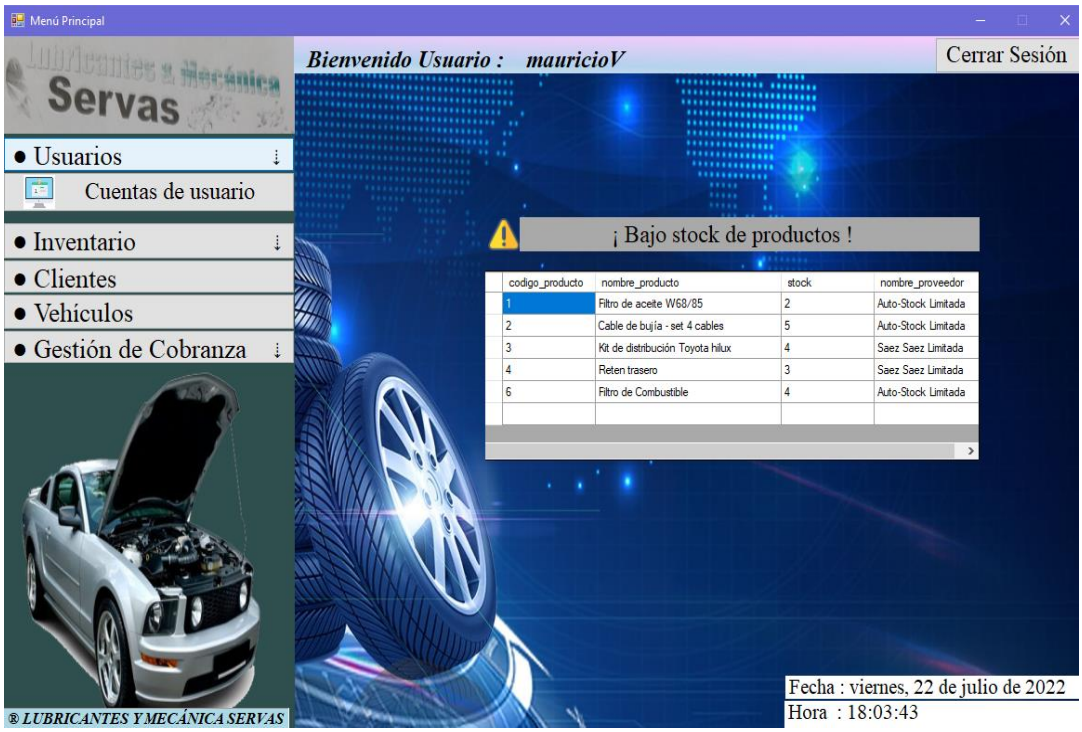


Figura 2-4. Menú Principal del sistema con submenú de Usuarios.

Fuente : Elaboración Propia.

2.5.2.2 Submenú de inventario

- Pantalla corresponde al submenú de inventario que contiene los mantenedores de proveedor , productos , compra de productos , detalle de compras (entradas de stock) , retiro de productos(salidas de stock) y ajustes de productos

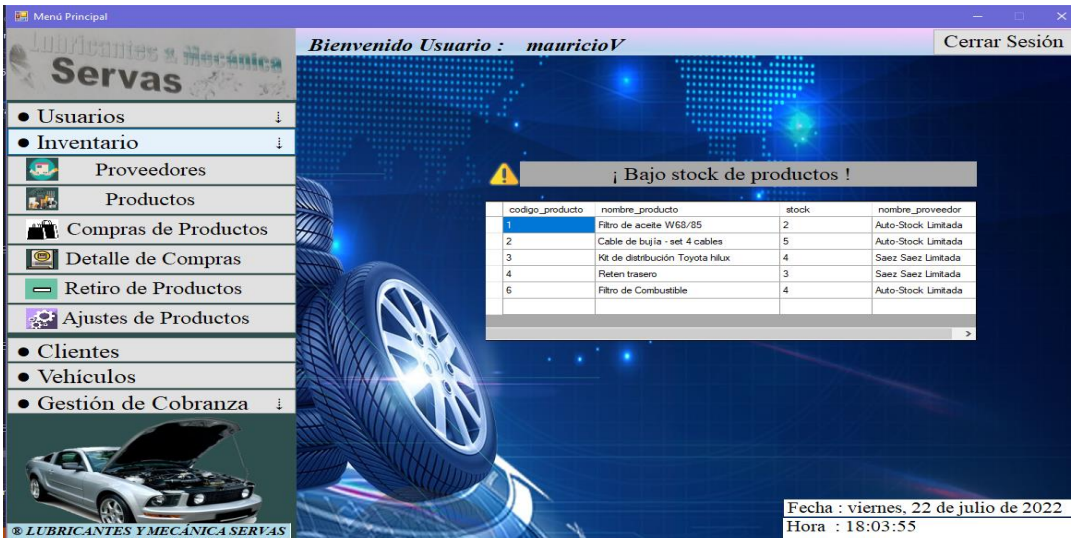


Figura 2-5. Menú Principal del sistema con submenú Inventario

Fuente : Elaboración Propia.

2.5.2.3 Submenú de gestión de cobranza

- Pantalla corresponde al submenú de gestión de cobranza que contiene los módulos de boleta/ factura y pagos . Esta sección está encargada de controlar a los deudores de acuerdo con los pagos realizados , junto con ello en el módulo de clientes se podrá apreciar si es deudor o no.

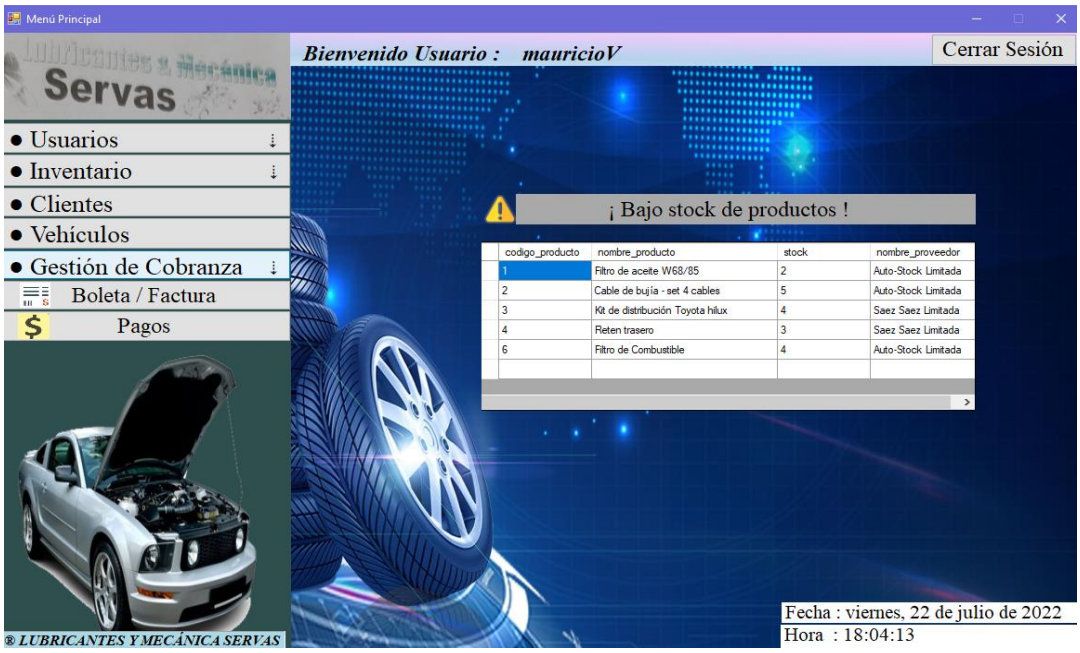


Figura 2-6. Menú Principal del sistema con submenú Gestión de Cobranza.

Fuente : Elaboración Propia.

2.5.3 Proveedores

- Pantalla correspondiente al ingreso de proveedores de los cuales abastecen el taller de acuerdo con la compra de repuestos.

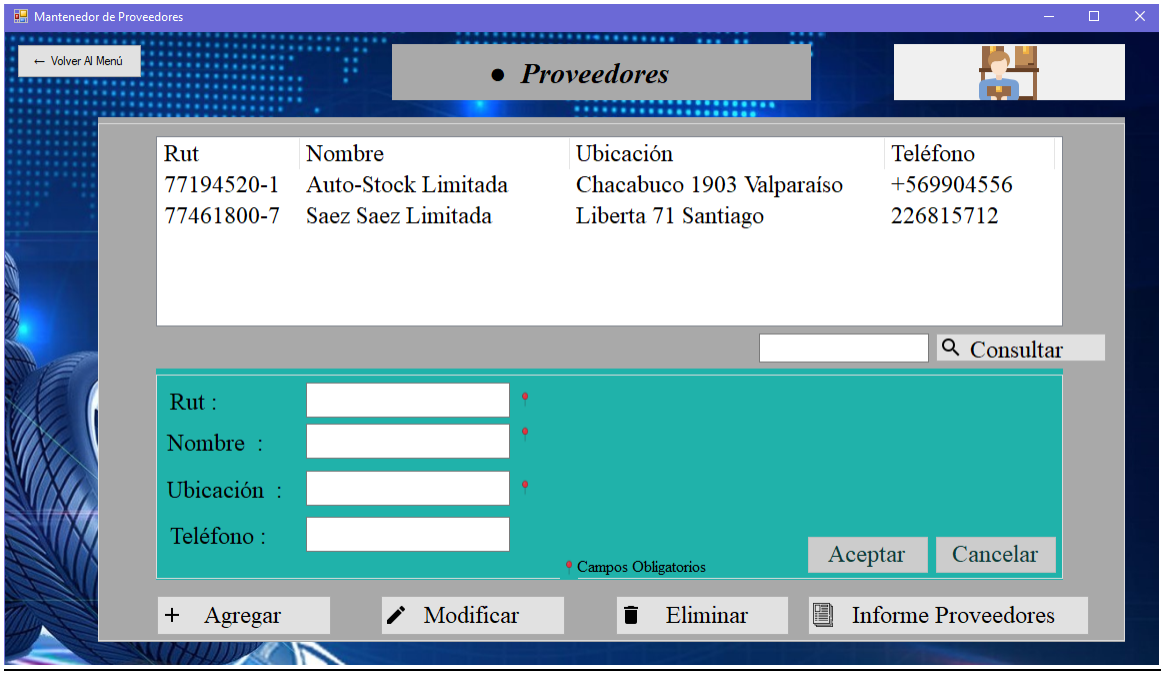


Figura 2-7. Mantenedor de proveedores

Fuente : Elaboración Propia.

2.5.4 Productos

- Pantalla correspondiente al ingreso de productos del taller mecánico , en él se refleja el stock que mantiene cada producto de acuerdo con los ajustes realizados en cuanto a su incremento y/o reducción .

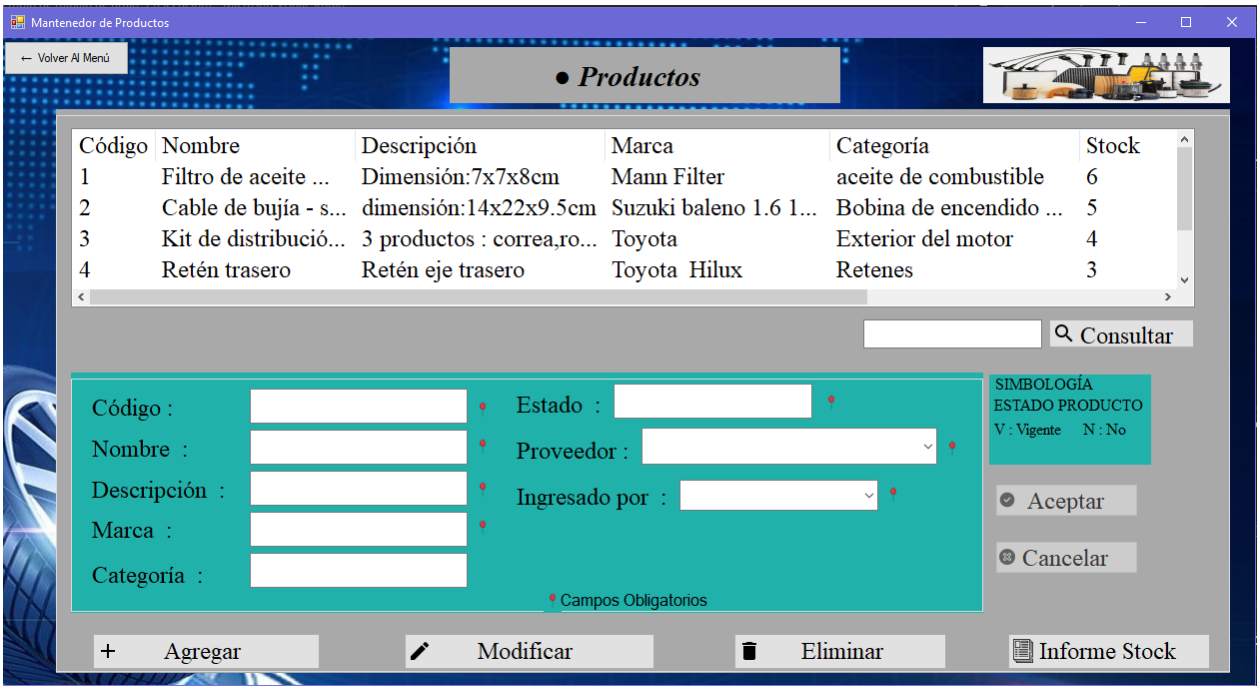


Figura 2-8. Mantenedor de productos.

Fuente : Elaboración Propia.

2.5.5 Detalle de compras

- Pantalla correspondiente a la entrada de productos (incremento de stock) y posteriormente el stock final se verá reflejado en la pantalla productos.

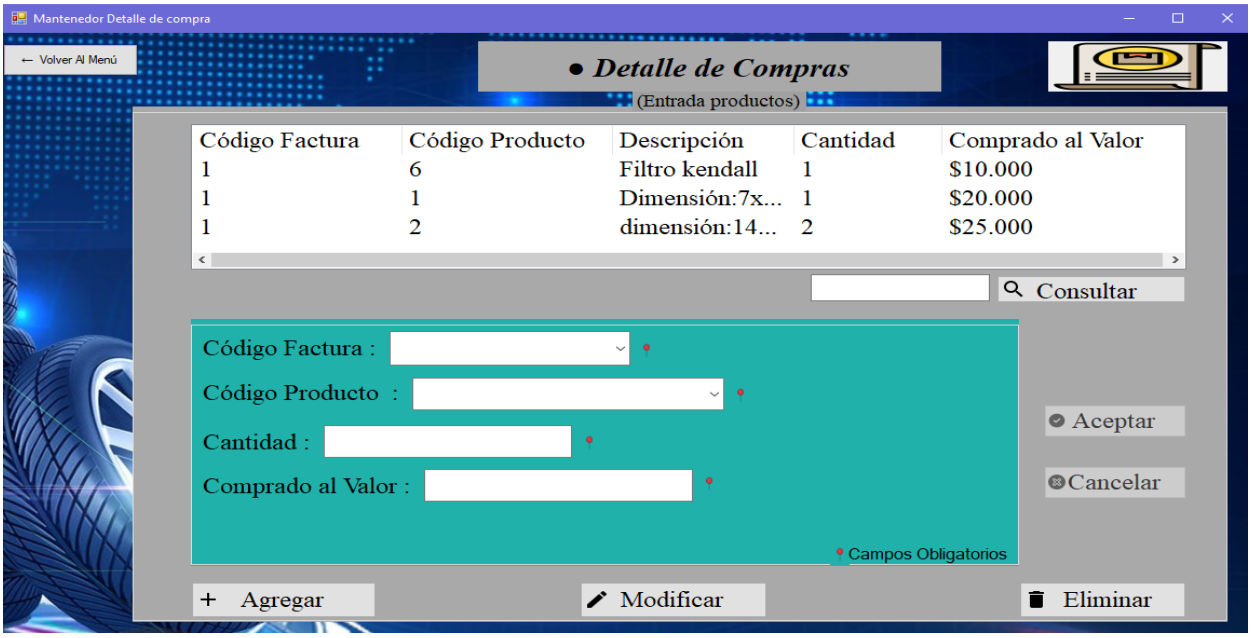


Figura 2-9 . Mantenedor detalle de compras.

Fuente : Elaboración Propia.

2.5.6 Retiro de Productos

- Pantalla correspondiente al retiro de productos , los cuales se han ocupado para las reparaciones (Control salida Stock).

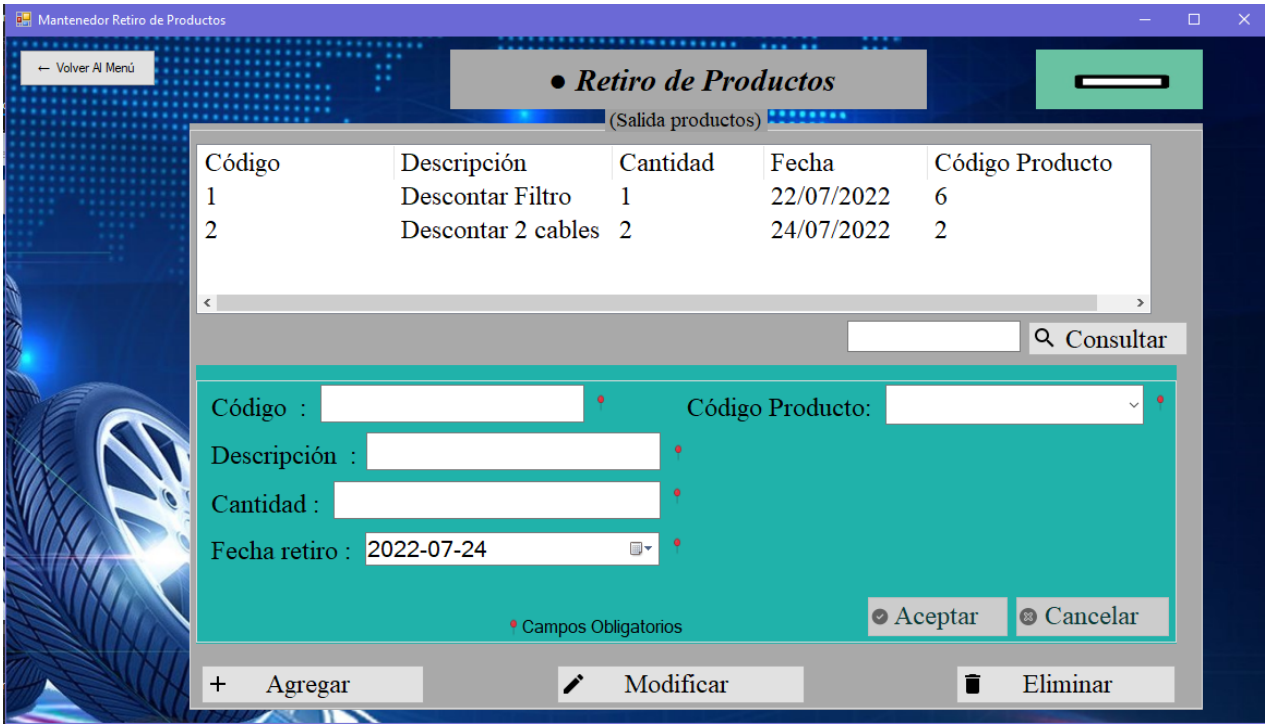


Figura 2-10. Mantenedor Retiro de Productos.

Fuente : Elaboración Propia.

2.5.7 Ajustes de Producto

- Pantalla correspondiente al ajuste realizado a un producto por concepto de robo, extravío, etc. (incrementos o perdidas de stock)

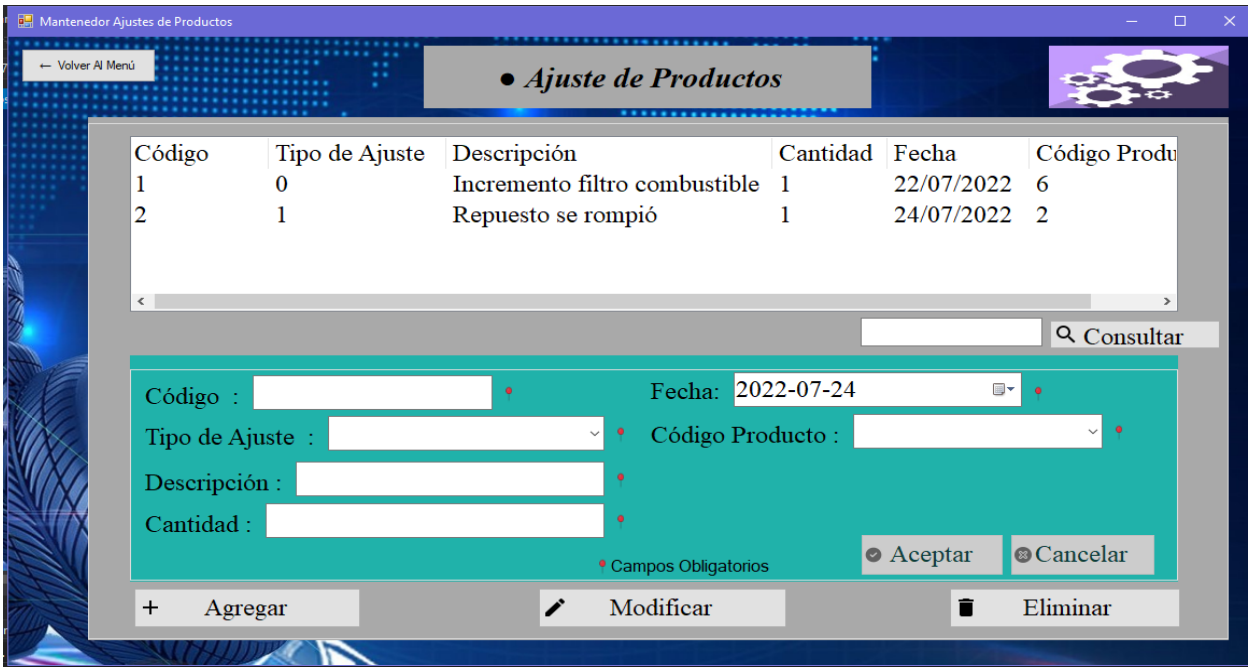


Figura 2-11. Mantenedor Ajuste de Productos .

Fuente : Elaboración Propia.

2.5.8 Clientes

- Pantalla correspondiente al ingreso de clientes donde se refleja su situación financiera si es deudor o no .

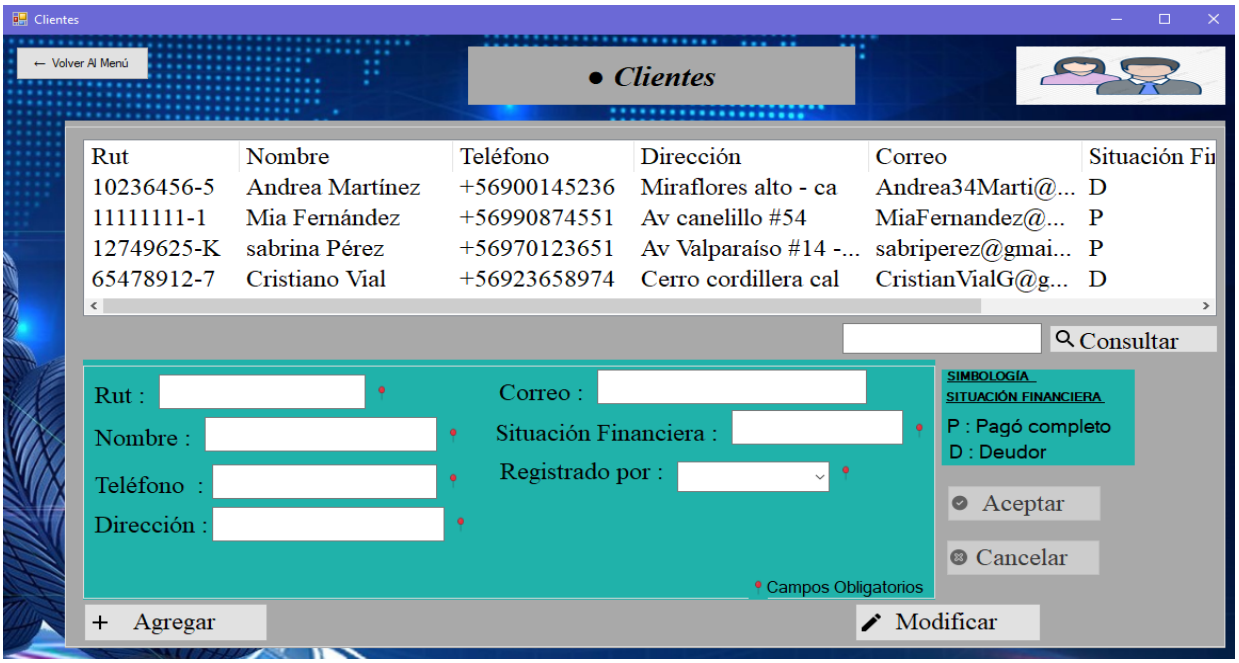


Figura 2-12. Módulo de Clientes.

Fuente : Elaboración Propia.

2.5.9 Vehículos

- Pantalla correspondiente al ingreso de un vehiculo asociado al Rut de un cliente , junto con ello permite reconocer un vehiculo de un cliente asociado a una deuda por la reparación o mantención del él.

Figura 2-13. Módulo de Vehículos

Fuente : Elaboración Propia.

2.5.10 Boleta / Factura

- Pantalla correspondiente al ingreso de una boleta (cuenta) que debe pagar un cliente deudor por el servicio entregado.

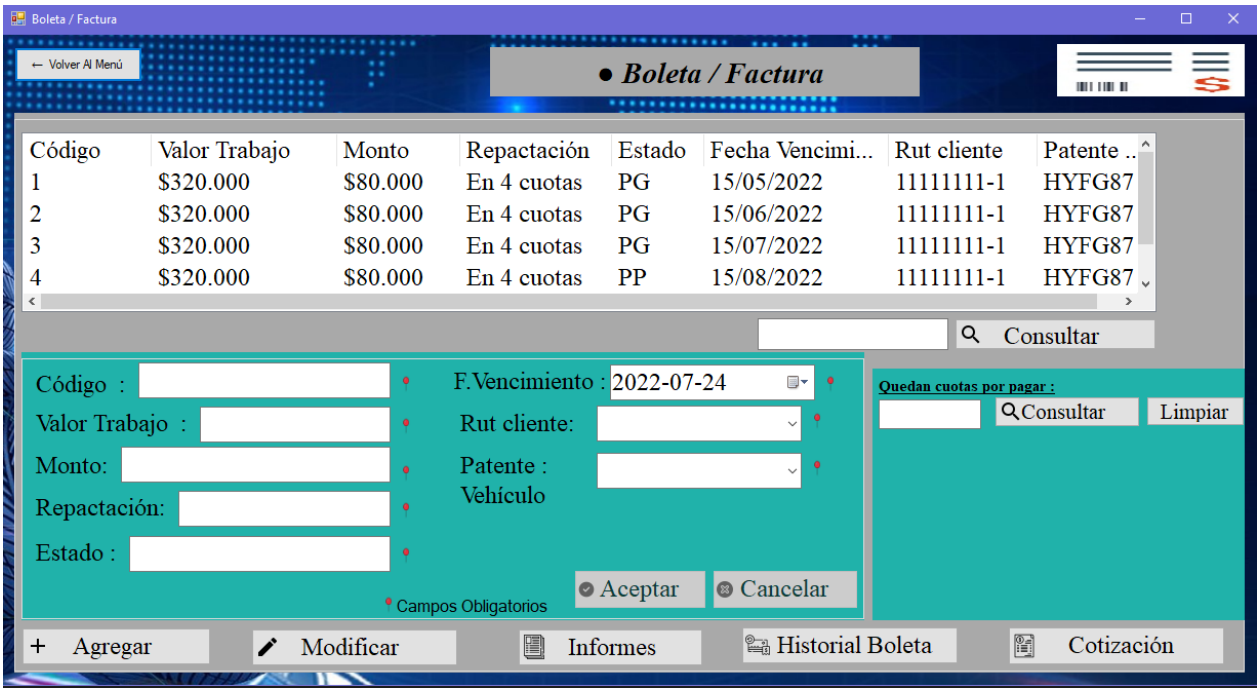


Figura 2-14. Módulo de Boleta / Factura .

Fuente : Elaboración Propia.

2.5.11 Pagos

- Pantalla correspondiente al ingreso de pagos realizados por los clientes deudores.

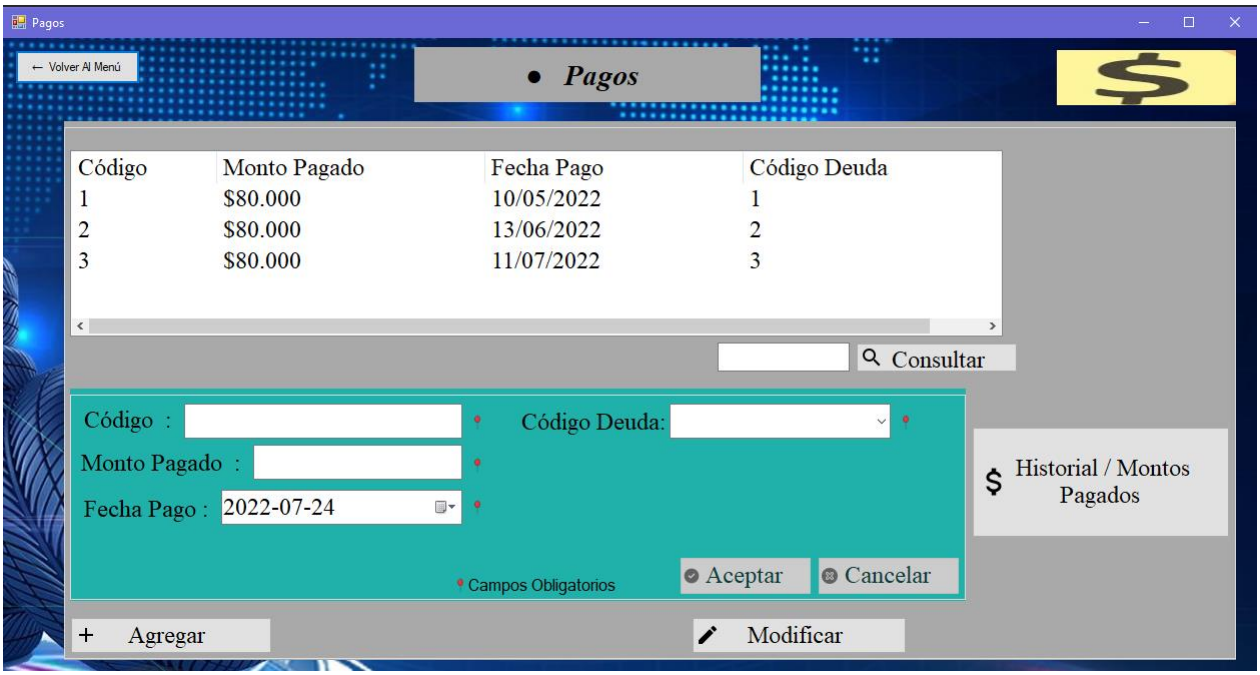


Figura 2-15. Módulo de Pagos.

Fuente : Elaboración Propia.

2.5.12. Ejemplos de algunas pantallas para control de inventario

2.5.12.1. Ingreso de Producto

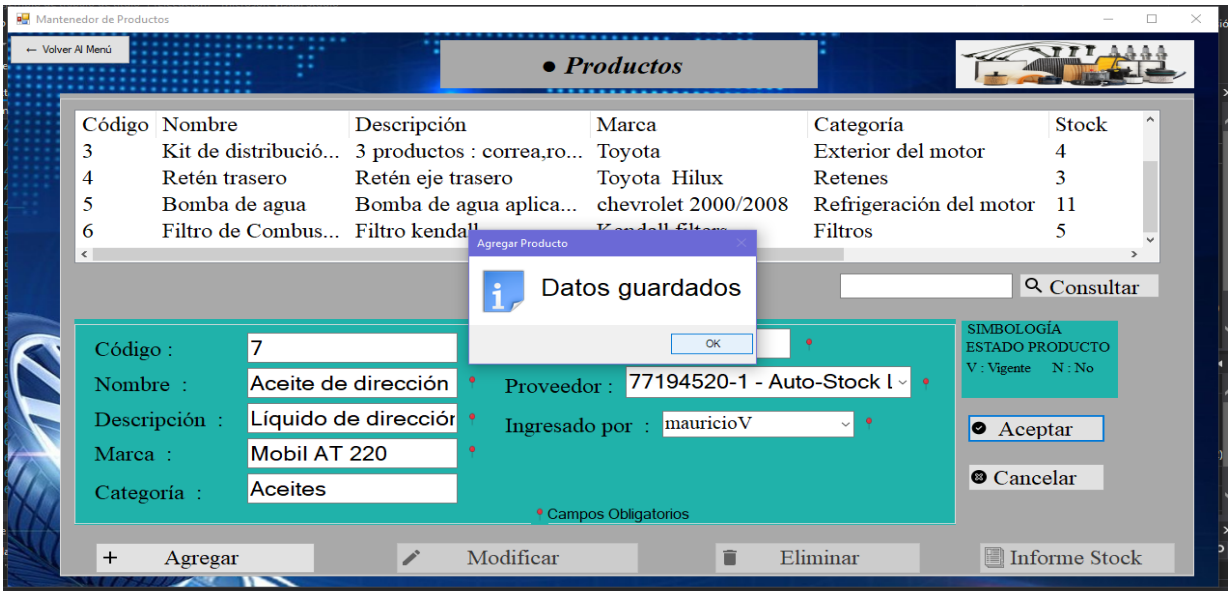


Figura 2-16. Ingreso de Producto con código 7.

Fuente : Elaboración Propia.

2.5.12.2. Ingreso de aumento de stock

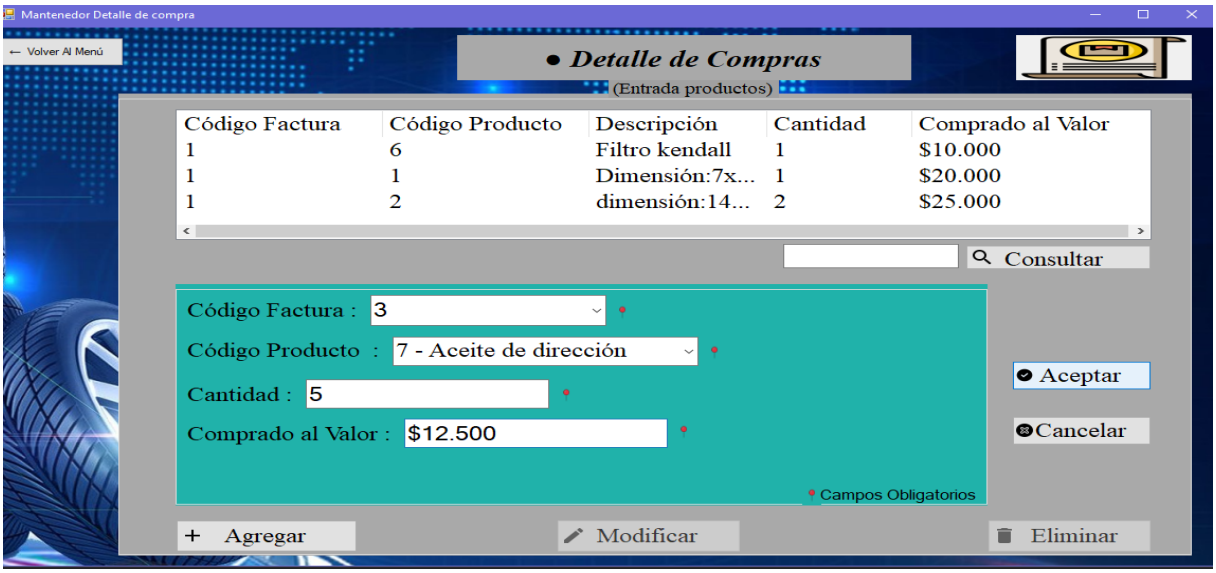


Figura 2-17.Ingreso de aumento de stock del código 7 .

Fuente : Elaboración Propia.

2.5.12.3. Stock de producto actualizado en su incremento

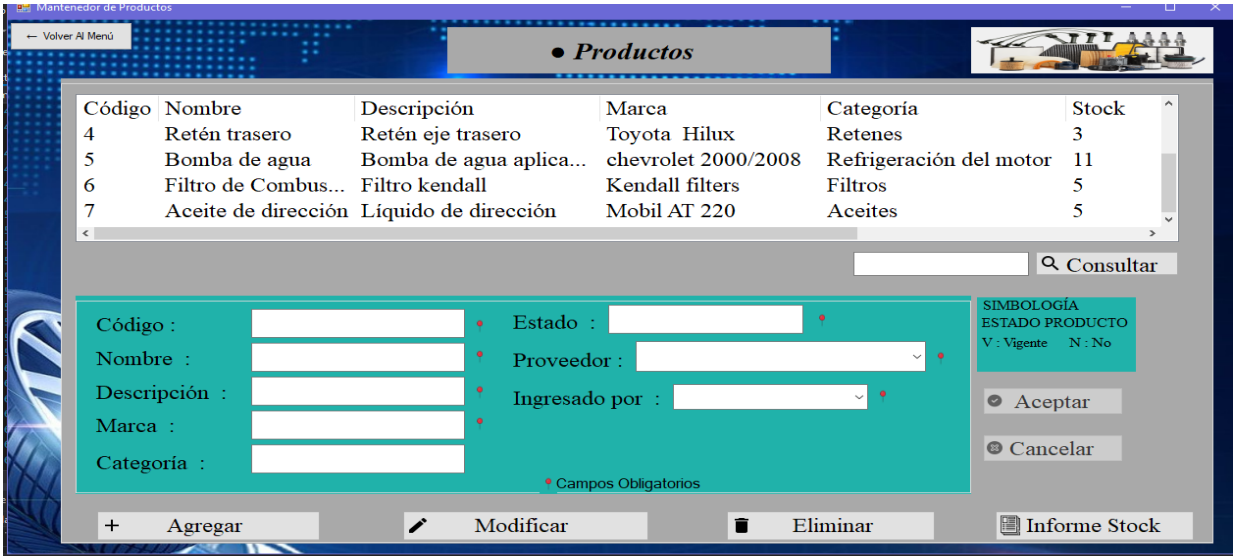


Figura 2-18. Stock de producto con código 7 actualizado en su incremento .

Fuente : Elaboración Propia.

2.5.12.4. Ingreso descuento de stock

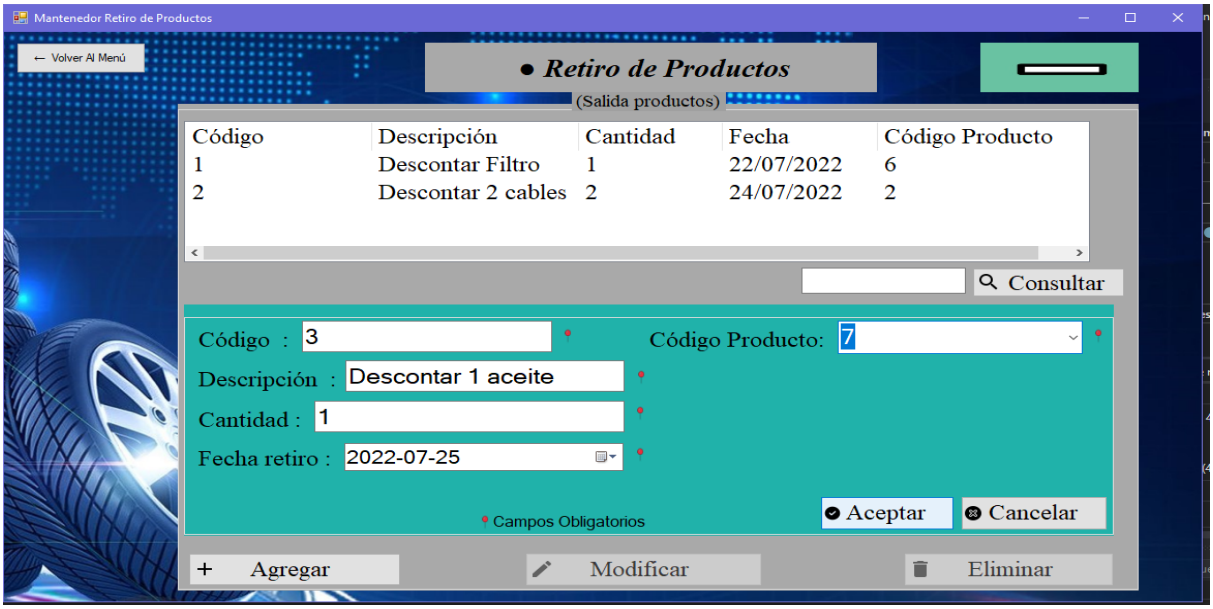


Figura 2-19. Ingreso descuento de stock código 7 .

Fuente : Elaboración Propia.

2.5.12.5. Stock actualizado en su descuento

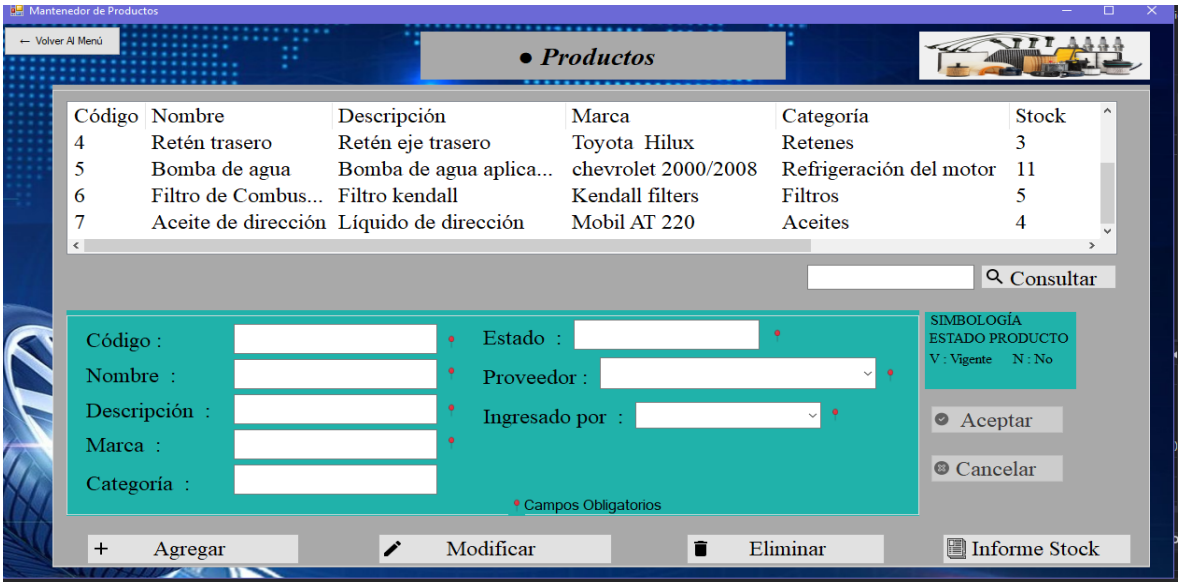


Figura 2-20. Stock con código 7 actualizado en su descuento .

Fuente : Elaboración Propia.

CONCLUSIONES

La implementación de sistemas es importante para reducir costos en cuanto a tiempo y a brindar un apoyo en la administración , así posibilitar la coordinación en el entorno de trabajo . En consecuencia y en virtud de lo aplicado , se comienza con el desarrollo de este trabajo durante octubre de 2021 donde se construye el sistema a partir de sus aspectos relevantes en cuanto a su análisis y diseño . Mientras que , el desarrollo del código empieza a finales de marzo , pues para lograr los objetivos correspondientes a nivel de implementación se debió indagar a través de documentos la información de algunas herramientas de software a aplicar , como también , la implementación de reportes (ReportViewer) , manejar una base de datos mysql y eventos de Windows Forms de acuerdo con sus componentes . Lo mencionado anteriormente , tiene lugar en el entorno de desarrollo visual studio – visual Basic . Sin embargo , como resultado final fue un extenso proyecto debido a las necesidades que se clarifica dentro de éste . De esta manera , se termina de concluir este primer semestre 2022 con la construcción completa del sistema requerido . Si bien , si este sistema desea ser implementado estará a disposición de la sociedad si lo requiere. No obstante, si ha de existir alguna al proyección del sistema se debe hacer frente a lo que requiera la sociedad , puede ser enfocado a seguir proyectándose en el control de deudores o implementar otras funciones que estimen convenientes tanto para el inventario como para otras opciones futuras.

En efecto , también se pueden realizar mejoras al sistemas sobre la visualización o adecuar algún registro , código de identificación o bien , eliminar la inserción de factura de compra de producto junto con el detalle para tratar las salidas o entradas de inventario en el mantenedor de productos , etc.

Para finalizar , lograr con cumplir los objetivos propuestos principalmente en el control de inventario es una de las propuestas más importantes del sistema , debido a que debe reflejarse el control de stock de los repuestos . Por otro lado la formación profesional es importante gracias

a los contenidos de base que proporcionó la carrera , permitiendo el desarrollo del sistema y realizar un buen manejo a nivel de software como también de interfaz gráfica acompañado de documentos anexos para la realización de éste.

BIBLIOGRAFÍA

Documentación :

- **Mysql : Consultas y manejo de base de datos**
  - https://downloads.mysql.com/docs/refman-5.0-es.pdf

- **Mysql : Conector de consultas**
  - https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-data-adapter.html

- **Conversiones :**
  - https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/functions/type-conversion-functions

- **Eventos de propiedades Windows Forms :**
  - https://docs.microsoft.com/es-es/dotnet/desktop/winforms/order-of-events-in-windows-forms?view=netframeworkdesktop-4.8

- **ReportViewer : Visual Basic**
  - https://si.ua.es/es/documentacion/informes-net/documentos/apuntes-de-referencia/reportviewer.pdf

- **ReportViewer – orígenes de datos :**
  - https://si.ua.es/es/documentacion/informes-net/documentos/apuntes-de-referencia/origen-de-datos.pdf

- **Lectura Visual Basic 2012 : paso a paso**
  - Autor : James Foxall

- Editorial : Anaya Multimedia
- Año : 2013
- Nº de páginas : 638

ANEXO

CÓDIGO DEL SITEMA

- Mantenedor de productos

```
Imports System.Data.Sql

Imports System.Text

Imports MySql.Data.MySqlClient

Imports System.Security.Cryptography

Public Class Productos

   Dim Conex As New MySqlConnection

   Dim Conexi As New MySqlConnection

   Dim mysql As New MySqlDataAdapter

   Dim mysql1 As New MySqlDataAdapter

   Dim mys As MySqlDataReader

   Dim Mysqll As DataSet

   Dim Mysqll1 As DataSet

   Dim datos As New MySqlCommand

   Dim mostrar As New DataTable

   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

      Me.Close()

      Form2.Show()

   End Sub
```

```
    Private Sub Productos_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Conex.ConnectionString= "server=localhost;user=root;password=
;database=taller_mecanico"
        If Conex.State = ConnectionState.Open Then
            Conex.Close()
        End If
        Conex.Open()
        MostrarListview()
        TxtNombreProd.Clear()
        Button_informeProd.Visible = True
        Btn_AceptarProd.Enabled = False
        Btn_cancelarProd.Enabled = False
    End Sub
    Public Sub MostrarCombox()
        Try
            Conex.Open()
            Dim consulta As String = String.Empty
            Dim i As Integer
            consulta = "SELECT * FROM  proveedores"
            mysql = New MySqlDataAdapter(consulta, Conex)
            Mysqll = New DataSet
```

```
Mysqll.Tables.Add("proveedores")

mysql.Fill(Mysqll.Tables("proveedores"))

Dim rut As String

Dim nombre As String

If Mysqll.Tables(0).Rows.Count <> 0 Then

   For i = 0 To Mysqll.Tables(0).Rows.Count - 1

      rut = Mysqll.Tables(0).Rows(i).Item("rut_proveedor")

      nombre = Mysqll.Tables(0).Rows(i).Item("nombre_proveedor")

      ComboBox_ProveedorProd.Items.Add(rut & " | " & nombre)





   Next

End If

Dim consultas As String = String.Empty

consultas = "SELECT * FROM  usuarios"

mysql1 = New MySqlDataAdapter(consultas, Conex)

Mysqll1 = New DataSet

Mysqll1.Tables.Add("usuarios")

mysql1.Fill(Mysqll1.Tables("usuarios"))

ComboBox_nombreusuarioP.DataSource = Mysqll1.Tables("usuarios")
```

```
        ComboBox_nombreusuarioP.DisplayMember = "nombre_usuario"
    Catch ex As Exception
    End Try
End Sub
Public Sub Limpiar()
    For i = 0 To List_productos.Items.Count - 1
        List_productos.Items.Clear()
    Next
End Sub
Public Sub RefrescarAgregar()
    Txt_CodigoProd.Clear()
    TxtNombreProd.Clear()
    Txt_DescripcionProd.Clear()
    TxtMarcaProd.Clear()
    Txt_CategoriaProd.Clear()
    Txt_EstadoProd.Clear()
    ComboBox_ProveedorProd.DataSource = Nothing

    ComboBox_nombreusuarioP.DataSource = Nothing
    Btn_AgregarProd.Enabled = True
    Btn_ModificarProd.Enabled = True
```

```
    Btn_eliminarPord.Enabled = True
    ComboBox_ProveedorProd.Text = ""
    Button_informeProd.Enabled = True
    Btn_AceptarProd.Enabled = False
    Btn_cancelarProd.Enabled = False
End Sub
Public Sub MostrarListview()
    Dim Li As New ListViewItem
    Try
        Dim consulta As String = String.Empty
        consulta = "SELECT * FROM productos"
        datos.CommandText = consulta
        datos.Connection = Conex
        mys = datos.ExecuteReader
        While mys.Read
            Li = List_productos.Items.Add(mys.Item("codigo_producto").ToString)
            Li.SubItems.Add(mys.Item("nombre_producto").ToString)
            Li.SubItems.Add(mys.Item("descripcion_producto").ToString)
            Li.SubItems.Add(mys.Item("marca_producto").ToString)
            Li.SubItems.Add(mys.Item("categoria_producto").ToString)
            Li.SubItems.Add(mys.Item("stock").ToString)
```

```
            Li.SubItems.Add(mys.Item("estado").ToString)
            Li.SubItems.Add(mys.Item("Proveedores_rut_proveedor").ToString)
            Li.SubItems.Add(mys.Item("Usuarios_nombre_usuario").ToString)
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub Btn_AgregarProd_Click(sender As Object, e As EventArgs) Handles
Btn_AgregarProd.Click
    MostrarCombox()
    Btn_AgregarProd.Enabled = True
    Btn_ModificarProd.Enabled = False
    Btn_eliminarPord.Enabled = False
    Txt_CodigoProd.Clear()
    TxtNombreProd.Clear()
    Txt_DescripcionProd.Clear()
    TxtMarcaProd.Clear()
    Txt_CategoriaProd.Clear()
    Txt_EstadoProd.Clear()
```

```
        Btn_AceptarProd.Enabled = True

        Btn_cancelarProd.Enabled = True

        Txt_CodigoProd.Enabled = True

        Button_informeProd.Enabled = False

    End Sub

    Private Sub Btn_AceptarProd_Click(sender As Object, e As EventArgs) Handles
Btn_AceptarProd.Click

        If Txt_CodigoProd.Text <> "" And TxtNombreProd.Text <> "" And Txt_DescripcionProd.Text
<> "" And TxtMarcaProd.Text <> "" And (Txt_CategoriaProd.Text <> "" Or
Txt_CategoriaProd.Text = "") And Txt_EstadoProd.Text <> "" And
ComboBox_ProveedorProd.Text <> "" And ComboBox_nombreusuarioP.Text <> "" And
Len(Txt_CodigoProd.Text) <= 6 And Len(TxtNombreProd.Text) <= 50 And
Len(Txt_DescripcionProd.Text) <= 70 And Len(TxtMarcaProd.Text) <= 25 And
(Len(Txt_CategoriaProd.Text) <= 20 Or Len(Txt_CategoriaProd.Text) = 0) And
Len(Txt_EstadoProd.Text) <= 1 Then


            Dim consultaCodigoPro As String = String.Empty

            consultaCodigoPro = "Select * from productos where codigo_producto = '" &
Txt_CodigoProd.Text & "'"

            datos.CommandText = consultaCodigoPro

            datos.Connection = Conex

            mys = datos.ExecuteReader

            Dim cant As Integer = 0

            While mys.Read
```

```
        cant = cant + 1
    End While
    If cant = 1 Then
        Dim msg As New mb.ShowMessagebox
        msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
        msg.ShowBox("codigo producto ya existe", 0, mb.FStyle.Exclamation, "producto")
        mys.Close()
    Else
        mys.Close()
        Dim cadenaComboArray As String
        cadenaComboArray = Mid(ComboBox_ProveedorProd.Text, 1, 10)
        If IsNumeric(Txt_CodigoProd.Text) And Not IsNumeric(TxtNombreProd.Text) And Not
IsNumeric(TxtMarcaProd.Text) And Not IsNumeric(Txt_EstadoProd.Text) Then


            datos.CommandText = "insert into
productos(codigo_producto,nombre_producto,descripcion_producto,marca_producto,categori
a_producto,stock,estado,Proveedores_rut_proveedor,Usuarios_nombre_usuario) values('" &
Txt_CodigoProd.Text & "','" & TxtNombreProd.Text & "','" & Txt_DescripcionProd.Text & "','" &
TxtMarcaProd.Text & "','" & Txt_CategoriaProd.Text & "','" & " " & "','" & Txt_EstadoProd.Text &
"','" & cadenaComboArray & "','" & ComboBox_nombreusuarioP.Text & "')"
            datos.ExecuteNonQuery()
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
```

```
            msg.ShowBox("Datos guardados", 0, mb.FStyle.Information, "Agregar Producto")
            Limpiar()
            MostrarListview()
            RefrescarAgregar()
        Else
            If Not IsNumeric(Txt_CodigoProd.Text) Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("codigo debe ser numerico", 0, mb.FStyle.Information, "Producto")
                Txt_CodigoProd.Clear()
                Txt_CodigoProd.Focus()
            End If
            If IsNumeric(TxtNombreProd.Text) Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("nombre no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
                TxtNombreProd.Clear()
                TxtNombreProd.Focus()

            End If
```

```
                If IsNumeric(TxtMarcaProd.Text) Then
                    Dim msg As New mb.ShowMessagebox
                    msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                    msg.ShowBox(" marca no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
                    TxtMarcaProd.Clear()
                    TxtMarcaProd.Focus()
                End If
                If IsNumeric(Txt_EstadoProd.Text) Then
                    Dim msg As New mb.ShowMessagebox
                    msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                    msg.ShowBox("Estado no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
                    Txt_EstadoProd.Clear()
                    Txt_EstadoProd.Focus()
                End If
            End If
        End If
    Else
        If Txt_CodigoProd.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
```

```
            msg.ShowBox("codigo no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf Txt_CodigoProd.Text <> "" Then
            If Len(Txt_CodigoProd.Text) > 6 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo codigo puede tener solo hasta 6 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                    Txt_CodigoProd.Clear()
                    Txt_CodigoProd.Focus()
                End If
            End If
        End If
        If TxtNombreProd.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("nombre no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf TxtNombreProd.Text <> "" Then
            If Len(TxtNombreProd.Text) > 50 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
```

```
            msg.ShowBox("Campo nombre puede tener solo hasta 50 digitos ", 0,
mb.FStyle.Exclamation, "productos")
            If MsgBoxResult.Ok = 1 Then
                TxtNombreProd.Clear()
                TxtNombreProd.Focus()
            End If
        End If
    End If
    If Txt_DescripcionProd.Text = "" Then
        Dim msg As New mb.ShowMessagebox
        msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
        msg.ShowBox("descripcion no debe estar vacio", 0, mb.FStyle.Information, "Producto")
    ElseIf Txt_DescripcionProd.Text <> "" Then
        If Len(Txt_DescripcionProd.Text) > 70 Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Campo descripcion puede tener solo hasta 70 digitos ", 0,
mb.FStyle.Exclamation, "productos")
            If MsgBoxResult.Ok = 1 Then
                Txt_DescripcionProd.Clear()
                Txt_DescripcionProd.Focus()
            End If
```

```
            End If
        End If
        If TxtMarcaProd.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("marca no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf TxtMarcaProd.Text <> "" Then
            If Len(TxtMarcaProd.Text) > 25 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo marca puede tener solo hasta 25 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                    TxtMarcaProd.Clear()
                    TxtMarcaProd.Focus()



                End If
            End If
        End If
        If Txt_EstadoProd.Text = "" Then
```

```
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Estado no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf Txt_EstadoProd.Text <> "" Then
            If Len(Txt_EstadoProd.Text) > 1 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo codigo puede tener solo hasta 1 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                    Txt_EstadoProd.Clear()
                    Txt_EstadoProd.Focus()
                End If
            End If
        End If
    End If
End Sub
Public Sub RefrescarModificar()
    Txt_CodigoProd.Clear()
    TxtNombreProd.Clear()
    Txt_DescripcionProd.Clear()
```

```
        TxtMarcaProd.Clear()
        Txt_CategoriaProd.Clear()


        Txt_EstadoProd.Clear()
        ComboBox_nombreusuarioP.DataSource = Nothing
        Btn_AgregarProd.Enabled = True
        Btn_ModificarProd.Enabled = True
        Btn_eliminarPord.Enabled = True
        Txt_CodigoProd.Enabled = True
        Btn_AceptarProd.Enabled = False
        ComboBox_ProveedorProd.Text = ""
    End Sub
    Private Sub Btn_ModificarProd_Click(sender As Object, e As EventArgs) Handles
Btn_ModificarProd.Click
        If List_productos.SelectedItems.Count >= 0 Then
            If Txt_CodigoProd.Text <> "" And TxtNombreProd.Text <> "" And
Txt_DescripcionProd.Text <> "" And TxtMarcaProd.Text <> "" And (Txt_CategoriaProd.Text <> ""
Or Txt_CategoriaProd.Text = "") And Txt_EstadoProd.Text <> "" And
ComboBox_ProveedorProd.Text <> "" And ComboBox_nombreusuarioP.Text <> "" And
Len(Txt_CodigoProd.Text) <= 6 And Len(TxtNombreProd.Text) <= 50 And
Len(Txt_DescripcionProd.Text) <= 70 And Len(TxtMarcaProd.Text) <= 25 And
(Len(Txt_CategoriaProd.Text) <= 20 Or Len(Txt_CategoriaProd.Text) = 0) And
Len(Txt_EstadoProd.Text) <= 1 Then
```

```
        If IsNumeric(Txt_CodigoProd.Text) And Not IsNumeric(TxtNombreProd.Text) And Not
IsNumeric(TxtMarcaProd.Text) And Not IsNumeric(Txt_EstadoProd.Text) Then

            datos.CommandText = "Update productos set nombre_producto ='" &
TxtNombreProd.Text & "' , descripcion_producto ='" & Txt_DescripcionProd.Text & "',
marca_producto = '" & TxtMarcaProd.Text & "', categoria_producto='" &
Txt_CategoriaProd.Text & "', estado ='" & Txt_EstadoProd.Text & "',
Proveedores_rut_proveedor ='" & ComboBox_ProveedorProd.Text & "' ,
Usuarios_nombre_usuario = '" & ComboBox_nombreusuarioP.Text & " ' where codigo_producto
='" & Txt_CodigoProd.Text & """

            datos.ExecuteNonQuery()
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Producto Actualizado", 0, mb.FStyle.Information, "Modificar
Producto")
            Limpiar()
            MostrarListview()
            RefrescarModificar()
        Else
            If Not IsNumeric(Txt_CodigoProd.Text) Then
                Dim msg As New mb.ShowMessagebox
```

```
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("codigo debe ser numerico", 0, mb.FStyle.Information, "Producto")
            Txt_CodigoProd.Clear()
            Txt_CodigoProd.Focus()
        End If
        If IsNumeric(TxtNombreProd.Text) Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("nombre no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
            TxtNombreProd.Clear()
            TxtNombreProd.Focus()
        End If

        If IsNumeric(TxtMarcaProd.Text) Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox(" marca no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
            TxtMarcaProd.Clear()
            TxtMarcaProd.Focus()
        End If
```

```
            If IsNumeric(Txt_EstadoProd.Text) Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Estado no debe ser numerico", 0, mb.FStyle.Information,
"Producto")
                Txt_EstadoProd.Clear()
                Txt_EstadoProd.Focus()
            End If
        End If
    Else
        If TxtNombreProd.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("nombre no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf TxtNombreProd.Text <> "" Then
            If Len(TxtNombreProd.Text) > 50 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo nombre puede tener solo hasta 50 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                    TxtNombreProd.Clear()
```

```
                TxtNombreProd.Focus()
            End If
        End If
    End If
    If Txt_DescripcionProd.Text = "" Then
        Dim msg As New mb.ShowMessagebox
        msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
        msg.ShowBox("descripcion no debe estar vacio", 0, mb.FStyle.Information,
"Producto")
    ElseIf Txt_DescripcionProd.Text <> "" Then
        If Len(Txt_DescripcionProd.Text) > 70 Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Campo descripcion puede tener solo hasta 70 digitos ", 0,
mb.FStyle.Exclamation, "productos")
            If MsgBoxResult.Ok = 1 Then
                Txt_DescripcionProd.Clear()
                Txt_DescripcionProd.Focus()
            End If
        End If
    End If
    If TxtMarcaProd.Text = "" Then
```

```
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("marca no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf TxtMarcaProd.Text <> "" Then
            If Len(TxtMarcaProd.Text) > 25 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo marca puede tener solo hasta 25 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                    TxtMarcaProd.Clear()
                    TxtMarcaProd.Focus()
                End If
            End If
        End If
        If Txt_EstadoProd.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Estado no debe estar vacio", 0, mb.FStyle.Information, "Producto")
        ElseIf Txt_EstadoProd.Text <> "" Then
            If Len(Txt_EstadoProd.Text) > 1 Then
```

```
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Campo codigo puede tener solo hasta 1 digitos ", 0,
mb.FStyle.Exclamation, "productos")
                If MsgBoxResult.Ok = 1 Then
                  Txt_EstadoProd.Clear()
                  Txt_EstadoProd.Focus()
                End If
              End If


            End If
          End If
        End If
    End Sub
    Private Sub List_productos_Click(sender As Object, e As EventArgs) Handles
List_productos.Click
        If List_productos.SelectedItems.Count >= 0 Then
          MostrarCombox()
          Txt_CodigoProd.Text = List_productos.SelectedItems(0).SubItems(0).Text
          TxtNombreProd.Text = List_productos.SelectedItems(0).SubItems(1).Text
          Txt_DescripcionProd.Text = List_productos.SelectedItems(0).SubItems(2).Text
          TxtMarcaProd.Text = List_productos.SelectedItems(0).SubItems(3).Text
```

```
        Txt_CategoriaProd.Text = List_productos.SelectedItems(0).SubItems(4).Text

        Txt_EstadoProd.Text = List_productos.SelectedItems(0).SubItems(6).Text

        ComboBox_ProveedorProd.Text = List_productos.SelectedItems(0).SubItems(7).Text

        ComboBox_nombreusuarioP.Text = List_productos.SelectedItems(0).SubItems(8).Text

        Txt_CodigoProd.Enabled = True

        TxtNombreProd.Enabled = True

        Txt_DescripcionProd.Enabled = True

        TxtMarcaProd.Enabled = True

        Txt_CategoriaProd.Enabled = True

        Txt_EstadoProd.Enabled = True

        ComboBox_nombreusuarioP.Enabled = True

        ComboBox_ProveedorProd.Enabled = True

        Btn_AceptarProd.Enabled = False

        Btn_cancelarProd.Enabled = False


    End If

  End Sub

  Private Sub Btn_eliminarPord_Click(sender As Object, e As EventArgs) Handles
Btn_eliminarPord.Click

    ShowException()

  End Sub
```

```
Public Sub ShowException()
    If List_productos.SelectedItems.Count >= 0 Then
        Try
            If Conex.State = ConnectionState.Open Then
                Conex.Close()
            End If
            Conex.Open()
            datos.CommandText = "Delete from productos where codigo_producto='" &
Txt_CodigoProd.Text & "' And Proveedores_rut_proveedor='" &
ComboBox_ProveedorProd.Text & "'"
            datos.ExecuteNonQuery()
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Datos Eliminado", 0, mb.FStyle.Information, "Eliminar producto ")
            Limpiar()
            MostrarListview()
            Txt_CodigoProd.Clear()
            TxtNombreProd.Clear()
            Txt_DescripcionProd.Clear()
            TxtMarcaProd.Clear()
            Txt_CategoriaProd.Clear()
            Txt_EstadoProd.Clear()
```

```
            Txt_CodigoProd.Enabled = True


            ComboBox_nombreusuarioP.Text = ""
            ComboBox_ProveedorProd.Text = ""
        Catch exception As MySqlException
            MessageBox.Show("NO SE PUEDE ELIMINAR AL ESTAR REFERENCIADO , ELIMINE LOS
CAMPOS DONDE ESTE CAMPO ESTÉ ASOCIADO  " & exception.Message)
            Limpiar()
            MostrarListview()
            Txt_CodigoProd.Clear()
            TxtNombreProd.Clear()
            Txt_DescripcionProd.Clear()
            Txt_CodigoProd.Enabled = True
            TxtMarcaProd.Clear()
            Txt_CategoriaProd.Clear()
            Txt_EstadoProd.Clear()
            ComboBox_nombreusuarioP.Text = ""
            ComboBox_ProveedorProd.Text = ""
        End Try
    End If
End Sub
```

```
    Private Sub Btn_cancelarProd_Click(sender As Object, e As EventArgs) Handles
Btn_cancelarProd.Click
        Txt_CodigoProd.Clear()
        TxtNombreProd.Clear()
        Txt_DescripcionProd.Clear()
        TxtMarcaProd.Clear()
        Txt_CategoriaProd.Clear()
        Txt_EstadoProd.Clear()
        ComboBox_nombreusuarioP.DataSource = Nothing
        Btn_ModificarProd.Enabled = True
        Btn_eliminarPord.Enabled = True
        Btn_AceptarProd.Enabled = False
        Btn_cancelarProd.Enabled = False
        Button_informeProd.Enabled = True
        ComboBox_ProveedorProd.Text = ""
        Conex.Close()
    End Sub
    Private Sub Btn_BuscarProd_Click(sender As Object, e As EventArgs) Handles
Btn_BuscarProd.Click
        Dim Li As New ListViewItem
        Limpiar()
```

```
Try
    Conex.Open()
    Dim consultaBuscar As String = String.Empty
    consultaBuscar = "SELECT * FROM productos where codigo_producto LIKE '%" &
Txt_BuscarProd.Text & "%' or nombre_producto LIKE '%" & Txt_BuscarProd.Text & "%' or
descripcion_producto LIKE '%" & Txt_BuscarProd.Text & "%' or marca_producto  LIKE '%" &
Txt_BuscarProd.Text & "%' or categoria_producto Like '%" & Txt_BuscarProd.Text & "%' or stock
LIKE '%" & Txt_BuscarProd.Text & "%' or estado  LIKE '%" & Txt_BuscarProd.Text & "%' or
Proveedores_rut_proveedor Like '%" & Txt_BuscarProd.Text & "%' or Usuarios_nombre_usuario
LIKE '%" & Txt_BuscarProd.Text & "%'"
    datos.CommandText = consultaBuscar
    datos.Connection = Conex
    mys = datos.ExecuteReader
    While mys.Read
        Li = List_productos.Items.Add(mys.Item("codigo_producto").ToString)
        Li.SubItems.Add(mys.Item("nombre_producto").ToString)
        Li.SubItems.Add(mys.Item("descripcion_producto").ToString)
        Li.SubItems.Add(mys.Item("marca_producto").ToString)
        Li.SubItems.Add(mys.Item("categoria_producto").ToString)
        Li.SubItems.Add(mys.Item("stock").ToString)
        Li.SubItems.Add(mys.Item("estado").ToString)
        Li.SubItems.Add(mys.Item("Proveedores_rut_proveedor").ToString)
        Li.SubItems.Add(mys.Item("Usuarios_nombre_usuario").ToString)
```

```
            End While
            Conex.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
    Private Sub Txt_BuscarProd_TextChanged(sender As Object, e As EventArgs) Handles
Txt_BuscarProd.TextChanged
        If Txt_BuscarProd.Text = "" Then
            Conex.Open()
            Limpiar()
            MostrarListview()
        End If
    End Sub
    Private Sub Button_informeProd_Click(sender As Object, e As EventArgs) Handles
Button_informeProd.Click
        ReporteProductos.Show()
    End Sub
End Class
```

- Mantenedor de retiro de productos

```
Imports System.Data.Sql

Imports System.Text

Imports MySql.Data.MySqlClient

Imports System.Security.Cryptography

Public Class RetirodeProductos

   Dim Conex As New MySqlConnection

   Dim mysql As New MySqlDataAdapter

   Dim mysql1 As New MySqlDataAdapter

   Dim mys As MySqlDataReader

   Dim Mysqll As DataSet

   Dim Mysqll1 As DataSet

   Dim datos As New MySqlCommand

   Dim mostrar As New DataTable

   Dim guardarstock1 As Integer

   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

      Me.Close()

      Form2.Show()

   End Sub

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

```
        Conex.ConnectionString = "server=localhost;user=root;password=
;database=taller_mecanico"
        If Conex.State = ConnectionState.Open Then
            Conex.Close()
        End If
        Conex.Open()
        MostrarListview()
        Btn_aceptarRet.Enabled = False
        Btn_cancelarRet.Enabled = False
        DateTimePicker_RetiroPr.MaxDate = Today
    End Sub
    Public Sub MostrarListview()
        Dim Li As New ListViewItem
        Dim Fecha As Date
        Try
            Dim consulta As String = String.Empty
            consulta = "SELECT * FROM retiro_productos"
            datos.CommandText = consulta
            datos.Connection = Conex
            mys = datos.ExecuteReader
            While mys.Read
```

```
            Fecha = mys.Item("fecha_retiro_producto").ToString
            Li = List_Retiro.Items.Add(mys.Item("codigo_retiro_producto").ToString)
            Li.SubItems.Add(mys.Item("descripcion_retiro_producto").ToString)
            Li.SubItems.Add(mys.Item("cantidad_retiro_producto").ToString)
            Li.SubItems.Add(Fecha)
            Li.SubItems.Add(mys.Item("Productos_codigo_producto").ToString)
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Public Sub MostrarCombox()
    Try
        Conex.Open()
        Dim consulta As String = String.Empty
        Dim i As Integer
        consulta = "SELECT * FROM  productos"
        mysql = New MySqlDataAdapter(consulta, Conex)
        Mysqll = New DataSet
        Mysqll.Tables.Add("productos")
```

```
        mysql.Fill(Mysqll.Tables("productos"))
        Dim codigo As Integer
        Dim nombre As String
        If Mysqll.Tables(0).Rows.Count <> 0 Then
            For i = 0 To Mysqll.Tables(0).Rows.Count - 1
                codigo = Mysqll.Tables(0).Rows(i).Item("codigo_producto")
                nombre = Mysqll.Tables(0).Rows(i).Item("nombre_producto")
                ComboBox_CodigoProdRet.Items.Add(codigo & " - " & nombre)
            Next
        End If
    Catch ex As Exception
    End Try
End Sub
Public Sub Limpiar()
    For i = 0 To List_Retiro.Items.Count - 1
        List_Retiro.Items.Clear()
    Next
End Sub
Public Sub RefrescarAgregar()
    Txt_CodigoRet.Clear()
    Txt_CantidadRet.Clear()
```

```
        Txt_DescripcionRet.Clear()
        Btn_agregarRet.Enabled = True
        Btn_cancelarRet.Enabled = False
        Btn_ModificarRet.Enabled = True
        Btn_EliminarRet.Enabled = True
        Btn_aceptarRet.Enabled = False
        Btn_cancelarRet.Enabled = False
        ComboBox_CodigoProdRet.Text = ""
    End Sub
    Private Sub Btn_agregarRet_Click(sender As Object, e As EventArgs) Handles
Btn_agregarRet.Click
        MostrarCombox()
        Btn_agregarRet.Enabled = True
        Btn_ModificarRet.Enabled = False
        Btn_EliminarRet.Enabled = False
        Btn_agregarRet.Enabled = True
        Btn_aceptarRet.Enabled = True
        Btn_cancelarRet.Enabled = True
    End Sub
    Private Sub Btn_aceptarRet_Click(sender As Object, e As EventArgs) Handles
Btn_aceptarRet.Click
```

```
    If Txt_CodigoRet.Text <> "" And Txt_DescripcionRet.Text <> "" And
DateTimePicker_RetiroPr.Text <> "" And ComboBox_CodigoProdRet.Text <> "" And
Len(Txt_CodigoRet.Text) <= 6 And Len(Txt_DescripcionRet.Text) <= 50 And
Len(Txt_CantidadRet.Text) <= 4 Then
        Dim consultaCodigoRet As String = String.Empty
        Dim cadenaComboArray As String
        cadenaComboArray = Mid(ComboBox_CodigoProdRet.Text, 1, 1)
        consultaCodigoRet = "Select * from retiro_productos where codigo_retiro_producto = '"
& Txt_CodigoRet.Text & "'"
        datos.CommandText = consultaCodigoRet
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim cant As Integer = 0
        While mys.Read
           cant = cant + 1
        End While
        If cant = 1 Then
           Dim msg As New mb.ShowMessagebox
           msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
           msg.ShowBox(" codigo retiro ya existe", 0, mb.FStyle.Exclamation, "producto")
           mys.Close()
        Else
```

```
mys.Close()
Dim mensajeActualizarStock As String = String.Empty
mensajeActualizarStock = "Select * from productos where codigo_producto = '" &
cadenaComboArray & "'"
datos.CommandText = mensajeActualizarStock
datos.Connection = Conex
mys = datos.ExecuteReader
Dim cantidadd As Integer = 0
Dim stockProductoo As String
Dim EstadoProducto As String
Dim guardarStock As String
Dim restarStock As Integer
Dim nombreproducto As String
While mys.Read
   cantidadd = cantidadd + 1
   stockProductoo = mys.Item("stock").ToString
   EstadoProducto = mys.Item("estado").ToString
   nombreproducto = mys.Item("nombre_producto").ToString
End While
If cantidadd = 1 Then
   guardarStock = stockProductoo
```

```
If stockProductoo = 0 And EstadoProducto = "N" Then
    Dim msg As New mb.ShowMessagebox
    msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
    msg.ShowBox("No se puede descontar producto no está vigente ", 0,
mb.FStyle.Exclamation, "producto")
    mys.Close()
End If
If stockProductoo <> 0 And EstadoProducto = "V" Then
    mys.Close()
    If Txt_CantidadRet.Text <= stockProductoo Then
        restarStock = guardarStock - Txt_CantidadRet.Text
        datos.CommandText = "insert into retiro_productos(codigo_retiro_producto ,
descripcion_retiro_producto , cantidad_retiro_producto , fecha_retiro_producto ,
Productos_codigo_producto) values('" & Txt_CodigoRet.Text & "','" & Txt_DescripcionRet.Text
& "','" & Txt_CantidadRet.Text & "','" & DateTimePicker_RetiroPr.Text & "','" &
cadenaComboArray & "')"
        datos.ExecuteNonQuery()
        datos.CommandText = "Update productos set stock = '" & restarStock & " '
where codigo_producto ='" & cadenaComboArray & "'"
        datos.ExecuteNonQuery()
        MsgBox("El producto :" & nombreproducto & " " & "Con el descuento realizado
queda un stock de  : " & restarStock)
        Dim msg As New mb.ShowMessagebox
```

```
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("Dato guardado", 0, mb.FStyle.Exclamation, "producto")
                Dim mensajeActualizacion As String = String.Empty
                If restarStock = "0" Then
                    mensajeActualizacion = "Select estado from productos where
codigo_producto = '" & cadenaComboArray & "'"
                    datos.CommandText = mensajeActualizacion
                    datos.Connection = Conex
                    mys = datos.ExecuteReader
                    Dim cantidadActualizacion As Integer = 0
                    While mys.Read
                        cantidadActualizacion = cantidadActualizacion + 1
                    End While
                    If cantidadActualizacion = 1 Then
                        Dim actualizacionEstadoo As String
                        mys.Close()
                        actualizacionEstadoo = "N"
                        datos.CommandText = "Update productos set estado ='" &
actualizacionEstadoo & " ' where codigo_producto ='" & cadenaComboArray & "'"
                        datos.ExecuteNonQuery()
                    End If
                End If
```

```
                mys.Close()
                Limpiar()
                MostrarListview()
                RefrescarAgregar()
              Else
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("cantidad debe ser menor", 0, mb.FStyle.Exclamation,
"producto")
              End If
            End If
          End If
        End If
      Else
        If Txt_CodigoRet.Text = "" Then
          Dim msg As New mb.ShowMessagebox
          msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
          msg.ShowBox("codigo  no puede estar vacio ", 0, mb.FStyle.Exclamation, "producto")
        ElseIf Txt_CodigoRet.Text <> "" And Not IsNumeric(Txt_CodigoRet.Text) Then
          Dim msg As New mb.ShowMessagebox
          msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
```

```
            msg.ShowBox("codigo debe ser numerico ", 0, mb.FStyle.Exclamation, "producto")
            Txt_CodigoRet.Clear()
            Txt_CodigoRet.Focus()


        ElseIf Txt_CodigoRet.Text <> "" And IsNumeric(Txt_CodigoRet.Text) Then
            If Len(Txt_CodigoRet.Text) > 6 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("codigo debe tener hasta 6 digitos ", 0, mb.FStyle.Exclamation,
"producto")
                If MsgBoxResult.Ok = 1 Then
                    Txt_CodigoRet.Clear()
                    Txt_CodigoRet.Focus()
                End If
            End If
        End If
        If Txt_DescripcionRet.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("descripcion  no puede estar vacio ", 0, mb.FStyle.Exclamation,
"producto")
```

```
        ElseIf Txt_DescripcionRet.Text <> "" Then
            If Len(Txt_DescripcionRet.Text) > 50 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("DEscripcion debe tener 50 digitos ", 0, mb.FStyle.Exclamation,
"producto")
                If MsgBoxResult.Ok = 1 Then
                    Txt_DescripcionRet.Clear()
                    Txt_DescripcionRet.Focus()



                End If
            End If


        End If
        If Txt_CantidadRet.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Campo cantidad  no puede estar vacio ", 0, mb.FStyle.Exclamation,
"producto")
        ElseIf Txt_CantidadRet.Text <> "" And Not IsNumeric(Txt_CantidadRet.Text) Then
            Dim msg As New mb.ShowMessagebox
```

```
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("cantidad debe ser numerico ", 0, mb.FStyle.Exclamation, "producto")
            Txt_CantidadRet.Clear()
            Txt_CantidadRet.Focus()


        ElseIf Txt_CantidadRet.Text <> "" And IsNumeric(Txt_CantidadRet.Text) Then
            If Len(Txt_CantidadRet.Text) > 4 Then
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("cantidad debe tener 6 digitos ", 0, mb.FStyle.Exclamation,
"producto")
                If MsgBoxResult.Ok = 1 Then
                    Txt_CantidadRet.Clear()
                    Txt_CantidadRet.Focus()
                End If
            End If
        End If
    End If
End Sub
Public Sub RefrescarModificar()
    Txt_CodigoRet.Clear()
```

```
        Btn_agregarRet.Enabled = True
        Btn_ModificarRet.Enabled = True
        Btn_EliminarRet.Enabled = True
        Btn_aceptarRet.Enabled = False
        Txt_DescripcionRet.Clear()
        Txt_CantidadRet.Clear()
        ComboBox_CodigoProdRet.Text = ""
    End Sub
    Private Sub Btn_ModificarRet_Click(sender As Object, e As EventArgs) Handles
Btn_ModificarRet.Click
        Dim restaurar As Integer
        Dim restaurarAnterior As Integer
        Dim guardarCodigoProd As Integer
        Dim cadenaComboArray As String
        cadenaComboArray = Mid(ComboBox_CodigoProdRet.Text, 1, 1)
        If List_Retiro.SelectedItems.Count >= 0 Then
            guardarstock1 = List_Retiro.SelectedItems(0).SubItems(2).Text
            guardarCodigoProd = List_Retiro.SelectedItems(0).SubItems(4).Text
            If Txt_CodigoRet.Text <> "" And Txt_DescripcionRet.Text <> "" And
DateTimePicker_RetiroPr.Text <> "" And ComboBox_CodigoProdRet.Text <> "" And
Len(Txt_CodigoRet.Text) <= 6 And Len(Txt_DescripcionRet.Text) <= 50 And
Len(Txt_CantidadRet.Text) <= 4 Then
```

```
If guardarCodigoProd <> cadenaComboArray Then
    Dim mensajeAnt As String = String.Empty
    mensajeAnt = "Select * from productos where codigo_producto = '" &
guardarCodigoProd & "'"
    datos.CommandText = mensajeAnt
    datos.Connection = Conex
    mys = datos.ExecuteReader
    Dim cantt As Integer = 0
    Dim stockProdd As Integer
    Dim restarStock As Integer
    While mys.Read
        cantt = cantt + 1
        stockProdd = mys.Item("stock").ToString
    End While
    If cantt = 1 Then
        restaurar = stockProdd + guardarstock1
        mys.Close()
        datos.CommandText = "Update productos set stock='" & restaurar & "' where
codigo_producto ='" & guardarCodigoProd & "'"
        datos.ExecuteNonQuery()
        Dim mensajeAnte As String = String.Empty
```

```
            mensajeAnte = "Select * from productos where codigo_producto = '" &
cadenaComboArray & "'"
            datos.CommandText = mensajeAnte
            datos.Connection = Conex
            mys = datos.ExecuteReader
            Dim cantt1 As Integer = 0
            Dim stockpr As Integer
            While mys.Read
              cantt1 = cantt1 + 1
              stockpr = mys.Item("stock").ToString
            End While
            If cantt1 = 1 Then
              mys.Close()
              restarStock = stockpr - Txt_CantidadRet.Text
              If stockpr >= Txt_CantidadRet.Text Then
                datos.CommandText = "Update productos set stock='" & restarStock & "'
where codigo_producto ='" & cadenaComboArray & "'"
                datos.ExecuteNonQuery()
                datos.CommandText = "Update retiro_productos set
descripcion_retiro_producto ='" & Txt_DescripcionRet.Text & "' , cantidad_retiro_producto ='"
& Txt_CantidadRet.Text & " ' , fecha_retiro_producto='" & DateTimePicker_RetiroPr.Text & "' ,
Productos_codigo_producto='" & cadenaComboArray & "' where codigo_retiro_producto ='" &
Txt_CodigoRet.Text & "'"
```

```
datos.ExecuteNonQuery()
Limpiar()
MostrarListview()
Dim msg As New mb.ShowMessagebox
msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
msg.ShowBox("Actualizado ", 0, mb.FStyle.Exclamation, "producto")
RefrescarModificar()
Dim mensajeActualizarD As String = String.Empty
If restarStock = "0" Then
    mensajeActualizarD = "Select estado from productos where
codigo_producto = '" & cadenaComboArray & "'"
    datos.CommandText = mensajeActualizarD
    datos.Connection = Conex
    mys = datos.ExecuteReader
    Dim cantidadActualizacion As Integer = 0
    While mys.Read
        cantidadActualizacion = cantidadActualizacion + 1
    End While

    If cantidadActualizacion = 1 Then
        Dim actua As String
```

```
                    mys.Close()
                    actua = "N"
                    datos.CommandText = "Update productos set estado ='" & actua & " '
where codigo_producto ='" & cadenaComboArray & "'"
                    datos.ExecuteNonQuery()
                  End If
                End If
              Else
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("no se puede descontar esta cantidad  ", 0,
mb.FStyle.Exclamation, "producto")
              End If
            End If
          End If
        Else
          Dim restarStockk1 As Integer
          Dim restStockk As Integer
          Dim consultacodigo As String = String.Empty
          consultacodigo = "Select * from productos where codigo_producto = '" &
cadenaComboArray & "'"
          datos.CommandText = consultacodigo
```

```
datos.Connection = Conex
mys = datos.ExecuteReader
Dim canttidad As Integer = 0
Dim stockproduct As Integer
While mys.Read
   canttidad = canttidad + 1
   stockproduct = mys.Item("stock").ToString
End While
If canttidad = 1 Then
   mys.Close()
   restarStockk1 = stockproduct + guardarstock1
   restStockk = restarStockk1 - Txt_CantidadRet.Text
   If stockproduct >= Txt_CantidadRet.Text Then
      datos.CommandText = "Update productos set stock='" & restStockk & "' where
codigo_producto ='" & cadenaComboArray & "'"
      datos.ExecuteNonQuery()
      datos.CommandText = "Update retiro_productos set
descripcion_retiro_producto ='" & Txt_DescripcionRet.Text & "' , cantidad_retiro_producto ='"
& Txt_CantidadRet.Text & " ' , fecha_retiro_producto='" & DateTimePicker_RetiroPr.Text & "' ,
Productos_codigo_producto='" & cadenaComboArray & "' where codigo_retiro_producto ='" &
Txt_CodigoRet.Text & "'"
      datos.ExecuteNonQuery()
      Limpiar()
```

```
MostrarListview()
Dim msg As New mb.ShowMessagebox
msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
msg.ShowBox("actualizado ", 0, mb.FStyle.Exclamation, "producto")
RefrescarModificar()
Dim mensajeActualizarD As String = String.Empty
If restStockk = "0" Then


    mensajeActualizarD = "Select estado from productos where codigo_producto
= '" & cadenaComboArray & "'"
    datos.CommandText = mensajeActualizarD
    datos.Connection = Conex
    mys = datos.ExecuteReader
    Dim cantidadActualizacion As Integer = 0
    While mys.Read
        cantidadActualizacion = cantidadActualizacion + 1
    End While
    If cantidadActualizacion = 1 Then
        Dim actualizacionEstado As String
        mys.Close()
        actualizacionEstado = "N"
```

```
                    datos.CommandText = "Update productos set estado ='" &
actualizacionEstado & " ' where codigo_producto ='" & cadenaComboArray & "'"
                    datos.ExecuteNonQuery()
                  End If
                End If
              Else
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("no se puede descontar esta cantidad ", 0, mb.FStyle.Exclamation,
"producto")
              End If
            End If
          End If
        Else
          If Txt_DescripcionRet.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("descripcion  no puede estar vacio ", 0, mb.FStyle.Exclamation,
"producto")
          ElseIf Txt_DescripcionRet.Text <> "" Then
            If Len(Txt_DescripcionRet.Text) > 50 Then
              Dim msg As New mb.ShowMessagebox
```

```
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("DEscripcion debe tener 50 digitos ", 0, mb.FStyle.Exclamation,
"producto")
                If MsgBoxResult.Ok = 1 Then
                    Txt_DescripcionRet.Clear()
                    Txt_DescripcionRet.Focus()
                End If
            End If
        End If
        If Txt_CantidadRet.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Campo cantidad  no puede estar vacio ", 0, mb.FStyle.Exclamation,
"producto")
        ElseIf Txt_CantidadRet.Text <> "" And Not IsNumeric(Txt_CantidadRet.Text) Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("cantidad debe ser numérico ", 0, mb.FStyle.Exclamation, "producto")
            Txt_CantidadRet.Clear()
            Txt_CantidadRet.Focus()

        ElseIf Txt_CantidadRet.Text <> "" And IsNumeric(Txt_CantidadRet.Text) Then
```

```
                If Len(Txt_CantidadRet.Text) > 4 Then
                    Dim msg As New mb.ShowMessagebox
                    msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                    msg.ShowBox("cantidad debe tener 6 digitos ", 0, mb.FStyle.Exclamation,
"producto")
                    If MsgBoxResult.Ok = 1 Then
                        Txt_CantidadRet.Clear()
                        Txt_CantidadRet.Focus()
                    End If
                End If
            End If
        End If
    End If
End Sub
Private Sub List_Retiro_Click(sender As Object, e As EventArgs) Handles List_Retiro.Click
    If List_Retiro.SelectedItems.Count >= 0 Then
        MostrarCombox()
        Txt_CodigoRet.Text = List_Retiro.SelectedItems(0).SubItems(0).Text
        Txt_DescripcionRet.Text = List_Retiro.SelectedItems(0).SubItems(1).Text
        Txt_CantidadRet.Text = List_Retiro.SelectedItems(0).SubItems(2).Text
        DateTimePicker_RetiroPr.Text = List_Retiro.SelectedItems(0).SubItems(3).Text
```

```
        ComboBox_CodigoProdRet.Text = List_Retiro.SelectedItems(0).SubItems(4).Text
        Txt_CodigoRet.Enabled = False
    End If
End Sub

Private Sub Btn_EliminarRet_Click(sender As Object, e As EventArgs) Handles
Btn_EliminarRet.Click
    If List_Retiro.SelectedItems.Count >= 0 Then
        Dim guardarCodigoEliminado As String
        guardarCodigoEliminado = List_Retiro.SelectedItems(0).SubItems(4).Text
        Dim mensajeElimar As String = String.Empty
        Dim guardarStockElim As Integer
        Dim sumarStockElim As Integer
        guardarStockElim = List_Retiro.SelectedItems(0).SubItems(2).Text
        Dim cadenaComboArray As String
        cadenaComboArray = Mid(ComboBox_CodigoProdRet.Text, 1, 1)
        mensajeElimar = "Select * from productos where codigo_producto = '" &
cadenaComboArray & "'"
        datos.CommandText = mensajeElimar
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim canttidadElim As Integer = 0
```

```
        Dim stockproductElim As Integer
        While mys.Read
            canttidadElim = canttidadElim + 1
            stockproductElim = mys.Item("stock").ToString
        End While
        If canttidadElim = 1 Then
            mys.Close()
            sumarStockElim = stockproductElim + guardarStockElim
            datos.CommandText = "Update productos set stock='" & sumarStockElim & "' where
codigo_producto ='" & cadenaComboArray & "'"
            datos.ExecuteNonQuery()
            datos.CommandText = "Delete from retiro_productos where codigo_retiro_producto
='" & Txt_CodigoRet.Text & "'"
            datos.ExecuteNonQuery()
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Datos Eliminado", 0, mb.FStyle.Information, "Eliminar retiro ")
        End If
        Limpiar()
        MostrarListview()
        Txt_CodigoRet.Clear()
        Txt_DescripcionRet.Clear()
```

```
        ComboBox_CodigoProdRet.Text = ""
        Txt_CantidadRet.Clear()
        Txt_CodigoRet.Enabled = True
    End If
  End Sub
  Private Sub Btn_cancelarRet_Click(sender As Object, e As EventArgs) Handles
Btn_cancelarRet.Click
    Txt_CodigoRet.Clear()
    Txt_DescripcionRet.Clear()
    Txt_CantidadRet.Clear()
    ComboBox_CodigoProdRet.DataSource = Nothing
    Btn_aceptarRet.Enabled = False
    Btn_cancelarRet.Enabled = False
    Btn_agregarRet.Enabled = True
    Btn_ModificarRet.Enabled = True
    Btn_EliminarRet.Enabled = True
    Conex.Close()
  End Sub
  Private Sub Btn_BuscarRet_Click(sender As Object, e As EventArgs) Handles
Btn_BuscarRet.Click
    Dim Li As New ListViewItem
    Limpiar()
```

```
Try
    Conex.Open()
    Dim consultaBuscar As String = String.Empty
    consultaBuscar = "SELECT * FROM retiro_productos where codigo_retiro_producto LIKE
'%" & Txt_BuscarRetiro.Text & "%' or descripcion_retiro_producto LIKE '%" &
Txt_BuscarRetiro.Text & "%' or descripcion_retiro_producto LIKE '%" & Txt_BuscarRetiro.Text &
"%' or cantidad_retiro_producto LIKE '%" & Txt_BuscarRetiro.Text & "%'or
fecha_retiro_producto LIKE '%" & Txt_BuscarRetiro.Text & "%' or Productos_codigo_producto
LIKE '%" & Txt_BuscarRetiro.Text & "%'"
    datos.CommandText = consultaBuscar
    datos.Connection = Conex
    mys = datos.ExecuteReader
    While mys.Read
        Li = List_Retiro.Items.Add(mys.Item("codigo_retiro_producto").ToString)
        Li.SubItems.Add(mys.Item("descripcion_retiro_producto").ToString)
        Li.SubItems.Add(mys.Item("cantidad_retiro_producto").ToString)
        Li.SubItems.Add(mys.Item("fecha_retiro_producto").ToString)
        Li.SubItems.Add(mys.Item("Productos_codigo_producto").ToString)
    End While
    Conex.Close()
Catch ex As Exception
    MsgBox(ex.Message)
End Try
```

```
    End Sub


    Private Sub Txt_BuscarRetiro_TextChanged(sender As Object, e As EventArgs) Handles
Txt_BuscarRetiro.TextChanged
        If Txt_BuscarRetiro.Text = "" Then
            Conex.Open()
            Limpiar()
            MostrarListview()
        End If
    End Sub
End Class
```

- **<u>Mantenedor de detalle de compras de productos (entradas)</u>**

```
Imports System.Data.Sql
Imports System.Text
Imports MySql.Data.MySqlClient
Imports System.Security.Cryptography
Public Class DetalledeCompra
    Dim Conex As New MySqlConnection
    Dim mysql As New MySqlDataAdapter
    Dim mysql1 As New MySqlDataAdapter
    Dim mys As MySqlDataReader
```

```
    Dim Mysqll As DataSet
    Dim Mysqll1 As DataSet
    Dim datos As New MySqlCommand
    Dim mostrar As New DataTable
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        Form2.Show()
    End Sub
    Private Sub DetalledeCompra_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Conex.ConnectionString = "server=localhost;user=root;password=
;database=taller_mecanico"
        If Conex.State = ConnectionState.Open Then
            Conex.Close()
        End If
        Conex.Open()
        MostrarListview()
        Btn_aceptarDet.Enabled = False
        Btn_cancelarDet.Enabled = False
    End Sub
    Public Sub MostrarListview()
        Dim Li As New ListViewItem
```

```
    Try
        Dim consulta As String = String.Empty
        consulta = "SELECT * FROM detalle_compras"
        datos.CommandText = consulta
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim montoPag As String
        While mys.Read
            Li=List_Detalle.Items.Add(mys.Item("Compra_Productos_codigo_factura").ToString)
            Li.SubItems.Add(mys.Item("Productos_codigo_producto").ToString)
            Li.SubItems.Add(mys.Item("descripcion_compra_producto").ToString)
            Li.SubItems.Add(mys.Item("cantidad_compra").ToString)
            'Li.SubItems.Add(mys.Item("comprado_al_valor").ToString)
            montoPag = Format(CType(mys.Item("comprado_al_valor"), Decimal), "$#,##0")
            Li.SubItems.Add(montoPag)
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
```

```
Public Sub MostrarCombox()
  Try
    Conex.Open()
    Dim consulta As String = String.Empty
    Dim rut_proveedor, nombre_proveedor As String
    Dim i As Integer
    consulta = "SELECT * FROM  compra_productos"
    mysql = New MySqlDataAdapter(consulta, Conex)
    Mysqll = New DataSet
    Mysqll.Tables.Add("compra_productos")
    mysql.Fill(Mysqll.Tables("compra_productos"))
    ComboBox_CodigoFacturaDet.DataSource = Mysqll.Tables("compra_productos")
    ComboBox_CodigoFacturaDet.DisplayMember = "codigo_factura"
    Dim consultas As String = String.Empty
    consultas = "SELECT * FROM  productos"
    mysql1 = New MySqlDataAdapter(consultas, Conex)
    Mysqll1 = New DataSet
    Mysqll1.Tables.Add("productos")
    Dim codigo As Integer
    Dim nombre As String
    mysql1.Fill(Mysqll1.Tables("productos"))
```

```
        If Mysqll1.Tables(0).Rows.Count <> 0 Then
            For i = 0 To Mysqll1.Tables(0).Rows.Count - 1
                codigo = Mysqll1.Tables(0).Rows(i).Item("codigo_producto")
                nombre = Mysqll1.Tables(0).Rows(i).Item("nombre_producto")
                ComboBox_codigoProduDEt.Items.Add(codigo & " - " & nombre)
            Next
        End If
    Catch ex As Exception
    End Try
End Sub
Public Sub Limpiar()
    For i = 0 To List_Detalle.Items.Count - 1
        List_Detalle.Items.Clear()
    Next
End Sub
Public Sub RefrescarAgregar()
    TxtCantidadDet.Clear()
    Txt_CompradoDet.Clear()
    ComboBox_CodigoFacturaDet.DataSource = Nothing
    ComboBox_codigoProduDEt.DataSource = Nothing
    Btn_ModificarDet.Enabled = True
```

```
        BtnEliminarDet.Enabled = True
        Btn_aceptarDet.Enabled = False
        Btn_cancelarDet.Enabled = False
        ComboBox_codigoProduDEt.Text = ""
    End Sub
    Private Sub Btn_agregarDet_Click(sender As Object, e As EventArgs) Handles
Btn_agregarDet.Click
        MostrarCombox()
        Btn_agregarDet.Enabled = True
        Btn_ModificarDet.Enabled = False
        BtnEliminarDet.Enabled = False
        Btn_aceptarDet.Enabled = True
        Btn_cancelarDet.Enabled = True
    End Sub
    Private Sub Btn_aceptarDet_Click(sender As Object, e As EventArgs) Handles
Btn_aceptarDet.Click
        Dim monto1 As String
        Dim montoFinal1 As Integer
        monto1 = Txt_CompradoDet.Text.Replace("$", "")
        montoFinal1 = monto1.Replace(".", "")
        If ComboBox_CodigoFacturaDet.Text <> "" And ComboBox_codigoProduDEt.Text <> "" And
TxtCantidadDet.Text <> "" And Txt_CompradoDet.Text <> "" And (Len(TxtCantidadDet.Text) <= 4
```

```
Or montoFinal1.ToString.Length <= 7) And (Len(TxtCantidadDet.Text) <= 4 And
montoFinal1.ToString.Length <= 7) Then
        Dim consultaDetalle As String = String.Empty
        consultaDetalle = "Select descripcion_producto , estado from productos where
codigo_producto = '" & ComboBox_codigoProduDEt.Text & "'"
        datos.CommandText = consultaDetalle
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim cant As Integer = 0
        Dim Descrip As String
        Dim estado As String
        While mys.Read
           cant = cant + 1
           Descrip = mys.Item("descripcion_producto").ToString
           estado = mys.Item("estado").ToString
        End While
        If cant = 1 And estado = "V" Then
           mys.Close()
           Dim monto As String
           Dim montoFinal As Integer
           monto = Txt_CompradoDet.Text.Replace("$", "")
           montoFinal = monto.Replace(".", "")
```

```
            Dim cadenaComboArray As String

            cadenaComboArray = Mid(ComboBox_codigoProduDEt.Text, 1, 1)

            datos.CommandText = "insert into
detalle_compras(Compra_Productos_codigo_factura,Productos_codigo_producto,descripcion_c
ompra_producto,cantidad_compra,comprado_al_valor) values('" &
ComboBox_CodigoFacturaDet.Text & "','" & cadenaComboArray & "','" & Descrip & "','" &
TxtCantidadDet.Text & "','" & montoFinal & "')"

            datos.ExecuteNonQuery()

            datos.CommandText = "Update productos set stock = stock + '" & TxtCantidadDet.Text
& " ' where codigo_producto ='" & ComboBox_codigoProduDEt.Text & "'"

            datos.ExecuteNonQuery()

            Dim msg As New mb.ShowMessagebox

            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))

            msg.ShowBox("Datos guardados", 0, mb.FStyle.Information, "Agregar detalle")

            Limpiar()

            MostrarListview()

            RefrescarAgregar()

        End If

    Else

        If TxtCantidadDet.Text = "" Then

            Dim msg As New mb.ShowMessagebox

            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))

            msg.ShowBox("cantidad no debe estar vacio", 0, mb.FStyle.Information, "Detalle")
```

```
End If
If Txt_CompradoDet.Text = "" Then
   Dim msg As New mb.ShowMessagebox
   msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
   msg.ShowBox("compra  valor no debe estar vacio", 0, mb.FStyle.Information, "Detalle")
End If
If Not IsNumeric(TxtCantidadDet.Text) Then
   Dim msg As New mb.ShowMessagebox
   msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
   msg.ShowBox("cantidad debe ser numerico", 0, mb.FStyle.Information, "Detalle")
End If
If Len(TxtCantidadDet.Text) > 4 Then
   Dim msg As New mb.ShowMessagebox
   msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
   msg.ShowBox("codigo debe tener 4 digitos", 0, mb.FStyle.Information, "Detalle")
   If MsgBoxResult.Ok = 1 Then
      TxtCantidadDet.Clear()
      TxtCantidadDet.Focus()
   End If
End If
If Len(Txt_CompradoDet.Text) > 8 Then
```

```
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("comprado debe tener 7 digitos", 0, mb.FStyle.Information, "Detalle")
            If MsgBoxResult.Ok = 1 Then
                Txt_CompradoDet.Clear()
                Txt_CompradoDet.Focus()
            End If
        End If
    End If
End Sub
Public Sub RefrescarModificar()
    Btn_agregarDet.Enabled = True
    Btn_ModificarDet.Enabled = True
    BtnEliminarDet.Enabled = True
    Btn_aceptarDet.Enabled = False
    ComboBox_CodigoFacturaDet.Enabled = True
    ComboBox_codigoProduDEt.Text = ""
    TxtCantidadDet.Clear()
    Txt_CompradoDet.Clear()
    ComboBox_CodigoFacturaDet.DataSource = Nothing
End Sub
```

```
Private Sub Btn_ModificarDet_Click(sender As Object, e As EventArgs) Handles
Btn_ModificarDet.Click
    Dim monto1 As String
    Dim montoFinal1 As Integer
    monto1 = Txt_CompradoDet.Text.Replace("$", "")
    montoFinal1 = monto1.Replace(".", "")
    If List_Detalle.SelectedItems.Count >= 0 Then
        If ComboBox_CodigoFacturaDet.Text <> "" And ComboBox_codigoProduDEt.Text <> ""
And TxtCantidadDet.Text <> "" And Txt_CompradoDet.Text <> "" And (Len(TxtCantidadDet.Text)
<= 4 Or montoFinal1.ToString.Length <= 7) And (Len(TxtCantidadDet.Text) <= 4 And
montoFinal1.ToString.Length <= 7) Then
            Dim codigoSeleccionado As String
            Dim ConsultarCantidad As String
            Dim sumarnuevaCanti As Integer
            Dim cadenaComboArray As String
            cadenaComboArray = Mid(ComboBox_codigoProduDEt.Text, 1, 1)
            codigoSeleccionado = List_Detalle.SelectedItems(0).SubItems(1).Text
            ConsultarCantidad = List_Detalle.SelectedItems(0).SubItems(3).Text
            Dim consultaProducto As String = String.Empty
            consultaProducto = "Select codigo_producto , estado , stock from productos where
codigo_producto = '" & codigoSeleccionado & "'"
            datos.CommandText = consultaProducto
```

```
datos.Connection = Conex
mys = datos.ExecuteReader
Dim cant As Integer = 0
Dim codigo As String
Dim estado As String
Dim stock As Integer
Dim resta As Integer
While mys.Read
    cant = cant + 1
    codigo = mys.Item("codigo_producto").ToString
    estado = mys.Item("estado").ToString
    stock = mys.Item("stock").ToString
End While
If cant = 1 And estado = "V" Then
    mys.Close()
    If codigoSeleccionado <> ComboBox_codigoProduDEt.Text And ConsultarCantidad =
TxtCantidadDet.Text Then
        If TxtCantidadDet.Text > stock Then
            resta = TxtCantidadDet.Text - stock
        ElseIf stock > TxtCantidadDet.Text Then
            resta = stock - TxtCantidadDet.Text
```

```
ElseIf stock = TxtCantidadDet.Text Then
    resta = stock - TxtCantidadDet.Text
End If
Dim sumarCantidadNuevaDIF As Integer
Dim consultaProductoDifer As String = String.Empty
consultaProductoDifer = "Select codigo_producto , estado , stock ,
descripcion_producto from productos where codigo_producto = '" & cadenaComboArray & "'"
datos.CommandText = consultaProductoDifer
datos.Connection = Conex
mys = datos.ExecuteReader
Dim cantdif As Integer = 0
Dim codigodif As String
Dim estadodif As String
Dim descripcion As String
Dim stockdif As Integer
Dim sumaDif As Integer
While mys.Read
    cantdif = cantdif + 1
    stockdif = mys.Item("stock").ToString
    descripcion = mys.Item("descripcion_producto").ToString
End While
```

```
If cantdif = 1 And estado = "V" Then
    mys.Close()
    sumaDif = TxtCantidadDet.Text + stockdif
    datos.CommandText = "Update productos set stock = '" & resta & " ' where
codigo_producto ='" & codigoSeleccionado & "'"
    datos.ExecuteNonQuery()
    datos.CommandText = "Update productos set stock = '" & sumaDif & " ' where
codigo_producto ='" & cadenaComboArray & "'"
    datos.ExecuteNonQuery()
    Dim montoCant As String
    Dim montoCant1 As Integer
    montoCant = Txt_CompradoDet.Text.Replace("$", "")
    montoCant1 = montoCant.Replace(".", "")
    datos.CommandText = "Update detalle_compras set
Productos_codigo_producto ='" & cadenaComboArray & " ' , descripcion_compra_producto = '"
& descripcion & "',cantidad_compra='" & TxtCantidadDet.Text & "', comprado_al_valor='" &
montoCant1 & "' where Compra_Productos_codigo_factura ='" &
ComboBox_CodigoFacturaDet.Text & "' and Productos_codigo_producto='" &
codigoSeleccionado & "'"
    datos.ExecuteNonQuery()
    Limpiar()
    MostrarListview()
```

```
                RefrescarModificar()
                Dim msg As New mb.ShowMessagebox
                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
                msg.ShowBox("detalle Actualizado", 0, mb.FStyle.Information, "Modificar
compra")
            End If
        End If
        If codigoSeleccionado = ComboBox_codigoProduDEt.Text And ConsultarCantidad <>
TxtCantidadDet.Text Then
            If ConsultarCantidad > stock Then
                resta = ConsultarCantidad - stock
            ElseIf stock > ConsultarCantidad Then
                resta = stock - ConsultarCantidad
            ElseIf stock = ConsultarCantidad Then
                resta = stock - ConsultarCantidad
            End If
            datos.CommandText = "Update productos set stock = '" & resta & " ' where
codigo_producto ='" & codigoSeleccionado & "'"
            datos.ExecuteNonQuery()
            sumarnuevaCanti = resta + TxtCantidadDet.Text
            Dim montoCantidadigual As String
            Dim montoCantidad1igual As Integer
```

```
montoCantidadigual = Txt_CompradoDet.Text.Replace("$", "")

montoCantidad1igual = montoCantidadigual.Replace(".", "")

datos.CommandText = "Update detalle_compras set Productos_codigo_producto
='" & cadenaComboArray & " ' , cantidad_compra='" & TxtCantidadDet.Text & "',
comprado_al_valor='" & montoCantidad1igual & "' where Compra_Productos_codigo_factura
='" & ComboBox_CodigoFacturaDet.Text & "' and Productos_codigo_producto='" &
cadenaComboArray & "'"

datos.ExecuteNonQuery()

datos.CommandText = "Update productos set stock = '" & sumarnuevaCanti & " '
where codigo_producto ='" & cadenaComboArray & "'"

datos.ExecuteNonQuery()

Limpiar()

MostrarListview()

RefrescarModificar()

Dim msg As New mb.ShowMessagebox

msg.Fonts(New Font("Arial", 20, FontStyle.Regular))

msg.ShowBox("detalle Actualizado", 0, mb.FStyle.Information, "Modificar
compra")

ElseIf codigoSeleccionado = ComboBox_codigoProduDEt.Text And ConsultarCantidad
= TxtCantidadDet.Text Then

Dim montoCantidad As String

Dim montoCantidad1 As Integer

montoCantidad = Txt_CompradoDet.Text.Replace("$", "")
```

```
                montoCantidad1 = montoCantidad.Replace(".", "")

                datos.CommandText = "Update detalle_compras set comprado_al_valor = '" &
montoCantidad1 & " '  where Compra_Productos_codigo_factura ='" &
ComboBox_CodigoFacturaDet.Text & "' and Productos_codigo_producto='" &
cadenaComboArray & "'"

                datos.ExecuteNonQuery()

                Limpiar()

                MostrarListview()

                RefrescarModificar()

                Dim msg As New mb.ShowMessagebox

                msg.Fonts(New Font("Arial", 20, FontStyle.Regular))

                msg.ShowBox("detalle Actualizado", 0, mb.FStyle.Information, "Modificar
compra")


            End If

        End If

    Else

        If TxtCantidadDet.Text = "" Then

            Dim msg As New mb.ShowMessagebox

            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))

            msg.ShowBox("cantidad no debe estar vacio", 0, mb.FStyle.Information, "Detalle")

        End If
```

```
        If Txt_CompradoDet.Text = "" Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("compra  valor no debe estar vacio", 0, mb.FStyle.Information,
"Detalle")
        End If
        If Not IsNumeric(TxtCantidadDet.Text) Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("cantidad debe ser numerico", 0, mb.FStyle.Information, "Detalle")
        End If
        If Len(TxtCantidadDet.Text) > 4 Then
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("codigo debe tener 4 digitos", 0, mb.FStyle.Information, "Detalle")
            If MsgBoxResult.Ok = 1 Then
                TxtCantidadDet.Clear()
                TxtCantidadDet.Focus()
            End If
        End If
        If Len(Txt_CompradoDet.Text) > 8 Then
```

```
            Dim msg As New mb.ShowMessagebox
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("comprado debe tener 7 digitos", 0, mb.FStyle.Information, "Detalle")
            'MsgBox("Compra debe tener solo hasta 7 digitos")
            If MsgBoxResult.Ok = 1 Then
                Txt_CompradoDet.Clear()
                Txt_CompradoDet.Focus()
            End If
        End If
    End If
  End If
End Sub
Private Sub List_Detalle_Click(sender As Object, e As EventArgs) Handles List_Detalle.Click
  If List_Detalle.SelectedItems.Count >= 0 Then
    MostrarCombox()
    ComboBox_CodigoFacturaDet.Enabled = False
    ComboBox_CodigoFacturaDet.Text = List_Detalle.SelectedItems(0).SubItems(0).Text
    ComboBox_codigoProduDEt.Text = List_Detalle.SelectedItems(0).SubItems(1).Text
    TxtCantidadDet.Text = List_Detalle.SelectedItems(0).SubItems(3).Text
    Txt_CompradoDet.Text = List_Detalle.SelectedItems(0).SubItems(4).Text
  End If
```

```
End Sub

Private Sub BtnEliminarDet_Click(sender As Object, e As EventArgs) Handles
BtnEliminarDet.Click
    ShowException()
End Sub
Public Sub ShowException()
    If List_Detalle.SelectedItems.Count >= 0 Then
        Try
            If Conex.State = ConnectionState.Open Then
                Conex.Close()
            End If
            Conex.Open()
            Dim guardarCodigoEliminadoAjust As String
            guardarCodigoEliminadoAjust = List_Detalle.SelectedItems(0).SubItems(1).Text
            Dim mensajeElimarAjust As String = String.Empty
            Dim guardarStockElimAjust As Integer
            Dim sumarStockElimAjust As Integer
            guardarStockElimAjust = List_Detalle.SelectedItems(0).SubItems(3).Text
            Dim cadenaComboArray As String
            cadenaComboArray = Mid(ComboBox_codigoProduDEt.Text, 1, 1)
```

```
        mensajeElimarAjust = "Select * from productos where codigo_producto = '" &
cadenaComboArray & "'"
        datos.CommandText = mensajeElimarAjust
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim canttidadElimAjust As Integer = 0
        Dim stockproductElimAjust As Integer
        While mys.Read
            canttidadElimAjust = canttidadElimAjust + 1
            stockproductElimAjust = mys.Item("stock").ToString
        End While
        If canttidadElimAjust = 1 Then
            mys.Close()
            sumarStockElimAjust = stockproductElimAjust + guardarStockElimAjust
            datos.CommandText = "Update productos set stock='" & sumarStockElimAjust & "'
where codigo_producto ='" & cadenaComboArray & "'"
            datos.ExecuteNonQuery()
            datos.CommandText = "Delete from detalle_compras where
Compra_Productos_codigo_factura='" & ComboBox_CodigoFacturaDet.Text & "' and
Productos_codigo_producto='" & cadenaComboArray & "'"
            datos.ExecuteNonQuery()
            Dim msg As New mb.ShowMessagebox
```

```
            msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
            msg.ShowBox("Datos Eliminado", 0, mb.FStyle.Information, "Eliminar ajuste ")
        End If
        Limpiar()
        MostrarListview()
        TxtCantidadDet.Clear()
        Txt_CompradoDet.Clear()
        ComboBox_CodigoFacturaDet.DataSource = Nothing
        ComboBox_codigoProduDEt.Text = ""
        ComboBox_CodigoFacturaDet.Enabled = True
    Catch exception As MySqlException
        MessageBox.Show("NO SE PUEDE ELIMINAR AL ESTAR REFERENCIADO , ELIMINE LOS
CAMPOS DONDE ESTE CAMPO ESTÉ ASOCIADO  " & exception.Message)
        Limpiar()
        ComboBox_CodigoFacturaDet.Enabled = True
        MostrarListview()
        TxtCantidadDet.Clear()
        Txt_CompradoDet.Clear()
        ComboBox_CodigoFacturaDet.DataSource = Nothing
        ComboBox_codigoProduDEt.Text = ""
    End Try
```

```
        End If
    End Sub
    Private Sub Btn_cancelarDet_Click(sender As Object, e As EventArgs) Handles
Btn_cancelarDet.Click
        TxtCantidadDet.Clear()
        Txt_CompradoDet.Clear()
        ComboBox_CodigoFacturaDet.DataSource = Nothing
        ComboBox_codigoProduDEt.DataSource = Nothing
        Btn_aceptarDet.Enabled = False
        Btn_cancelarDet.Enabled = False
        Btn_agregarDet.Enabled = True
        BtnEliminarDet.Enabled = True
        Btn_ModificarDet.Enabled = True
        Conex.Close()
    End Sub
    Private Sub Btn_BuscarDet_Click(sender As Object, e As EventArgs) Handles
Btn_BuscarDet.Click
        Dim Li As New ListViewItem
        Limpiar()
        Try
            Conex.Open()
            Dim consultaBuscar As String = String.Empty
```

```
        consultaBuscar = "SELECT * FROM detalle_compras where
Compra_Productos_codigo_factura LIKE '%" & Txt_BuscarDet.Text & "%' or
Productos_codigo_producto LIKE '%" & Txt_BuscarDet.Text & "%' or
descripcion_compra_producto LIKE '%" & Txt_BuscarDet.Text & "%' or cantidad_compra LIKE
'%" & Txt_BuscarDet.Text & "%' or comprado_al_valor LIKE'%" & Txt_BuscarDet.Text & "%'"
        datos.CommandText = consultaBuscar
        datos.Connection = Conex
        mys = datos.ExecuteReader
        While mys.Read
          Li=List_Detalle.Items.Add(mys.Item("Compra_Productos_codigo_factura").ToString)
          Li.SubItems.Add(mys.Item("Productos_codigo_producto").ToString)
          Li.SubItems.Add(mys.Item("descripcion_compra_producto").ToString)
          Li.SubItems.Add(mys.Item("cantidad_compra").ToString)
          Li.SubItems.Add(mys.Item("comprado_al_valor").ToString)
        End While
        Conex.Close()
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
    End Sub
    Private Sub Txt_BuscarDet_TextChanged(sender As Object, e As EventArgs) Handles
Txt_BuscarDet.TextChanged
      If Txt_BuscarDet.Text = "" Then
```

```
        Conex.Open()
        Limpiar()
        MostrarListview()
    End If
  End Sub

  Private Sub Txt_CompradoDet_LostFocus(sender As Object, e As EventArgs) Handles
Txt_CompradoDet.LostFocus
    If Txt_CompradoDet.Text <> "" Then
        Txt_CompradoDet.Text = Format(CType(Me.Txt_CompradoDet.Text, Decimal),
"#.###,##")
    End If
  End Sub
  Private Sub Txt_CompradoDet_Leave(sender As Object, e As EventArgs) Handles
Txt_CompradoDet.Leave
    Txt_CompradoDet.Text = Format(Val(Txt_CompradoDet.Text.ToString), "$#,##0")
  End Sub
End Class
```

- **Módulo boleta**

```
Imports System.Data.Sql
Imports System.Text
Imports MySql.Data.MySqlClient
Imports System.Globalization
```

```
Imports System.Threading
Imports System.Security.Cryptography
Public Class Boleta
   Dim Conex As New MySqlConnection
   Dim mysql As New MySqlDataAdapter
   Dim mysql1 As New MySqlDataAdapter
   Dim mys As MySqlDataReader
   Dim Mysqll As DataSet
   Dim Mysqll1 As DataSet
   Dim datos As New MySqlCommand
   Dim mostrar As New DataTable
   Private Sub Boleta_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Conex.ConnectionString = "server=localhost;user=root;password=
;database=taller_mecanico"
      If Conex.State = ConnectionState.Open Then
         Conex.Close()
         MostrarListview()
      End If
      Conex.Open()
      MostrarListview()
      Btn_aceptarD.Enabled = False
      Btn_cancelarD.Enabled = False
      DataGridView2.Visible = False
      DataGridView1.Visible = False
      System.Threading.Thread.CurrentThread.CurrentCulture = New
System.Globalization.CultureInfo("es-CL")
      DataGridView1.Columns.Add("Cuotas", "Cuotas Pedidas")
```

```
    DataGridView1.Columns.Add("CuotasFinal", "Quedan cuotas")
  End Sub
  Public Sub MostrarListview()
    Dim Li As New ListViewItem
    Try
      Dim consulta As String = String.Empty
      Dim fechaVencimientoDeu As Date
      Dim ValorDeudaT As String
      Dim MontoAPagarPorDeuda As String

      consulta = "SELECT * FROM deudas"
      datos.CommandText = consulta
      datos.Connection = Conex
      mys = datos.ExecuteReader
      While mys.Read
        fechaVencimientoDeu = mys.Item("fecha_vencimiento").ToString
        ValorDeudaT = Format(CType(mys.Item("deuda_valor_trabajo"), Decimal), "$#,##0")
        MontoAPagarPorDeuda = Format(CType(mys.Item("monto_a_pagar"), Decimal),
"$#,##0")
        Li = List_Deud.Items.Add(mys.Item("codigo_deuda").ToString)
        Li.SubItems.Add(ValorDeudaT)
        Li.SubItems.Add(MontoAPagarPorDeuda)
        Li.SubItems.Add(mys.Item("forma_repactacion_deuda").ToString)
        Li.SubItems.Add(mys.Item("estado_deuda").ToString)
        Li.SubItems.Add(fechaVencimientoDeu)
        Li.SubItems.Add(mys.Item("Clientes_rut_cliente").ToString)
        Li.SubItems.Add(mys.Item("Vehiculos_patente_vehiculo").ToString.ToUpper)
```

```
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Public Sub MostrarCombox()
    Try

        Dim consulta As String = String.Empty
        Dim rut_proveedor, nombre_proveedor As String
        Dim i As Integer
        consulta = "SELECT * FROM clientes"
        mysql = New MySqlDataAdapter(consulta, Conex)
        Mysqll = New DataSet
        Mysqll.Tables.Add("clientes")
        mysql.Fill(Mysqll.Tables("clientes"))
        Dim rutCli As String
        Dim nombreCli As String
        If Mysqll.Tables(0).Rows.Count <> 0 Then
            For i = 0 To Mysqll.Tables(0).Rows.Count - 1
                rutCli = Mysqll.Tables(0).Rows(i).Item("rut_cliente")
                nombreCli = Mysqll.Tables(0).Rows(i).Item("nombre_cliente")
                ComboBox_rutD.Items.Add(rutCli & " - " & nombreCli)
            Next
        End If
        Dim consultaPatente As String = String.Empty
```

```
        consultaPatente = "SELECT * FROM  vehiculos"
        mysql1 = New MySqlDataAdapter(consultaPatente, Conex)
        Mysqll1 = New DataSet
        Mysqll1.Tables.Add("vehiculos")
        Dim paten As String
        Dim rutcli2 As String
        mysql1.Fill(Mysqll1.Tables("vehiculos"))
        If Mysqll1.Tables(0).Rows.Count <> 0 Then
            For i = 0 To Mysqll1.Tables(0).Rows.Count - 1
                paten = Mysqll1.Tables(0).Rows(i).Item("patente_vehiculo")
                rutcli2 = Mysqll1.Tables(0).Rows(i).Item("Clientes_rut_cliente")
                ComboBox_PatenteVehicDeu.Items.Add(paten & " - " & rutcli2)
            Next
        End If
    Catch ex As Exception
    End Try
End Sub
Public Sub Limpiar()
    For i = 0 To List_Deud.Items.Count - 1
        List_Deud.Items.Clear()
    Next
End Sub
Public Sub RefrescarAgregar()
    Txt_CodigoD.Clear()
    Txt_EstadoD.Clear()
    Txt_MontoD.Clear()
    Txt_repctD.Clear()
```

```
    Txt_ValorD.Clear()
    ComboBox_rutD.DataSource = Nothing
    Btn_aceptarD.Enabled = False
    Btn_cancelarD.Enabled = False
    Btn_ModificarD.Enabled = True
    ComboBox_PatenteVehicDeu.DataSource = Nothing
    ComboBox_PatenteVehicDeu.Text = ""
    ComboBox_rutD.Text = ""
    Btn_informesD.Enabled = True
    Btn_historial.Enabled = True
    Button_cotizacionDeuda.Enabled = True
End Sub
Private Sub Btn_agregarD_Click(sender As Object, e As EventArgs) Handles Btn_agregarD.Click
    MostrarCombox()
    Btn_agregarD.Enabled = True
    Btn_ModificarD.Enabled = False
    Btn_aceptarD.Enabled = True
    Btn_cancelarD.Enabled = True
    Btn_informesD.Enabled = False
    Btn_historial.Enabled = False
    Button_cotizacionDeuda.Enabled = False
End Sub
Private Sub Btn_aceptarD_Click(sender As Object, e As EventArgs) Handles Btn_aceptarD.Click
    Dim ValorDeudaTotall1 As String
    Dim ValorDeudaFinal1 As Integer
    If Conex.State = ConnectionState.Open Then
        Conex.Close()
```

```
End If
Conex.Open()
Dim monto1 As String
Dim montoFinal1 As Integer
monto1 = Txt_ValorD.Text.Replace("$", "")
montoFinal1 = monto1.Replace(".", "")
Dim monto2 As String
Dim montoFinal2 As Integer
monto2 = Txt_MontoD.Text.Replace("$", "")
montoFinal1 = monto2.Replace(".", "")
If Txt_CodigoD.Text <> "" And Txt_ValorD.Text <> "" And Txt_MontoD.Text <> "" And Txt_repctD.Text <> "" And Txt_EstadoD.Text <> "" And DateTimePicker_Deu.Text <> "" And ComboBox_rutD.Text <> "" And Len(Txt_CodigoD.Text) <= 10 And montoFinal1.ToString.Length <= 7 And montoFinal2.ToString.Length <= 7 And Len(Txt_repctD.Text) <= 20 And Len(Txt_EstadoD.Text) <= 2 Then
    Dim consultaCodigoDeuda As String = String.Empty
    consultaCodigoDeuda = "Select * from deudas where codigo_deuda = '" & Txt_CodigoD.Text & "'"
    datos.CommandText = consultaCodigoDeuda
    datos.Connection = Conex
    mys = datos.ExecuteReader
    Dim cantideuda As Integer = 0
    Dim forma As String
    While mys.Read
        cantideuda = cantideuda + 1
    End While
    If cantideuda = 1 Then
        MsgBox("ya existe este codigo de deuda")
```

```
        mys.Close()
    Else
        mys.Close()
        Dim ValorDeudaTotall As String
        Dim ValorDeudaFinal As Integer
        Dim MontoPagar As String
        Dim MontoPagarFinal As Integer
        ValorDeudaTotall = Txt_ValorD.Text.Replace("$", "")
        ValorDeudaFinal = ValorDeudaTotall.Replace(".", "")
        MontoPagar = Txt_MontoD.Text.Replace("$", "")
        MontoPagarFinal = MontoPagar.Replace(".", "")
        Dim cadenaComboArray As String
        cadenaComboArray = Mid(ComboBox_rutD.Text, 1, 10)
        Dim cadenaComboArray2 As String
        cadenaComboArray2 = Mid(ComboBox_PatenteVehicDeu.Text, 1, 6)
        datos.CommandText = "insert into deudas
(codigo_deuda,deuda_valor_trabajo,monto_a_pagar,forma_repactacion_deuda,estado_deuda,
fecha_vencimiento,Clientes_rut_cliente,Vehiculos_patente_vehiculo) values('" &
Txt_CodigoD.Text & "','" & ValorDeudaFinal & "','" & MontoPagarFinal & "','" & Txt_repctD.Text
& "','" & Txt_EstadoD.Text & "','" & DateTimePicker_Deu.Text & "','" & cadenaComboArray & "','"
& cadenaComboArray2 & "')"
        datos.ExecuteNonQuery()
        Dim msg As New mb.ShowMessagebox
        msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
        msg.ShowBox("datos guardados", 0, mb.FStyle.Exclamation, "producto")
        Limpiar()
        MostrarListview()
        RefrescarAgregar()
```

```
    End If
Else
    If Txt_CodigoD.Text = "" Then
        MsgBox("campo codigo no puede estar vacio ")
    End If
    If Txt_CodigoD.Text <> "" And Not IsNumeric(Txt_CodigoD.Text) Then
        MsgBox(" codigo Debe ser numerico")
        Txt_CodigoD.Clear()
        Txt_CodigoD.Focus()
        Txt_ValorD.Clear()
    ElseIf Txt_CodigoD.Text <> "" And IsNumeric(Txt_CodigoD.Text) Then
        If Len(Txt_CodigoD.Text) > 10 Then
            MsgBox("Codigo debe tener solo hasta 10 digitos")
            If MsgBoxResult.Ok = 1 Then
                Txt_CodigoD.Clear()
                Txt_CodigoD.Focus()
                Txt_ValorD.Clear()
            End If
        End If
    End If
    If Txt_ValorD.Text = "" Then
        MsgBox(" valor no puede esta vacio ")
    End If
    If Txt_ValorD.Text <> "" And Not IsNumeric(Txt_ValorD.Text) Then
        MsgBox(" valor Debe ser numerico")
        Txt_ValorD.Clear()
        Txt_ValorD.Focus()
```

```
    Txt_MontoD.Clear()
ElseIf Txt_ValorD.Text <> "" And IsNumeric(Txt_ValorD.Text) Then
    ValorDeudaTotall1 = Txt_ValorD.Text.Replace("$", "")
    ValorDeudaFinal1 = ValorDeudaTotall1.Replace(".", "")
    If ValorDeudaFinal1.ToString.Length > 7 Then
        MsgBox("Valor de la deuda debe ser hasta de 7 digitos")
        Txt_ValorD.Clear()
        Txt_ValorD.Focus()
        Txt_MontoD.Clear()

    End If
End If
If Txt_MontoD.Text = "" Then
    MsgBox(" Monto no pued estar vacio ")
End If
If Txt_MontoD.Text <> "" And Not IsNumeric(Txt_MontoD.Text) Then
    MsgBox(" monto Debe ser numerico")
    Txt_MontoD.Clear()
    Txt_MontoD.Focus()
    Txt_repctD.Clear()
ElseIf Txt_MontoD.Text <> "" And IsNumeric(Txt_MontoD.Text) Then
    Dim MontoPagar As String
    Dim MontoPagarFinal As Integer
    MontoPagar = Txt_MontoD.Text.Replace("$", "")
    MontoPagarFinal = MontoPagar.Replace(".", "")
    If MontoPagarFinal.ToString.Length > 7 Then
        MsgBox("Monto debe ser hasta de 7 digitos")
```

```
        Txt_MontoD.Focus()
        Txt_MontoD.Clear()
        Txt_repctD.Clear()
    End I
End If
If Txt_repctD.Text = "" Then
    MsgBox("repactación  no puede estar vacio ")
End If
If Txt_repctD.Text <> "" And Not IsNumeric(Txt_repctD.Text) Then
    If Len(Txt_repctD.Text) > 20 Then
        MsgBox("Forma de Repacto debe tener solo hasta 20 digitos")
        If MsgBoxResult.Ok = 1 Then
            Txt_repctD.Clear()
            Txt_repctD.Focus()
            Txt_EstadoD.Clear()
        End If
    End If
ElseIf Txt_repctD.Text <> "" And IsNumeric(Txt_repctD.Text) Then
    MsgBox(" repacto Debe ser un texto")
    Txt_repctD.Clear()
    Txt_repctD.Focus()
End If
If Txt_EstadoD.Text = "" Then
    MsgBox("Estado no puede esta vacio ")
End If
If Txt_EstadoD.Text <> "" And IsNumeric(Txt_EstadoD.Text) Then
    MsgBox("Estado Debe ser texto")
```

```
            Txt_EstadoD.Clear()
            Txt_EstadoD.Focus()
        ElseIf Txt_EstadoD.Text <> "" And Not IsNumeric(Txt_EstadoD) Then
            If Len(Txt_EstadoD.Text) > 2 Then
                MsgBox("Estado de la deuda debe tener solo hasta 2 digitos")
                If MsgBoxResult.Ok = 1 Then
                    Txt_EstadoD.Clear()
                    Txt_EstadoD.Focus()




                End If
            End If
        End If
    End If
End Sub
Public Sub RefrescarModificar()
    Txt_CodigoD.Clear()
    Txt_EstadoD.Clear()
    Txt_MontoD.Clear()
    Txt_repctD.Clear()
    Txt_ValorD.Clear()
    ComboBox_rutD.DataSource = Nothing
    Btn_aceptarD.Enabled = False
    Btn_ModificarD.Enabled = True
End Sub
```

```
Private Sub Btn_ModificarD_Click(sender As Object, e As EventArgs) Handles Btn_ModificarD.Click
    Dim ValorDeudaTotall1 As String
    Dim ValorDeudaFinal1 As Integer
    Conex.Open()
    Dim monto1 As String
    Dim montoFinal1 As Integer
    monto1 = Txt_ValorD.Text.Replace("$", "")
    montoFinal1 = monto1.Replace(".", "")
    Dim monto2 As String
    Dim montoFinal2 As Integer
    monto2 = Txt_MontoD.Text.Replace("$", "")
    montoFinal1 = monto2.Replace(".", "")
    If List_Deud.SelectedItems.Count >= 0 Then
        If Txt_CodigoD.Text <> "" And Txt_ValorD.Text <> "" And Txt_MontoD.Text <> "" And Txt_repctD.Text <> "" And Txt_EstadoD.Text <> "" And DateTimePicker_Deu.Text <> "" And ComboBox_rutD.Text <> "" And Len(Txt_CodigoD.Text) <= 10 And montoFinal1.ToString.Length <= 7 And montoFinal2.ToString.Length <= 7 And Len(Txt_repctD.Text) <= 20 And Len(Txt_EstadoD.Text) <= 2 Then
            Dim ValorDeudaTotall As String
            Dim ValorDeudaFinal As Integer
            Dim MontoPagar As String
            Dim MontoPagarFinal As Integer
            Dim cadenaComboArray As String
            cadenaComboArray = Mid(ComboBox_rutD.Text, 1, 10)
            Dim cadenaComboArray2 As String
            cadenaComboArray2 = Mid(ComboBox_PatenteVehicDeu.Text, 1, 6)
            ValorDeudaTotall = Txt_ValorD.Text.Replace("$", "")
```

```
        ValorDeudaFinal = ValorDeudaTotall.Replace(".", "")
        MontoPagar = Txt_MontoD.Text.Replace("$", "")
        MontoPagarFinal = MontoPagar.Replace(".", "")
        datos.CommandText = "Update deudas set codigo_deuda ='" & Txt_CodigoD.Text & "' ,
deuda_valor_trabajo ='" & ValorDeudaFinal & "',  monto_a_pagar = '" & MontoPagarFinal & "',
forma_repactacion_deuda='" & Txt_repctD.Text & "', estado_deuda ='" & Txt_EstadoD.Text & "',
fecha_vencimiento ='" & DateTimePicker_Deu.Text & "', Clientes_rut_cliente = '" &
cadenaComboArray & "', Vehiculos_patente_vehiculo ='" & cadenaComboArray2 & " ' where
codigo_deuda ='" & Txt_CodigoD.Text & """
        datos.ExecuteNonQuery()
        Dim msg As New mb.ShowMessagebox
        msg.Fonts(New Font("Arial", 20, FontStyle.Regular))
        msg.ShowBox("actualizado", 0, mb.FStyle.Exclamation, "producto")
        'MsgBox("Actualizado")
        Limpiar()
        MostrarListview()
        RefrescarModificar()
    Else
        If Txt_CodigoD.Text = "" Then
            MsgBox("campo codigo no puede estar vacio ")
        End If
        If Txt_CodigoD.Text <> "" And Not IsNumeric(Txt_CodigoD.Text) Then
            MsgBox(" codigo Debe ser numerico")
            Txt_CodigoD.Clear()
            Txt_CodigoD.Focus()
            Txt_ValorD.Clear()
        ElseIf Txt_CodigoD.Text <> "" And IsNumeric(Txt_CodigoD.Text) Then
            If Len(Txt_CodigoD.Text) > 10 Then
```

```
        MsgBox("Codigo debe tener solo hasta 10 digitos")
        If MsgBoxResult.Ok = 1 Then
            Txt_CodigoD.Clear()
            Txt_CodigoD.Focus()
            Txt_ValorD.Clear()
        End If
    End If
End If
If Txt_ValorD.Text = "" Then
    MsgBox("Valor no puede estar vacio ")
End If
If Txt_ValorD.Text <> "" And Not IsNumeric(Txt_ValorD.Text) Then

    MsgBox(" valor Debe ser numerico")
    Txt_ValorD.Clear()
    Txt_ValorD.Focus()
    Txt_MontoD.Clear()
ElseIf Txt_ValorD.Text <> "" And IsNumeric(Txt_ValorD.Text) Then
    ValorDeudaTotall1 = Txt_ValorD.Text.Replace("$", "")
    ValorDeudaFinal1 = ValorDeudaTotall1.Replace(".", "")
    If ValorDeudaFinal1.ToString.Length > 7 Then
        MsgBox("Valor de la deuda debe ser hasta de 7 digitos")
        Txt_ValorD.Clear()
        Txt_ValorD.Focus()
        Txt_MontoD.Clear()
    End If
End If
```

```
If Txt_MontoD.Text = "" Then
   MsgBox(" Monto  no puede estar vacio ")
End If
If Txt_MontoD.Text <> "" And Not IsNumeric(Txt_MontoD.Text) Then
   MsgBox("Monto Debe ser numerico")
   Txt_MontoD.Clear()
   Txt_MontoD.Focus()
   Txt_repctD.Clear()
ElseIf Txt_MontoD.Text <> "" And IsNumeric(Txt_MontoD.Text) Then
   Dim MontoPagar As String
   Dim MontoPagarFinal As Integer
   MontoPagar = Txt_MontoD.Text.Replace("$", "")
   MontoPagarFinal = MontoPagar.Replace(".", "")
   If MontoPagarFinal.ToString.Length > 7 Then
      MsgBox("Monto debe ser hasta de 7 digitos")
      Txt_MontoD.Focus()
      Txt_MontoD.Clear()
      Txt_repctD.Clear()
   End If
End If
If Txt_repctD.Text = "" Then
   MsgBox("Repactacion no puede estar vacio ")
End If
If Txt_repctD.Text <> "" And Not IsNumeric(Txt_repctD.Text) Then
   If Len(Txt_repctD.Text) > 20 Then
      MsgBox("Forma de Repacto debe tener solo hasta 20 digitos")
      If MsgBoxResult.Ok = 1 Then
```

```
            Txt_repctD.Clear()
            Txt_repctD.Focus()
            Txt_EstadoD.Clear()
        End If
    End If
ElseIf Txt_repctD.Text <> "" And IsNumeric(Txt_repctD.Text) Then
    MsgBox("repacto Debe ser un texto")
    Txt_repctD.Clear()
    Txt_repctD.Focus()
End If
If Txt_EstadoD.Text = "" Then
    MsgBox("Estado no puede estar vacio ")
End If

If Txt_EstadoD.Text <> "" And IsNumeric(Txt_EstadoD.Text) Then
    MsgBox(" estado Debe ser texto")
    Txt_EstadoD.Clear()
    Txt_EstadoD.Focus()
ElseIf Txt_EstadoD.Text <> "" And Not IsNumeric(Txt_EstadoD) Then
    If Len(Txt_EstadoD.Text) > 2 Then
        MsgBox("Estado de la deuda debe tener solo hasta 2 digitos")
        If MsgBoxResult.Ok = 1 Then
            Txt_EstadoD.Clear()
            Txt_EstadoD.Focus()
        End If
    End If
End If
```

```
        End If
    End If
End Sub
Private Sub List_Deud_Click(sender As Object, e As EventArgs) Handles List_Deud.Click
    If List_Deud.SelectedItems.Count >= 0 Then
        MostrarCombox()
        Txt_CodigoD.Text = List_Deud.SelectedItems(0).SubItems(0).Text
        Txt_ValorD.Text = List_Deud.SelectedItems(0).SubItems(1).Text
        Txt_MontoD.Text = List_Deud.SelectedItems(0).SubItems(2).Text
        Txt_repctD.Text = List_Deud.SelectedItems(0).SubItems(3).Text
        Txt_EstadoD.Text = List_Deud.SelectedItems(0).SubItems(4).Text
        DateTimePicker_Deu.Text = List_Deud.SelectedItems(0).SubItems(5).Text
        ComboBox_rutD.Text = List_Deud.SelectedItems(0).SubItems(6).Text
        ComboBox_PatenteVehicDeu.Text = List_Deud.SelectedItems(0).SubItems(7).Text
        Txt_CodigoD.Enabled = False
        Btn_aceptarD.Enabled = False
        Btn_cancelarD.Enabled = False
    End If
End Sub
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.Close()
    Form2.Show()
End Sub
Private Sub Btn_informesD_Click(sender As Object, e As EventArgs) Handles
Btn_informesD.Click
    Me.Close()
    VentanaInforme.Show()
```

```
End Sub
Private Sub Btn_historial_Click(sender As Object, e As EventArgs) Handles Btn_historial.Click
    Me.Close()
    HistorialBoleta.Show()
End Sub
Private Sub Btn_BuscarD_Click(sender As Object, e As EventArgs) Handles Btn_BuscarD.Click
    Dim Li As New ListViewItem
    Limpiar()
    Dim fechaVencimientoDeuB As Date
    Try
        Conex.Open()
        Dim consultaBuscar As String = String.Empty
        consultaBuscar = "SELECT * FROM deudas where codigo_deuda LIKE '%" & Txt_BuscarDeud.Text & "%' or deuda_valor_trabajo LIKE '%" & Txt_BuscarDeud.Text & "%' or monto_a_pagar LIKE '%" & Txt_BuscarDeud.Text & "%' or forma_repactacion_deuda LIKE '%" & Txt_BuscarDeud.Text & "%' or estado_deuda LIKE '%" & Txt_BuscarDeud.Text & "%'or fecha_vencimiento LIKE '%" & Txt_BuscarDeud.Text & "%' or Clientes_rut_cliente LIKE '%" & Txt_BuscarDeud.Text & "%' or Vehiculos_patente_vehiculo LIKE '%" & Txt_BuscarDeud.Text & "%'"
        datos.CommandText = consultaBuscar
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim ValorDeudaT As String
        Dim MontoAPagarPorDeuda As String
        While mys.Read
            fechaVencimientoDeuB = mys.Item("fecha_vencimiento").ToString
            ValorDeudaT = Format(CType(mys.Item("deuda_valor_trabajo"), Decimal), "$#,##0")
            MontoAPagarPorDeuda = Format(CType(mys.Item("monto_a_pagar"), Decimal), "$#,##0")
```

```
            Li = List_Deud.Items.Add(mys.Item("codigo_deuda").ToString)
            Li.SubItems.Add(ValorDeudaT)
            Li.SubItems.Add(MontoAPagarPorDeuda)
            Li.SubItems.Add(mys.Item("forma_repactacion_deuda").ToString)
            Li.SubItems.Add(mys.Item("estado_deuda").ToString)
            Li.SubItems.Add(fechaVencimientoDeuB)
            Li.SubItems.Add(mys.Item("Clientes_rut_cliente").ToString)
            Li.SubItems.Add(mys.Item("Vehiculos_patente_vehiculo").ToString)
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub Txt_BuscarDeud_TextChanged(sender As Object, e As EventArgs) Handles Txt_BuscarDeud.TextChanged
    If Txt_BuscarDeud.Text = "" Then
        Conex.Open()
        Limpiar()
        MostrarListview()
    End If
End Sub
Private Sub Btn_cancelarD_Click(sender As Object, e As EventArgs) Handles Btn_cancelarD.Click
    Txt_CodigoD.Clear()
    Txt_EstadoD.Clear()
    Txt_MontoD.Clear()
```

```
        Txt_repctD.Clear()
        Txt_ValorD.Clear()
        ComboBox_PatenteVehicDeu.DataSource = Nothing
        ComboBox_rutD.DataSource = Nothing
        Btn_aceptarD.Enabled = False
        Btn_cancelarD.Enabled = False
        Btn_agregarD.Enabled = True
        Btn_ModificarD.Enabled = True
    End Sub
    Private Sub Txt_MontoD_LostFocus(sender As Object, e As EventArgs) Handles
Txt_MontoD.LostFocus
        Dim MontoPagar As String
        Dim MontoPagarFinal As Integer
        If Txt_MontoD.Text <> "" Then
            Txt_MontoD.Text = Format(CType(Me.Txt_MontoD.Text, Decimal), "#.###,##")
        End If
    End Sub
    Private Sub Txt_ValorD_LostFocus(sender As Object, e As EventArgs) Handles
Txt_ValorD.LostFocus
        Dim ValorDeudaTotall As String
        Dim ValorDeudaFinal As Integer
        If Txt_ValorD.Text <> "" Then
            Txt_ValorD.Text = Format(CType(Me.Txt_ValorD.Text, Decimal), "#.###,##")
        End If
    End Sub
    Private Sub Button_consultaDeudacuota_Click(sender As Object, e As EventArgs) Handles
Button_consultaDeudacuota.Click
        DataGridView1.Visible = True
```

```
Dim consultaMostrarCLientes As String
Dim consultaStockmostrarD As String
Dim dtr As MySqlDataAdapter
Dim tabla As DataTable
Dim dtr2 As MySqlDataAdapter
Dim tabla2 As DataTable
Dim cont As Integer
Dim FormaRD As String
Dim FormaFinal As String

Try
    consultaMostrarCLientes = "select DISTINCT clientes.rut_cliente , clientes.nombre_cliente from clientes , deudas where deudas.Clientes_rut_cliente = '" & TextBoxConsultaDeuda.Text & "' and clientes.rut_cliente='" & TextBoxConsultaDeuda.Text & "'"
    consultaStockmostrarD = "Select Count(deudas.Clientes_rut_cliente), forma_repactacion_deuda from deudas where deudas.Clientes_rut_cliente='" & TextBoxConsultaDeuda.Text & "' and estado_deuda='PG'"
    dtr = New MySqlDataAdapter(consultaMostrarCLientes, Conex)
    tabla = New DataTable
    dtr.Fill(tabla)
    DataGridView1.DataSource = tabla
    dtr2 = New MySqlDataAdapter(consultaStockmostrarD, Conex)
    tabla2 = New DataTable
    dtr2.Fill(tabla2)
    DataGridView2.DataSource = tabla2
```

```
        DataGridView1.Rows(0).Cells("Cuotas").Value =
DataGridView2.Rows(0).Cells("forma_repactacion_deuda").Value

        FormaRD = Mid(DataGridView2.Rows(0).Cells("forma_repactacion_deuda").Value, 4)

        FormaFinal = Mid(FormaRD, 1, 2)

        cont = FormaFinal -
DataGridView2.Rows(0).Cells("Count(deudas.Clientes_rut_cliente)").Value

        DataGridView1.Rows(0).Cells("CuotasFinal").Value = cont

        If DataGridView1.Rows(0).Cells("CuotasFinal").Value = "0" Then

          Dim CambiarEstado As String

          CambiarEstado = "P"

          Conex.Open()

          datos.CommandText = "Update clientes set situacion_financiera_cliente ='" &
CambiarEstado & "' where rut_cliente ='" & DataGridView1.Rows(0).Cells("rut_cliente").Value &
"'"

          datos.ExecuteNonQuery()

        End If

      Catch ex As Exception

      End Try

    End Sub

    Private Sub Button_LimpiarCuota_Click(sender As Object, e As EventArgs) Handles
Button_LimpiarCuota.Click

      TextBoxConsultaDeuda.Clear()

      DataGridView1.DataSource = Nothing

      DataGridView2.DataSource = Nothing

      DataGridView1.Visible = False

    End Sub

    Private Sub Txt_ValorD_Leave(sender As Object, e As EventArgs) Handles Txt_ValorD.Leave

      Txt_ValorD.Text = Format(Val(Txt_ValorD.Text.ToString), "$#,##0")
```

```
    End Sub
    Private Sub Txt_MontoD_Leave(sender As Object, e As EventArgs) Handles Txt_MontoD.Leave
        Txt_MontoD.Text = Format(Val(Txt_MontoD.Text.ToString), "$#,##0")
    End Sub
    Private Sub Button_cotizacionDeuda_Click(sender As Object, e As EventArgs) Handles
Button_cotizacionDeuda.Click
        CotizacionDeuda.Show()
    End Sub
End Class
```

- **Módulo Pago**

```
Imports System.Data.Sql
Imports System.Text
Imports MySql.Data.MySqlClient
Imports System.Globalization
Imports System.Threading
Imports System.Security.Cryptography
Public Class Pagos
    Dim Conex As New MySqlConnection
    Dim mysql As New MySqlDataAdapter
    Dim mysql1 As New MySqlDataAdapter
    Dim mys As MySqlDataReader
    Dim Mysqll As DataSet
    Dim Mysqll1 As DataSet
    Dim datos As New MySqlCommand
    Dim mostrar As New DataTable
    Private Sub Pagos_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
    Conex.ConnectionString = "server=localhost;user=root;password=
;database=taller_mecanico"
    If Conex.State = ConnectionState.Open Then
        Conex.Close()
    End If
    Conex.Open()
    MostrarListview()
    Btn_aceptarPago.Enabled = False
    Btn_cancelarPago.Enabled = False
    System.Threading.Thread.CurrentThread.CurrentCulture = New
System.Globalization.CultureInfo("es-CL")
  End Sub
  Public Sub MostrarListview()
    Dim Li As New ListViewItem
    Try
        Dim consulta As String = String.Empty
        Dim fechapago As Date
        Dim montoPag As String
        consulta = "SELECT * FROM pagos"
        datos.CommandText = consulta
        datos.Connection = Conex
        mys = datos.ExecuteReader
        While mys.Read
            fechapago = mys.Item("fecha_pago").ToString
            montoPag = Format(CType(mys.Item("monto_pagado"), Decimal), "$#,##0")
            Li = List_pago.Items.Add(mys.Item("codigo_pago").ToString)
            'Li.SubItems.Add(mys.Item("monto_pagado").ToString)
```

```
            Li.SubItems.Add(montoPag)
            Li.SubItems.Add(fechapago)
            Li.SubItems.Add(mys.Item("Deudas_codigo_deuda").ToString)
        End While
        Conex.Close()
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Public Sub MostrarCombox()
    Try
        Conex.Open()
        Dim consulta As String = String.Empty
        consulta = "SELECT * FROM deudas"
        mysql = New MySqlDataAdapter(consulta, Conex)
        Mysqll = New DataSet
        Mysqll.Tables.Add("deudas")
        mysql.Fill(Mysqll.Tables("deudas"))
        ComboBox_codPago.DataSource = Mysqll.Tables("deudas")
        ComboBox_codPago.DisplayMember = "codigo_deuda"
    Catch ex As Exception
    End Try
End Sub
Public Sub Limpiar()
    For i = 0 To List_pago.Items.Count - 1
        List_pago.Items.Clear()
    Next
```

```
End Sub
Public Sub RefrescarAgregar()
    Txt_CodigoP.Clear()
    Txt_MontoP.Clear()
    ComboBox_codPago.DataSource = Nothing
    Btn_aceptarPago.Enabled = False
    Btn_cancelarPago.Enabled = False
    Btn_ModificarPago.Enabled = True
End Sub
Private Sub Btn_agregarPago_Click(sender As Object, e As EventArgs) Handles Btn_agregarPago.Click
    MostrarCombox()
    Btn_agregarPago.Enabled = True
    Btn_ModificarPago.Enabled = False
    Btn_aceptarPago.Enabled = True
    Btn_cancelarPago.Enabled = True
End Sub
Private Sub Btn_aceptarPago_Click(sender As Object, e As EventArgs) Handles Btn_aceptarPago.Click
    Dim monto2 As String
    Dim montoFinal2 As Integer
    monto2 = Txt_MontoP.Text.Replace("$", "")
    montoFinal2 = monto2.Replace(".", "")
    If Txt_CodigoP.Text <> "" And Txt_MontoP.Text <> "" And DateTimePicker_pagoFech.Text <> "" And ComboBox_codPago.Text <> "" And Len(Txt_CodigoP.Text) <= 10 And montoFinal2.ToString.Length <= 7 Then
        Dim consultaCodigoPago As String = String.Empty
```

```
        consultaCodigoPago = "Select * from pagos where codigo_pago = '" & Txt_CodigoP.Text &
"'"
        Dim monto As String
        datos.CommandText = consultaCodigoPago
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim cantiPago As Integer = 0
        While mys.Read
            cantiPago = cantiPago + 1
        End While
        If cantiPago = 1 Then
            MsgBox("ya existe este codigo de pago")
            mys.Close()
        Else
            mys.Close()
            Dim montoFinal As Integer
            monto = Txt_MontoP.Text.Replace("$", "")
            montoFinal = monto.Replace(".", "")
            MsgBox(Txt_MontoP.Text)
            datos.CommandText = "insert into pagos
(codigo_pago,monto_pagado,fecha_pago,Deudas_codigo_deuda) values('" & Txt_CodigoP.Text
& "','" & montoFinal & "','" & DateTimePicker_pagoFech.Text & "','" & ComboBox_codPago.Text
& "')"
            datos.ExecuteNonQuery()
            Dim estadoDeudaModif As String
            estadoDeudaModif = "PG"
            datos.CommandText = "Update deudas set estado_deuda ='" & estadoDeudaModif & "
' where codigo_deuda ='" & ComboBox_codPago.Text & "'"
```

```
        datos.ExecuteNonQuery()
        MsgBox("Datos guardados")
        Limpiar()
        MostrarListview()
        RefrescarAgregar()
    End If
Else
    If Txt_CodigoP.Text = "" Then
        MsgBox(" codigo no puede estar vacio ")
    ElseIf Txt_CodigoP.Text <> "" And Not IsNumeric(Txt_CodigoP.Text) Then
        MsgBox("Codigo debe ser numerico")
        Txt_CodigoP.Clear()
        Txt_CodigoP.Focus()
    ElseIf Txt_CodigoP.Text <> "" And IsNumeric(Txt_CodigoP.Text) Then
        If Len(Txt_CodigoP.Text) > 10 Then
            MsgBox("Codigo debe tener solo hasta 10 digitos")
            If MsgBoxResult.Ok = 1 Then
                Txt_CodigoP.Clear()
                Txt_CodigoP.Focus()
                Txt_MontoP.Clear()
            End If
        End If
    End If
    If Txt_MontoP.Text = "" Then
        MsgBox("monto no puede estar vacio ")
    ElseIf Txt_MontoP.Text <> "" And Not IsNumeric(Txt_CodigoP.Text) Then
        MsgBox("El monto debe ser numerico")
```

```
            Txt_MontoP.Clear()
            Txt_MontoP.Focus()
        ElseIf Txt_MontoP.Text <> "" And IsNumeric(Txt_MontoP.Text) Then
            Dim monto As String
            Dim montoFinal As Integer
            monto = Txt_MontoP.Text.Replace("$", "")
            montoFinal = monto.Replace(".", "")
            If montoFinal.ToString.Length > 7 Then
                MsgBox("Monto debe ser hasta de 7 digitos")
                Txt_MontoP.Clear()
                Txt_MontoP.Focus()
            End If
        End If
    End If
End Sub
Public Sub RefrescarModificar()
    Txt_CodigoP.Clear()
    Txt_MontoP.Clear()
    ComboBox_codPago.DataSource = Nothing
    Btn_aceptarPago.Enabled = False
    Btn_cancelarPago.Enabled = False
End Sub
Private Sub Btn_ModificarPago_Click(sender As Object, e As EventArgs) Handles Btn_ModificarPago.Click
    Dim monto2 As String
    Dim montoFinal2 As Integer
    monto2 = Txt_MontoP.Text.Replace("$", "")
```

```
montoFinal2 = monto2.Replace(".", "")
If List_pago.SelectedItems.Count >= 0 Then
    If Txt_CodigoP.Text <> "" And Txt_MontoP.Text <> "" And DateTimePicker_pagoFech.Text <> "" And ComboBox_codPago.Text <> "" And Len(Txt_CodigoP.Text) <= 10 And montoFinal2.ToString.Length <= 7 Then
        Dim monto As String
        Dim montoFinal As Integer
        monto = Txt_MontoP.Text.Replace("$", "")
        montoFinal = monto.Replace(".", "")
        datos.CommandText = "Update pagos set monto_pagado ='" & montoFinal & "' , fecha_pago ='" & DateTimePicker_pagoFech.Text & "', Deudas_codigo_deuda = '" & ComboBox_codPago.Text & " ' where codigo_pago ='" & Txt_CodigoP.Text & "'"
        datos.ExecuteNonQuery()
        MsgBox("Actualizado")
        Limpiar()
        MostrarListview()
        RefrescarModificar()
    Else

        If Txt_CodigoP.Text = "" Then
            MsgBox("Los codigo campos no pueden estar vacios ")
        ElseIf Txt_CodigoP.Text <> "" And Not IsNumeric(Txt_CodigoP.Text) Then
            MsgBox("Codigo debe ser numerico")
            Txt_CodigoP.Clear()
            Txt_CodigoP.Focus()
        ElseIf Txt_CodigoP.Text <> "" And IsNumeric(Txt_CodigoP.Text) Then
            If Len(Txt_CodigoP.Text) > 10 Then
                MsgBox("Codigo debe tener solo hasta 10 digitos")
```

```
            If MsgBoxResult.Ok = 1 Then
                Txt_CodigoP.Clear()
                Txt_CodigoP.Focus()
                Txt_MontoP.Clear()
            End If
        End If
    End If
    If Txt_MontoP.Text = "" Then
        MsgBox(" monto  no pueden estar vacio ")
    ElseIf Txt_MontoP.Text <> "" And Not IsNumeric(Txt_CodigoP.Text) Then
        'MsgBox("Codigo debe ser numerico")
        MsgBox("El monto debe ser numerico")
        Txt_MontoP.Clear()
        Txt_MontoP.Focus()
    ElseIf Txt_MontoP.Text <> "" And IsNumeric(Txt_MontoP.Text) Then
        Dim monto As String
        Dim montoFinal As Integer
        monto = Txt_MontoP.Text.Replace("$", "")
        montoFinal = monto.Replace(".", "")
        If montoFinal.ToString.Length > 7 Then
            MsgBox("Monto debe ser hasta de 7 digitos")
            Txt_MontoP.Clear()
            Txt_MontoP.Focus()
        End If
    End If
  End If
End If
```

```
End Sub
Private Sub List_pago_Click(sender As Object, e As EventArgs) Handles List_pago.Click
  If List_pago.SelectedItems.Count >= 0 Then
    MostrarCombox()
    Txt_CodigoP.Text = List_pago.SelectedItems(0).SubItems(0).Text
    Txt_MontoP.Text = List_pago.SelectedItems(0).SubItems(1).Text
    DateTimePicker_pagoFech.Text = List_pago.SelectedItems(0).SubItems(2).Text
    ComboBox_codPago.Text = List_pago.SelectedItems(0).SubItems(3).Text
    Txt_CodigoP.Enabled = False
    Btn_aceptarPago.Enabled = False
    Btn_cancelarPago.Enabled = False
  End If
End Sub
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
  Me.Close()
  Form2.Show()
End Sub

Private Sub Btn_BuscarD_Click(sender As Object, e As EventArgs) Handles Btn_BuscarD.Click
  Dim Li As New ListViewItem
  Limpiar()
  Dim fechapagoB As Date
  Try
    Conex.Open()
    Dim consultaBuscar As String = String.Empty
    consultaBuscar = "SELECT * FROM pagos where codigo_pago LIKE '%" & Txt_BuscarDeud.Text & "%' or monto_pagado LIKE'%" & Txt_BuscarDeud.Text & "%' or
```

```
fecha_pago LIKE '%" & Txt_BuscarDeud.Text & "%' or Deudas_codigo_deuda LIKE '%" & Txt_BuscarDeud.Text & "%'"
        'consultaBuscar = "Select * FROM vehiculos where patente_vehiculo Like '%" & Txt_vehiculos.Text & "%' or marca_vehiculo LIKE '%" & Txt_vehiculos.Text & "%' or modelo_vehiculo LIKE '%" & Txt_vehiculos.Text & "%' or fecha_recepcion_vehiculo LIKE'%" & Txt_vehiculos.Text & "%' or fecha_entrega_vehiculo LIKE '%" & Txt_vehiculos.Text & "%' or Clientes_rut_cliente LIKE '%" & Txt_vehiculos.Text & "%'"
        datos.CommandText = consultaBuscar
        datos.Connection = Conex
        mys = datos.ExecuteReader
        Dim fechapago As Date
        Dim montoPag As String
        While mys.Read
          fechapagoB = mys.Item("fecha_pago").ToString
          montoPag = Format(CType(mys.Item("monto_pagado"), Decimal), "$#,##0")
          'Li = List_pago.Items.Add(mys.Item("codigo_pago").ToString)
          Li.SubItems.Add(mys.Item("monto_pagado").ToString)
          Li.SubItems.Add(montoPag)
          Li.SubItems.Add(fechapagoB)
          Li.SubItems.Add(mys.Item("Deudas_codigo_deuda").ToString)
        End While
        Conex.Close()
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
    End Sub
    Private Sub Txt_BuscarDeud_TextChanged(sender As Object, e As EventArgs) Handles Txt_BuscarDeud.TextChanged
      If Txt_BuscarDeud.Text = "" Then
```

```
        Conex.Open()
        Limpiar()
        MostrarListview()
    End If
End Sub
Private Sub Button_calcularPago_Click(sender As Object, e As EventArgs) Handles Button_calcularPago.Click
    Me.Close()
    CalcularPago.Show()
End Sub
Private Sub Txt_MontoP_LostFocus(sender As Object, e As EventArgs) Handles Txt_MontoP.LostFocus
    If Txt_MontoP.Text <> "" Then
        Txt_MontoP.Text = Format(CType(Me.Txt_MontoP.Text, Decimal), "#.###,##")
    End If
End Sub
Private Sub Txt_MontoP_Leave(sender As Object, e As EventArgs) Handles Txt_MontoP.Leave
    Txt_MontoP.Text = Format(Val(Txt_MontoP.Text.ToString), "$#,##0")
End Sub
Private Sub Btn_cancelarPago_Click(sender As Object, e As EventArgs) Handles Btn_cancelarPago.Click
    Txt_CodigoP.Clear()
    ComboBox_codPago.DataSource = Nothing
    Btn_agregarPago.Enabled = True
    Btn_ModificarPago.Enabled = True
    Btn_aceptarPago.Enabled = False
    Btn_cancelarPago.Enabled = False
    Conex.Close()
```

```
    End Sub
End Class
```