FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

(Monclova, Coahuila)

PUNTO DE VENTA COMERCIALIZADORA

**Nombre:** juanita gpe. Calvillo de la cerda

**Profesora:** DR. SERGIO CASTAÑEDA

**Materia:** Sistemas de Informacion 2

**Salón:** 6

**Semestre:** 9no

**Turno:** Vespertino

26/Noviembre/2018

**ÍNDICE**

**Portada ………………………………………………………………………… 1**

**Indice …………………………………………………………………………… 2**

**Introduccion …………………………………………………………………… 3**

**Objetivo General ……………………………………………………………… 4**

**Objetivos Particulares ……………………………………………………..... 4**

**Arquitectura Computacional ……………………………………………….. 5**

**Estructura de Datos ………………………………………………….. 5**

**Modelo de Daros ……………………………………………………… 6**

**Definicion de tablas ………………………………………………….. 8**

**Documentacion ……………………………………………………………….. 10**

**Pantallas y Codigos ………………………………………………….. 10**

**Procedimientos lmacenados BD …………………………………... 38**

**Reportes ………………………………………………………………... 45**

**Referencias …………………………………………………………………….. 47**

**INTRODUCCIÓN**

Un punto de venta siempre es útil en un negocio, ya que con esto se tiene un buen control en los inventarios de los productos y se tiene un respaldo de las boletas y facturas, las cuales contienen el monto de las ventas y la fecha en la cual se realizó.

En cualquier negocio es muy importante tener una base de datos en la cual se tengan los datos de los empleados, los cargos de estos, los datos de sus clientes más frecuentes, los detalles de las ventas y los productos con sus precios. Estos son los datos que más se necesitan resguardar, ya que, con estos podemos crear un punto de venta funcional.

En el siguiente trabajo se muestra la base de datos de un punto de ventas, junto con la creación de sus tablas y los datos que están introducidos en cada tabla, esta base de datos fue creada en SQL Server 2012, también tiene los formularios de cada pantalla y sus clases y código, los formularios se crearon en Visual Basic 2013.

Este punto de venta es para un negocio el cual se dedica a comercializar productos tales como kits de cámaras de seguridad, bobinas de cables para redes, alarmas, detectores, amplificadores de wifi, entre otras cosas.



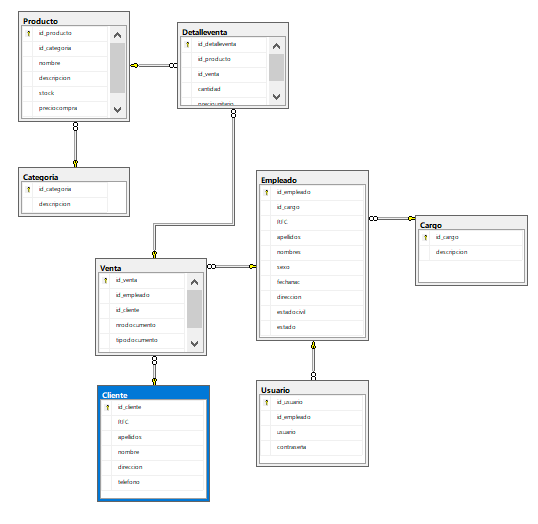
**OBJETIVO GENERAL**

El objetivo general de este proyecto es ayudar a los negocios pequeños a que puedan tener un punto de ventas fácil y básico, con el cual puedan tener un control de sus ventas, de sus clientes, empleados y productos. También es el aprendizaje mas afondo sobre algunas funciones en especificas de SQL Server y Visual Basic.

**OBJETIVOS PARTICULARES**

1. Crear un proyecto el cual le pueda servir a muchas personas en el manejo de sus negocios.
2. Desarrollar un proyecto funcional
3. Agregar nuevos clientes, empleados, productos, y ventas.
4. Modificar los datos de algún cliente, empleado, producto o venta.
5. Aprender más sobre la programación de proyectos que sirvan para problemas reales como lo es la gestión de ventas de un negocio.
6. Investigar y aprender más sobre la creación de base de datos, y saber más sobre la creación de diferentes tipos de procedimientos, en los cuales se utilicen queries para modificar o juntar información, tales como el select, insert, update, entre otras.

**ARQUITECTURA COMPUTACIONAL**

****

**MODELO DE DATOS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cliente** | | | | |
| **Columna** | **Tipo de datos** | | **Identidad** | |
| id\_cliente | int | | Primary key, not null | |
| RFC | varchar(13) | | null | |
| apellidos | varchar(50) | | null | |
| nombre | varchar(50) | | null | |
| direccion | varchar(50) | | null | |
| telefono | int | | null | |
|  |  | |  | |
| **Cargo** | | | | |
| **Columna** | **Tipo de datos** | | **Identidad** | |
| id\_cargo | int | | Primary key, not null | |
| descripcion | varchar(50) | | null | |
|  |  | |  | |
| **Empleado** | | | | |
| **Columna** | **Tipo de datos** | | **Identidad** | |
| id\_empleado | int | | Primary key, not null | |
| id\_cargo | int | | Foreign key, null | |
| RFC | varchar(13) | | null | |
| apellidos | varchar(50) | | null | |
| nombres | varchar(50) | | null | |
| sexo | varchar(30) | | null | |
| fechanac | date | | null | |
| direccion | varchar(50) | | null | |
| estadocivil | varchar(30) | | null | |
| estado | varchar(30) | | null | |
|  |  | |  | |
| **Categoría** | | | | |
| **Columna** | | **Tipo de datos** | | **Identidad** |
| id\_categoria | | int | | Primary key, not null |
| descripcion | | varchar(50) | | null |

|  |  |  |
| --- | --- | --- |
| **Productos** | | |
| **Columna** | **Tipo de datos** | **Identidad** |
| id\_producto | int | Primary key, not null |
| id\_categoria | int | Foreign key, null |
| nombre | varchar(50) | null |
| descripcion | varchar(60) | null |
| stock | int | null |
| preciocompra | decimal(10,2) | null |
| precio venta | decimal(10,2) | null |
|  |  |  |
| **Venta** | | |
| **Columna** | **Tipo de datos** | **Identidad** |
| id\_venta | int | Primary key, not null |
| id\_empleado | int | Foreign Key, null |
| id\_cliente | int | Foreign Key, null |
| nrodocumento | varchar(50) | null |
| tipodocumento | varchar(40) | null |
| fechaventa | date | null |
|  |  |  |
| **Detalleventa** | | |
| **Columna** | **Tipo de datos** | **Identidad** |
| id\_detalleventa | int | Primary key, not null |
| id\_producto | int | Foreign key, null |
| id\_venta | int | Foreign key, null |
| cantidad | int | null |
| preciounitario | decimal(10,2) | null |
|  |  |  |
| **Usuario** | | |
| **Columna** | **Tipo de datos** | **Identidad** |
| id\_usuario | int | Primary key, not null |
| id\_empleado | int | Foreign key, null |
| Usuario | varchar(50) | null |
| contraseña | varchar(50) | null |
|  |  |  |

**DEFINICION DE TABLAS**

**Tabla cliente**

create table Cliente(

id\_cliente int identity primary key not null,

RFC varchar(13),

apellidos nvarchar(50),

nombre nvarchar(50),

direccion nvarchar(50),

telefono int

)

**Tabla cargo de empleado**

create table Cargo(

id\_cargo int identity primary key not null,

descripcion nvarchar(50)

)

**Tabla Empleado**

create table Empleado(

id\_empleado int identity primary key not null,

id\_cargo int foreign key references Cargo(id\_cargo),

RFC varchar(13),

apellidos nvarchar(50),

nombres nvarchar(50),

sexo nvarchar(30),

fechanac date,

direccion nvarchar(50),

estadocivil nvarchar(30)

estado nvarchar(30)

)

**Tabla de categoría de productos**

create table Categoria(

id\_categoria int identity primary key not null,

descripcion nvarchar(50)

)

**Tabla de productos**

create table Producto(

id\_producto int identity primary key,

id\_categoria int foreign key references Categoria(id\_categoria),

nombre nvarchar(50),

descripcion nvarchar(60),

stock int,

preciocompra decimal(10,2),

precioventa decimal(10,2)

)

**Tabla de Ventas**

create table Venta(

id\_venta int identity primary key,

id\_empleado int foreign key references Empleado(id\_empleado),

id\_cliente int foreign key references Cliente(id\_cliente),

nrodocumento nvarchar(50),

tipodocumento nvarchar(40),

fechaventa date

)

**Tabla de detalle venta (aquí se guarda el importe de lo pagado)**

create table Detalleventa(

id\_detalleventa int identity primary key,

id\_producto int foreign key references Producto(id\_producto),

id\_venta int foreign key references Venta(id\_venta),

cantidad int,

preciounitario decimal(10,2)

)

**Tabla de usuario (solo empleados)**

create table Usuario(

id\_usuario int identity primary key,

id\_empleado int foreign key references Empleado(id\_empleado),

usuario nvarchar(50),

contraseña nvarchar(50)

)

**DOCUMENTACIÓN (Pantallas y Codigos)**

**Login**

Este en un pequeño login con el cual podrán acceder los empleados al sistema.

****

Public Class Login

Dim valores As New LUsuario

Dim funciones As New DUsuarios

Private Sub btniniciar\_Click(sender As Object, e As EventArgs) Handles btniniciar.Click

Try

valores.\_usuario = txtusuario.Text

valores.\_contraseña = txtcontraseña.Text

If funciones.AccederUsuario(valores) Then

MsgBox("Bienvenido al Sistema", MsgBoxStyle.Information, "Mensaje del Sistema")

FInicio.Show()

Me.Hide()

Else

MsgBox("Usuario o Contraseña Incorrecta", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnsalir\_Click(sender As Object, e As EventArgs) Handles btnsalir.Click

Me.Close()

End Sub

Private Sub txtusuario\_Click(sender As Object, e As EventArgs) Handles txtusuario.Click

validar()

End Sub

Private Sub txtusuario\_TextChanged(sender As Object, e As EventArgs) Handles txtusuario.TextChanged

validar()

End Sub

Private Sub txtcontraseña\_Click(sender As Object, e As EventArgs) Handles txtcontraseña.Click

validar()

End Sub

Private Sub txtcontraseña\_TextChanged(sender As Object, e As EventArgs) Handles txtcontraseña.TextChanged

validar()

End Sub

Private Sub lblusuario\_Click(sender As Object, e As EventArgs) Handles lblusuario.Click

lblusuario.Visible = False

txtusuario.Focus()

End Sub

Private Sub lblcontraseña\_Click(sender As Object, e As EventArgs) Handles lblcontraseña.Click

lblcontraseña.Visible = False

txtcontraseña.Focus()

End Sub

Public Sub validar()

If Trim(txtcontraseña.Text) = "" Then

lblcontraseña.Visible = True

Else

lblcontraseña.Visible = False

End If

If Trim(txtusuario.Text) = "" Then

lblusuario.Visible = True

Else

lblusuario.Visible = False

End If

End Sub

Private Sub Login\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

**Form de Inicio**

Esta es la pagina principal del sistema, en ella podemos ver las opciones de cliente, empleados, productos y ventas, también se puede ver el listado de ventas, el cual también sale cual oprimimos la opción ventas.



Public Class FInicio

Private Sub FInicio\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

PanelContenido(FListadoVentas)

Timer1.Start()

lblequipo.Text = (Environment.UserName).ToUpper

lbldia.Text = Now.Date

End Sub

Public Sub PanelContenido(ByVal Formulario As Form)

Panel1.Controls.Clear()

Formulario.TopLevel = False

Formulario.FormBorderStyle = Windows.Forms.FormBorderStyle.None

Formulario.Dock = DockStyle.Fill

Panel1.Controls.Add(Formulario)

Formulario.Show()

For Each cajatexto In Formulario.Controls

If TypeOf (cajatexto) Is TextBox Then

cajatexto.Clear()

End If

Next

End Sub

Private Sub btnclientes\_Click(sender As Object, e As EventArgs) Handles btnclientes.Click

PanelContenido(FListadoClientes)

End Sub

Private Sub btnempleados\_Click(sender As Object, e As EventArgs) Handles btnempleados.Click

PanelContenido(FListadoEmpleados)

End Sub

Private Sub btnproductos\_Click(sender As Object, e As EventArgs) Handles btnproductos.Click

PanelContenido(FListadoProductos)

End Sub

Private Sub btnventas\_Click(sender As Object, e As EventArgs) Handles btnventas.Click

PanelContenido(FListadoVentas)

End Sub

Private Sub Timer1\_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

lblhora.Text = TimeOfDay

lbldia.Text = Now.Date

End Sub

Private Sub btnSalir\_Click(sender As Object, e As EventArgs) Handles btnSalir.Click

If MsgBox("¿Esta Seguro de salir del Sistema?", MsgBoxStyle.Information + MsgBoxStyle.YesNo, "Mensaje del Sistema") = MsgBoxResult.Yes Then

End

End If

End Sub

Private Sub Label1\_Click(sender As Object, e As EventArgs) Handles Label1.Click

End Sub

End Class

**Form de Clientes**

Este listado sale cuando oprimimos la opción de clientes, aquí se muestran los clientes que están registrados al sistema, también tenemos dos botones con los cuales podemos agregar un nuevo cliente o editar uno ya existente.



Imports System.Data.SqlClient

Public Class FListadoClientes

Dim valores As New LCliente

Dim funciones As New DCliente

Dim vista As New DataView

Private Sub FListadoClientes\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

mostrarcliente()

End Sub

Private Sub txtbuscar\_TextChanged(sender As Object, e As EventArgs) Handles txtbuscar.TextChanged

Try

vista.RowFilter = cbobuscar.Text & " like '" & txtbuscar.Text & "%'"

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnNuevo\_Click(sender As Object, e As EventArgs) Handles btnNuevo.Click

Dim ventana As New FCliente

ventana.ShowDialog()

mostrarcliente()

End Sub

Public Sub mostrarcliente()

vista = funciones.MostrarClientes.DefaultView

Listado.DataSource = funciones.MostrarClientes

Listado.DataSource = vista

Listado.Columns(0).Visible = False

Listado.ClearSelection()

End Sub

Private Sub btnEditar\_Click(sender As Object, e As EventArgs) Handles btnEditar.Click

If Listado.SelectedRows.Count > 0 Then

FCliente.txtId.Text = Me.Listado.SelectedCells.Item(0).Value

FCliente.txtNombre.Text = Me.Listado.SelectedCells.Item(1).Value

FCliente.txtApellido.Text = Me.Listado.SelectedCells.Item(2).Value

FCliente.txtDireccion.Text = Me.Listado.SelectedCells.Item(3).Value

FCliente.txtTelefono.Text = Me.Listado.SelectedCells.Item(4).Value

FCliente.txtRFC.Text = Me.Listado.SelectedCells.Item(5).Value

FCliente.ShowDialog()

mostrarcliente()

Else

MsgBox("Seleccione un Registro, Por favor", MsgBoxStyle.Exclamation, "Editar Cliente")

End If

End Sub

Private Sub Listado\_CellDoubleClick(sender As Object, e As DataGridViewCellEventArgs) Handles Listado.CellDoubleClick

If txtflag.Text = "1" Then

FVenta.lblidcliente.Text = Me.Listado.SelectedCells.Item(0).Value

FVenta.txtdoc.Text = Me.Listado.SelectedCells.Item(5).Value

FVenta.txtcliente.Text = Me.Listado.SelectedCells.Item(1).Value & " " & Me.Listado.SelectedCells.Item(2).Value

txtflag.Text = "0"

Me.Close()

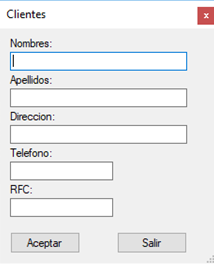
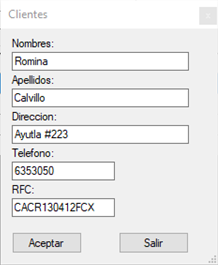
End If

End Sub

End Class

**Form Agregar y Editar**

Este es un solo formulario, pero sirve para agregar o editar un cliente dependiendo de que botón hayamos presionado en el listado de cliente. Cabe aclara que para editar primero se tiene que elegir un cliente del listado que se muestra en la tabla (DataGridview)

**Agregar**

Public Class FCliente

Dim valores As New LCliente

Dim funciones As New DCliente

Public Sub AgregarCliente()

Try

valores.\_nombre = txtNombre.Text

valores.\_RFC = txtRFC.Text

valores.\_apellido = txtApellido.Text

valores.\_direccion = txtDireccion.Text

valores.\_telefono = Convert.ToInt32(txtTelefono.Text)

If funciones.AgregarCliente(valores) Then

MsgBox("Cliente Agregado Correctamente", MsgBoxStyle.Information, "Registro Clientes")

Else

MsgBox("Cliente No Agregado Correctamente", MsgBoxStyle.Exclamation, "Registro Clientes")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

**Editar**

Public Sub EditarCliente()

Try

valores.\_idcliente = txtId.Text

valores.\_RFC = txtRFC.Text

valores.\_nombre = txtNombre.Text

valores.\_apellido = txtApellido.Text

valores.\_direccion = txtDireccion.Text

valores.\_telefono = Convert.ToInt32(txtTelefono.Text)

If funciones.EditarCliente(valores) Then

MsgBox("Cliente Editado Correctamente", MsgBoxStyle.Information, "Registro Clientes")

Else

MsgBox("Cliente No Editado Correctamente", MsgBoxStyle.Exclamation, "Registro Clientes")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnAceptar\_Click(sender As Object, e As EventArgs) Handles btnAceptar.Click

If txtApellido.Text = "" Or txtNombre.Text = "" Or txtDireccion.Text = "" Or txtRFC.Text = "" Or txtTelefono.Text = "" Then

MsgBox("Complete Todos Los Datos Porfavor", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Exit Sub

End If

If txtId.Text <> "" Then

If MsgBox("¿Desesa Guardar los Cambios?", MsgBoxStyle.Information + MsgBoxStyle.YesNo, "Mensaje del Sistema") = MsgBoxResult.No Then

Exit Sub

End If

EditarCliente()

Else

AgregarCliente()

End If

txtId.Clear()

FListadoClientes.mostrarcliente()

Me.Close()

End Sub

Private Sub btnSalir\_Click(sender As Object, e As EventArgs) Handles btnSalir.Click

Me.Close()

End Sub

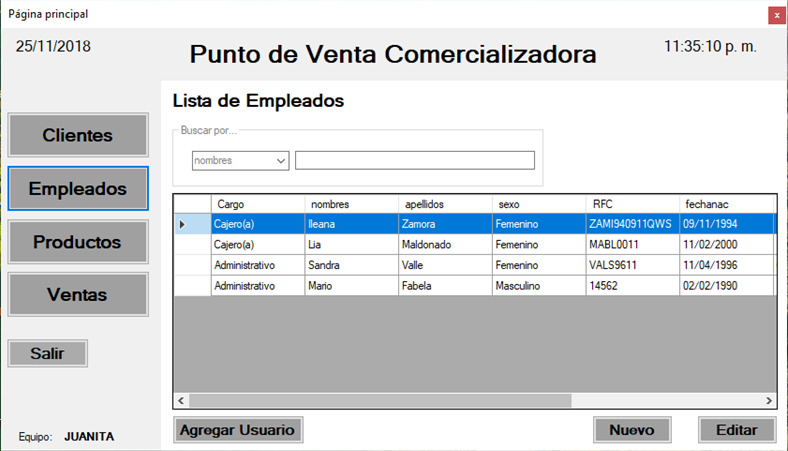
Private Sub FCliente\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

**Form de Empleados**

Aquí se muestra el listado de empleados, el cual sale cuando se presiona el botón empleados. Se observan dos botones uno de nuevo empleado y otro de editar empleado.



Public Class FListadoEmpleados

Private Sub FListadoEmpleados\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

mostrarempleados()

End Sub

Dim valores As New LEmpleados

Dim funciones As New DEmpleado

Dim vista As New DataView

Dim funcionesusuario As New DUsuarios

Dim valoresusuario As New LUsuario

Private Sub txtbuscar\_TextChanged(sender As Object, e As EventArgs) Handles txtbuscar.TextChanged

Try

vista.RowFilter = cbobuscar.Text & " like '" & txtbuscar.Text & "%'"

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnNuevo\_Click(sender As Object, e As EventArgs) Handles btnNuevo.Click

Dim ventana As New FEmpleado

ventana.ShowDialog()

Me.mostrarempleados()

End Sub

Public Sub mostrarempleados()

vista = funciones.MostrarEmpleado.DefaultView

Listado.DataSource = funciones.MostrarEmpleado

Listado.DataSource = vista

Listado.Columns(0).Visible = False

Listado.Columns(1).Visible = False

Listado.ClearSelection()

End Sub

Private Sub btnEditar\_Click(sender As Object, e As EventArgs) Handles btnEditar.Click

If Listado.SelectedRows.Count > 0 Then

FEmpleado.txtid.Text = Me.Listado.SelectedCells.Item(0).Value

FEmpleado.idcargo = Convert.ToInt32(Me.Listado.SelectedCells.Item(1).Value)

FEmpleado.txtnombre.Text = Me.Listado.SelectedCells.Item(3).Value

FEmpleado.txtapellido.Text = Me.Listado.SelectedCells.Item(4).Value

FEmpleado.cbosexo.Text = Me.Listado.SelectedCells.Item(5).Value

FEmpleado.txtrfc.Text = Me.Listado.SelectedCells.Item(6).Value

FEmpleado.txtfechanacimiento.Text =

Else

MsgBox("Seleccione un Registro, Por favor", MsgBoxStyle.Exclamation, "Editar Cliente")

End If

End Sub

Private Sub btnasignar\_Click(sender As Object, e As EventArgs) Handles btnasignar.Click

Try

Dim ventana As New FUsuario

If Listado.SelectedRows.Count > 0 Then

valoresusuario.\_idempleado = Convert.ToInt32(Me.Listado.SelectedCells.Item(0).Value)

If funcionesusuario.VerificarUsuario(valoresusuario) Then

MsgBox("Ya Existe un Usuario Creado", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Exit Sub

End If

ventana.txtid.Text = Me.Listado.SelectedCells.Item(0).Value

ventana.txtnya.Text = Me.Listado.SelectedCells.Item(3).Value & " " & Me.Listado.SelectedCells.Item(4).Value

ventana.txtcargo.Text = Me.Listado.SelectedCells.Item(2).Value

ventana.txtdni.Text = Me.Listado.SelectedCells.Item(6).Value

ventana.ShowDialog()

Else

MsgBox("Seleccione un Registro, Por favor", MsgBoxStyle.Exclamation, "Asignar Usuario")

End If

Catch ex As Exception

End Try

End Sub

Private Sub Listado\_CellDoubleClick(sender As Object, e As DataGridViewCellEventArgs) Handles Listado.CellDoubleClick

If txtflag.Text = "1" then

FVenta.lblidempleado.Text = Me.Listado.SelectedCells.Item(0).Value

FVenta.txtcargo.Text = Me.Listado.SelectedCells.Item(2).Value

FVenta.txtnya.Text = Me.Listado.SelectedCells.Item(3).Value & " " & Me.Listado.SelectedCells.Item(4).Value

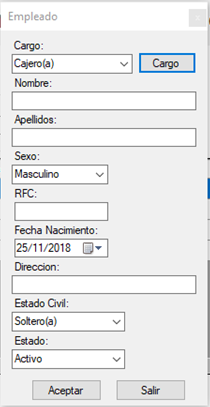
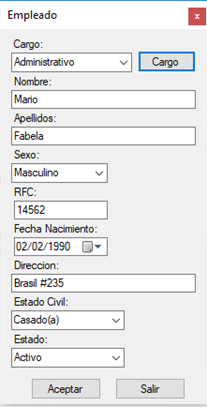
Me.txtflag.Text = "0"

Me.Close()

End If End Sub End Class

**Form Agregar y Editar**

Este formulario es para agregar o editar a un empleado dependiendo que botón de haya presionado en el listado de empleado.

Public Class FEmpleado

Public idcargo As Integer

Dim valores As New LEmpleados

Dim funciones As New DEmpleado

Dim funcionesca As New DCargo

Private Sub FEmpleado\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

cargarcargos()

End Sub

**Agregar**

Public Sub AgregarEmpleado()

Try

valores.\_idcargo = Convert.ToInt32(cbocargo.SelectedValue.ToString)

valores.\_nombre = txtnombre.Text

valores.\_apellidos = txtapellido.Text

valores.\_sexo = cbosexo.Text

valores.\_RFC = txtrfc.Text

valores.\_fechanacimiento = txtfechanacimiento.Text

valores.\_direccion = txtdireccion.Text

valores.\_estadocivil = cboestadocivil.Text

valores.\_estado = cboestado.Text

If funciones.AgregarEmpleado(valores) Then

MsgBox("Empleado Agregado Correctamente", MsgBoxStyle.Information, "Registro Empleado")

Else

MsgBox("Empleado No Agregado Correctamente", MsgBoxStyle.Exclamation, "Registro Empleado")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

**Editar**

Public Sub EditarEmpleado()

Try

valores.\_idempleado = Convert.ToInt32(txtid.Text)

valores.\_idcargo = Convert.ToInt32(cbocargo.SelectedValue.ToString)

valores.\_nombre = txtnombre.Text

valores.\_apellidos = txtapellido.Text

valores.\_sexo = cbosexo.Text

valores.\_RFC = txtrfc.Text

valores.\_fechanacimiento = txtfechanacimiento.Text

valores.\_direccion = txtdireccion.Text

valores.\_estadocivil = cboestadocivil.Text

valores.\_estado = cboestado.Text

If funciones.EditarEmpleado(valores) Then

MsgBox("Empleado Editado Correctamente", MsgBoxStyle.Information, "Registro Empleado")

Else

MsgBox("Empleado No Editado Correctamente", MsgBoxStyle.Exclamation, "Registro Empleado")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnAceptar\_Click(sender As Object, e As EventArgs) Handles btnAceptar.Click

If txtapellido.Text = "" Or txtnombre.Text = "" Or txtdireccion.Text = "" Or txtrfc.Text = "" Then

MsgBox("Complete Todos Los Datos Porfavor", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Exit Sub

End If

If txtid.Text <> "" Then

If MsgBox("¿Desesa Guardar los Cambios?", MsgBoxStyle.Information + MsgBoxStyle.YesNo, "Mensaje del Sistema") = MsgBoxResult.No Then

Exit Sub

End If

EditarEmpleado()

Else

AgregarEmpleado()

End If

txtid.Clear()

FListadoEmpleados.mostrarempleados()

Me.Close()

End Sub

Public Sub cargarcargos()

Try

cbocargo.DataSource = funcionesca.MostrarCargo

cbocargo.DisplayMember = "descripcion"

cbocargo.ValueMember = "id\_cargo"

If idcargo <> 0 Then

cbocargo.SelectedValue = idcargo

End If

idcargo = 0

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnSalir\_Click(sender As Object, e As EventArgs) Handles btnSalir.Click

Me.Close()

End Sub

Private Sub btncargo\_Click(sender As Object, e As EventArgs) Handles btncargo.Click

Dim ventana As New FCargo

ventana.ShowDialog()

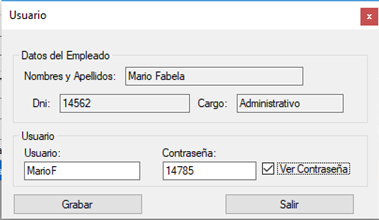
cargarcargos()

End Sub

End Class

**Form de Agregar Usuario**

Este form se utiliza para crearle una cuenta en el sistema a algún empleado ya sea nuevo o uno que ya tenga tiempo pero no tuviera una usuario. Para acceder a este formulario solo se tiene que presionar el botón de agregar usuario que se encuentra en el form de listado de empleados.



Public Class FUsuario

Dim valores As New LUsuario

Dim funciones As New DUsuarios

Private Sub chkver\_CheckedChanged(sender As Object, e As EventArgs) Handles chkver.CheckedChanged

If chkver.CheckState = CheckState.Checked Then

txtcontraseña.PasswordChar = ""

Else

txtcontraseña.PasswordChar = "•"

End If

End Sub

Private Sub btbgrabar\_Click(sender As Object, e As EventArgs) Handles btbgrabar.Click

If Trim(txtusuario.Text) = "" Or Trim(txtcontraseña.Text) = "" Then

MsgBox("Ingrese todos los datos porfavor", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Return

End If

Try valores.\_idempleado = Convert.ToInt32(txtid.Text)

valores.\_usuario = txtusuario.Text

valores.\_contraseña = txtcontraseña.Text

If funciones.AgregarUsuario(valores) Then

MsgBox("Usuario Agregado Correctamente", MsgBoxStyle.Information, "Registro de Usuario")

Else

MsgBox("Usuario No Agregado Correctamente", MsgBoxStyle.Information, "Registro de Usuario")

End If

txtusuario.Text = ""

txtcontraseña.Text = ""

Me.Close()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnsalir\_Click(sender As Object, e As EventArgs) Handles btnsalir.Click

Me.Close( End Sub

Private Sub FUsuario\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

**Form de Productos**

Presionando la opción de productos se puede ver el listado de productos, aquí también se pueden observar los botones de nuevo producto y editar producto.



Public Class FListadoProductos

Dim valores As New LProductos

Dim funciones As New DProductos

Dim vista As New DataView

Private Sub FListadoProductos\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

mostrarproductos()

End Sub

Private Sub txtbuscar\_TextChanged(sender As Object, e As EventArgs) Handles txtbuscar.TextChanged

Try

vista.RowFilter = cbobuscar.Text & " like '" & txtbuscar.Text & "%'"

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnNuevo\_Click(sender As Object, e As EventArgs) Handles btnNuevo.Click

Dim ventana As New FProducto

ventana.ShowDialog()

mostrarproductos()

End Sub

Public Sub mostrarproductos()

vista = funciones.MostrarProductos.DefaultView

Listado.DataSource = funciones.MostrarProductos

Listado.DataSource = vista

Listado.Columns(0).Visible = False

Listado.Columns(1).Visible = False

Listado.ClearSelection()

End Sub

Private Sub btnEditar\_Click(sender As Object, e As EventArgs) Handles btnEditar.Click

Try

If Listado.Rows.Count = 0 Then

MsgBox("No existen Registros Aún", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Return

End If

If Listado.SelectedRows.Count > 0 Then

FProducto.cargarcategorias()

FProducto.txtid.Text = Me.Listado.SelectedCells.Item(0).Value

FProducto.idcategoria = Convert.ToInt32(Me.Listado.SelectedCells.Item(1).Value)

FProducto.txtnombre.Text = Me.Listado.SelectedCells.Item(2).Value

FProducto.txtdescripcion.Text = Me.Listado.SelectedCells.Item(4).Value

FProducto.txtstock.Text = Me.Listado.SelectedCells.Item(5).Value

FProducto.txtpreciocompra.Text = Me.Listado.SelectedCells.Item(6).Value

FProducto.txtprecioventa.Text = Me.Listado.SelectedCells.Item(7).Value

FProducto.txtfechavencimiento.Text = Me.Listado.SelectedCells.Item(8).Value

FProducto.ShowDialog()

mostrarproductos()

Else

MsgBox("Seleccione un Registro, Por favor", MsgBoxStyle.Exclamation, "Editar Producto")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub Listado\_CellDoubleClick(sender As Object, e As DataGridViewCellEventArgs) Handles Listado.CellDoubleClick

If lblflag.Text = "1" Then

FVenta.lblidproducto.Text = Me.Listado.SelectedCells.Item(0).Value

FVenta.txtproducto.Text = Me.Listado.SelectedCells.Item(2).Value

FVenta.txtdescripcion.Text = Me.Listado.SelectedCells.Item(4).Value

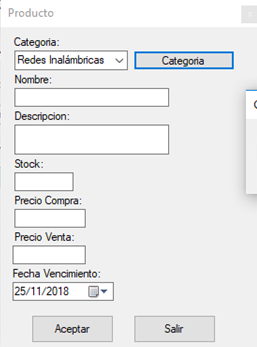
Me.Close()

End If

End Sub

**Form Agregar y Editar**

Este form nos actualiza o nos agrega a un producto.



Imports System.Data.SqlClient

Public Class FProducto

Public idcategoria As Integer

Dim valores As New LProductos

Dim funcionespro As New DProductos

Dim funcionescat As New DCategoria

Private Sub FProducto\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

cargarcategorias()

End Sub

**Agregar**

Public Sub AgregarProducto()

Try

valores.\_idcategoria = Convert.ToInt32(cbocategoria.SelectedValue.ToString)

valores.\_nombre = txtnombre.Text

valores.\_descripcion = txtdescripcion.Text

valores.\_stock = Convert.ToInt32(txtstock.Text)

valores.\_preciocompra = Convert.ToDecimal(txtpreciocompra.Text)

valores.\_precioventa = Convert.ToDecimal(txtprecioventa.Text)

If funcionespro.AgregarProducto(valores) Then

MsgBox("Producto Agregado Correctamente", MsgBoxStyle.Information, "Registro de Producto")

Else

MsgBox("Producto No Agregado Correctamente", MsgBoxStyle.Information, "Registro de Producto")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

**Editar**

Public Sub EditarProducto()

Try

valores.\_idproducto = Convert.ToInt32(txtid.Text)

valores.\_idcategoria = Convert.ToInt32(cbocategoria.SelectedValue.ToString)

valores.\_nombre = txtnombre.Text

valores.\_descripcion = txtdescripcion.Text

valores.\_stock = Convert.ToInt32(txtstock.Text)

valores.\_preciocompra = Convert.ToDecimal(txtpreciocompra.Text)

valores.\_precioventa = Convert.ToDecimal(txtprecioventa.Text)

If funcionespro.EditarProducto(valores) Then

MsgBox("Producto Editado Correctamente", MsgBoxStyle.Information, "Registro de Producto")

Else

MsgBox("Producto No Editado Correctamente", MsgBoxStyle.Information, "Registro de Producto")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub cargarcategorias()

Try

cbocategoria.DataSource = funcionescat.MostrarCategoria

cbocategoria.DisplayMember = "descripcion"

cbocategoria.ValueMember = "id\_categoria"

If idcategoria <> 0 Then

cbocategoria.SelectedValue = idcategoria

End If

idcategoria = 0

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnaceptar\_Click(sender As Object, e As EventArgs) Handles btnaceptar.Click

If txtnombre.Text = "" Or txtdescripcion.Text = "" Or txtstock.Text = "" Or txtpreciocompra.Text = "" Or txtprecioventa.Text = "" Then

MsgBox("Complete Todos Los Datos Porfavor", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Exit Sub

End If

If txtid.Text <> "" Then

If MsgBox("¿Desesa Guardar los Cambios?", MsgBoxStyle.Information + MsgBoxStyle.YesNo, "Mensaje del Sistema") = MsgBoxResult.No Then

Exit Sub

End If

EditarProducto()

Else

AgregarProducto()

End If

Me.txtid.Text = ""

FListadoProductos.mostrarproductos()

Me.Close()

End Sub

Private Sub btnCategoria\_Click(sender As Object, e As EventArgs) Handles btnCategoria.Click

Dim ventana As New FCategoria

ventana.ShowDialog()

cargarcategorias()

End Sub

Private Sub btnsalir\_Click(sender As Object, e As EventArgs) Handles btnsalir.Click

Me.Close()

End Sub

Private Sub Button1\_Click\_1(sender As Object, e As EventArgs)

cbocategoria.SelectedValue = 7

End Sub

End Class

**Form Venta**

En este form se visualiza el listado de ventas, aquí podemos ver dos botones, pero uno es de actualizar, que sirve para actualizar la tabla de ventas, y el otro es de nueva venta.



Public Class FListadoVentas

Private Sub FListadoVentas\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MostrarVenta()

End Sub

Dim valores As New LDetalleVenta

Dim funciones As New DVenta

Dim vista As New DataView

Private Sub txtbuscar\_TextChanged(sender As Object, e As EventArgs) Handles txtbuscar.TextChanged

Try

vista.RowFilter = cbobuscar.Text & " like '" & txtbuscar.Text & "%'"

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Sub MostrarVenta()

vista = funciones.MostrarVenta.DefaultView

Listado.DataSource = funciones.MostrarVenta

Listado.DataSource = vista

Listado.Columns(0).Visible = False

Listado.ClearSelection()

End Sub

Private Sub btnEditar\_Click(sender As Object, e As EventArgs) Handles btnEditar.Click

ReporteVenta.ShowDialog()

End Sub

Private Sub Listado\_CellDoubleClick(sender As Object, e As DataGridViewCellEventArgs) Handles Listado.CellDoubleClick

Try

Dim ventana As New FComprobante

ventana.idventa = Convert.ToInt32(Me.Listado.SelectedCells.Item(0).Value)

ventana.ShowDialog()

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnnuevo\_Click(sender As Object, e As EventArgs) Handles btnnuevo.Click

Try

FVenta.Show()

MostrarVenta()

Catch ex As Exception

End Try

End Sub

Private Sub btnactualizar\_Click(sender As Object, e As EventArgs) Handles btnactualizar.Click

MostrarVenta()

End Sub

Private Sub cbobuscar\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cbobuscar.SelectedIndexChanged

If cbobuscar.SelectedIndex = 0 Then

txtbuscar.MaxLength = 7

Else

txtbuscar.MaxLength = 100

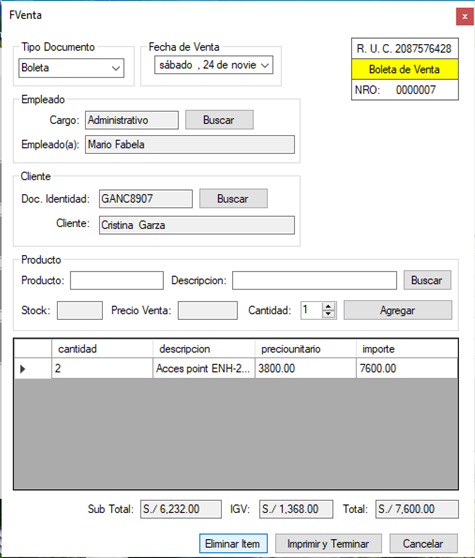
End If

End Sub

End Class

**Form Nueva Venta**

Este form se abre al momento de presionar el botón de nueva venta, aquí se ingre el tipo de dcumento que se desea, luego se agregan los datos del empleado que esta atendiendo, y tambein se agregan los datos del ciente y por ultimo se agregan o se buscan los datos de los productos que se desean agregar.



Public Class FVenta

Dim registroventa As Integer = 0

Dim valoresventa As New LVenta

Dim valoresdetalleventa As New LDetalleVenta

Dim funcionesventa As New DVenta

Dim funcionesdetalleventa As New DDetalleVenta

Private Sub FVenta\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

lblnrodoc.Text = funcionesventa.ObtenerNroDocumento

LimpiarTexbox()

Ordenarylimpiar()

End Sub

Private Sub btnbuscarE\_Click(sender As Object, e As EventArgs) Handles btnbuscarE.Click

Dim venta As New FListadoEmpleados

venta.txtflag.Text = 1

venta.ShowDialog()

End Sub

Private Sub btnbuscarC\_Click(sender As Object, e As EventArgs) Handles btnbuscarC.Click

Dim ventana As New FListadoClientes

ventana.txtflag.Text = 1

ventana.ShowDialog()

End Sub

Private Sub btnbuscarP\_Click(sender As Object, e As EventArgs) Handles btnbuscarP.Click

Dim ventana As New FListadoProductos

ventana.lblflag.Text = 1

ventana.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedDialog

ventana.ShowDialog()

End Sub

Private Sub btnagregar\_Click(sender As Object, e As EventArgs) Handles btnagregar.Click

Try

If registroventa = 0 Then

If AgregarVenta() Then

If AgregarDetalleVenta() Then

registroventa = 1

VerRegistroVenta()

Ordenarylimpiar()

Else

MsgBox("Falta Ingresar Datos del Producto", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

End If

Else

MsgBox("Falta Ingresar Datos Cliente o Empleado", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

End If

Else

If AgregarDetalleVenta() Then

VerRegistroVenta()

Ordenarylimpiar()

Else

MsgBox("Falta Ingresar Datos del Producto", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

End If

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Public Function AgregarVenta() As Boolean

Try

If lblidempleado.Text <> "0" And lblidcliente.Text <> "0" And txtcliente.Text <> "" Then

valoresventa.\_idempleado = lblidempleado.Text

valoresventa.\_idcliente = lblidcliente.Text

valoresventa.\_nrodocumento = lblnrodoc.Text

valoresventa.\_tipodocumento = cbotipodocumento.Text

valoresventa.\_fechaventa = txtfecha.Text

If funcionesventa.AgregarVenta(valoresventa) Then

txtidventa.Text = (funcionesdetalleventa.ObtenerIdVenta).ToString

Return True

Else

Return False

End If

Else

Return False

End If

Catch ex As Exception

MsgBox(ex.Message)

Return False

End Try

End Function

Public Function AgregarDetalleVenta() As Boolean

Try

If txtidventa.Text <> "0" And lblidproducto.Text <> "0" Then

valoresdetalleventa.\_idproducto = Convert.ToInt32(lblidproducto.Text)

valoresdetalleventa.\_idventa = Convert.ToDecimal(txtidventa.Text)

valoresdetalleventa.\_cantidad = Convert.ToInt32(txtcantidad.Value)

valoresdetalleventa.\_preciounitario = Convert.ToDecimal(txtprecio.Text)

If funcionesdetalleventa.AgregarDetalleVenta(valoresdetalleventa) Then

If funcionesdetalleventa.Disminuirstock(valoresdetalleventa) Then

Return True

Else

Return False

End If

Else

Return False

End If

Else

Return False

End If

Catch ex As Exception

MsgBox(ex.Message)

Return False

End Try

End Function

Public Sub VerRegistroVenta()

valoresdetalleventa.\_idventa = Convert.ToInt32(txtidventa.Text)

listado.DataSource = funcionesdetalleventa.VerRegistroDetalleVenta(valoresdetalleventa)

listado.Columns(0).Visible = False

listado.Columns(1).Visible = False

listado.ClearSelection()

txtsubtotal.Text = "S./ " & FormatNumber((0.82 \* Sumar("importe", listado)), 2)

txtigv.Text = "S./ " & FormatNumber((0.18 \* Sumar("importe", listado)), 2)

txttotal.Text = "S./ " & FormatNumber(Sumar("importe", listado), 2).ToString

End Sub

Public Sub LimpiarTexbox()

txtnya.Clear()

txtcargo.Clear()

txtcliente.Clear()

txtdoc.Clear()

txtsubtotal.Clear()

txtigv.Clear()

txttotal.Clear()

End Sub

Public Sub Ordenarylimpiar()

Try

lblidproducto.Text = "0"

txtproducto.Text = ""

txtdescripcion.Text = ""

txtstock.Text = ""

txtprecio.Text = ""

txtcantidad.Value = 1

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Sub btnregistrar\_Click(sender As Object, e As EventArgs) Handles btnregistrar.Click

If Trim(txtidventa.Text) = "0" Then

MsgBox("La venta no Tiene registros aún", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

End If

Comprobante.txtidventa2.Text = Me.txtidventa.Text

Comprobante.ShowDialog()

End Sub

Private Sub btneliminar\_Click(sender As Object, e As EventArgs) Handles btneliminar.Click

Try

If listado.Rows.Count = 0 Then

MsgBox("No existen Registros Aún", MsgBoxStyle.Exclamation, "Mensaje del Sistema")

Return

End If

If listado.SelectedRows.Count > 0 Then

valoresdetalleventa.\_iddetalleventa = Me.listado.SelectedCells.Item(0).Value

valoresdetalleventa.\_idproducto = Me.listado.SelectedCells.Item(1).Value

valoresdetalleventa.\_cantidad = Me.listado.SelectedCells.Item(2).Value

If funcionesdetalleventa.Aumentarstock(valoresdetalleventa) And funcionesdetalleventa.EliminarDetalleVenta(valoresdetalleventa) Then

VerRegistroVenta()

Else

MsgBox("No se Puede Eliminar el Registro")

End If

Else

MsgBox("Seleccione un Registro porfavor", MsgBoxStyle.Information, "Mensaje del Sistema")

End If

Catch ex As Exception

MsgBox(ex.Message)

End Try

End Sub

Private Function Sumar(ByVal nombre\_Columna As String, ByVal Dgv As DataGridView) As Double

Dim Total As Double = 0

' recorrer las filas y obtener los items de la columna indicada en "nombre\_Columna"

Try

If Dgv.RowCount > 0 Then

For i As Integer = 0 To Dgv.RowCount - 1

Total = Total + CDbl(Dgv.Item(nombre\_Columna.ToLower, i).Value)

Next

Return Total

Else

Return Nothing

End If

Catch ex As Exception

MsgBox(ex.Message.ToString)

Return Nothing

End Try

End Function

Private Sub btnsalir\_Click(sender As Object, e As EventArgs) Handles btnsalir.Click

Try

If Trim(txtidventa.Text) = "0" Then

Me.Close()

Return

End If

valoresdetalleventa.\_idventa = Convert.ToInt32(txtidventa.Text)

If funcionesventa.DeshacerVenta(valoresdetalleventa) Then

For Each row As DataGridViewRow In listado.Rows

Dim idproducto As Integer = Convert.ToInt32(row.Cells("id\_producto").Value)

Dim cantidad As Integer = Convert.ToInt32(row.Cells("cantidad").Value)

valoresdetalleventa.\_idproducto = idproducto

valoresdetalleventa.\_cantidad = cantidad

If funcionesdetalleventa.Aumentarstock(valoresdetalleventa) Then

End If

Next

Me.Close()

Else

MsgBox("Error al eliminar fuera", MsgBoxStyle.Information, "Mensaje del Sistema")

End If

Catch ex As Exception

End Try

End Sub

End Class

**PROCEDIMIENTOS DE BASE DE DATOS**

**PROCEDIMIENTOS PARA LOS CLIENTES**

create proc AgregarCliente

@RFC varchar(13),

@apellido nvarchar(50),

@nombre nvarchar(50),

@direccion nvarchar(50),

@telefono int

as

begin

insert into Cliente(RFC,apellidos,nombre,direccion,telefono) values(@RFC,@apellido,@nombre,@direccion,@telefono)

end

create proc EditarCliente

@id\_cliente int,

@RFC varchar(13),

@apellido nvarchar(50),

@nombre nvarchar(50),

@direccion nvarchar(50),

@telefono int

as

begin

update Cliente set RFC = @RFC, apellidos = @apellido, nombre = @nombre, direccion = @direccion, telefono = @telefono

where id\_cliente = @id\_cliente

end

create proc MostrarCliente

as

begin

select id\_cliente,nombre,apellidos,direccion,telefono,RFC from Cliente

end

**PROCEDIMIENTOS PARA LAS CATEGORIAS**

create proc AgregarCategoria

@descripcion nvarchar(50)

as

begin

insert into Categoria(descripcion) values (@descripcion)

end

create proc MostrarCategoria

as

begin

select id\_categoria,descripcion from Categoria

end

**PROCEDIMIENTOS PARA LOS PRODUCTOS**

create proc MostrarProductos

as

begin

select PR.id\_producto,PR.id\_categoria,PR.nombre as producto,CA.descripcion as categoria, PR.descripcion, PR.stock, PR.preciocompra, PR.precioventa from Producto PR

inner join Categoria CA on CA.id\_categoria = PR.id\_categoria

end

create proc AgregarProducto

@id\_categoria int,

@nombre nvarchar(50),

@descripcion nvarchar(50),

@stock int,

@preciocompra decimal(10,2),

@precioventa decimal(10,2)

as

begin

insert into Producto(id\_categoria,nombre,descripcion,stock,preciocompra,precioventa) values(@id\_categoria,@nombre,@descripcion,@stock,@preciocompra,@precioventa)

end

create proc EditarProducto

@id\_producto int,

@id\_categoria int,

@nombre nvarchar(50),

@descripcion nvarchar(50),

@stock int,

@preciocompra decimal(10,2),

@precioventa decimal(10,2)

as

begin

update Producto set id\_categoria = @id\_categoria, nombre = @nombre,descripcion = @descripcion,stock = @stock,preciocompra = @preciocompra,precioventa = @precioventa

where id\_producto = @id\_producto

end

**PROCEDIMIENTOS PARA LOS CARGOS (de los empleados)**

create proc MostrarCargo

as

begin

select id\_cargo, descripcion from Cargo

end

create proc AgregarCargo

@descripcion nvarchar(50)

as

begin

insert into Cargo(descripcion) values (@descripcion)

end

**PROCEDIMIENTOS PARA LOS EMPLEADOS**

Create proc MostrarEmpleados

as

begin

select EM.id\_empleado, EM.id\_cargo,CA.descripcion as Cargo,EM.nombres,EM.apellidos,EM.sexo,EM.RFC,EM.fechanac,EM.direccion,EM.estadocivil as 'estado civil',EM.estado from Empleado EM

inner Join Cargo CA on CA.id\_cargo = EM.id\_cargo

end

create proc AgregarEmpleado

@id\_cargo int,

@nombre nvarchar(50),

@apellidos nvarchar(50),

@sexo nvarchar(50),

@RFC varchar(13),

@fechanac nvarchar(20),

@direccion nvarchar(50),

@estadocivil nvarchar(30),

@estado nvarchar(30)

as

begin

insert into Empleado(id\_cargo,nombres,apellidos,sexo,RFC,fechanac,direccion,estadocivil,estado) values(@id\_cargo,@nombre,@apellidos,@sexo,@RFC,@fechanac,@direccion,@estadocivil,@estado)

end

create proc EditarEmpleado

@id\_empleado int,

@id\_cargo int,

@nombre nvarchar(50),

@apellidos nvarchar(50),

@sexo nvarchar(50),

@RFC varchar(13),

@fechanac nvarchar(20),

@direccion nvarchar(50),

@estadocivil nvarchar(30),

@estado nvarchar(30)

as

begin

update Empleado set id\_cargo = @id\_cargo, nombres = @nombre, apellidos = @apellidos, sexo = @sexo, RFC = @RFC, fechanac = @fechanac, direccion = @direccion, estadocivil = @estadocivil,estado= @estado where id\_empleado = @id\_empleado

end

**PROCEDIMIENTOS PARA LOS USUARIOS**

create proc AccederUsuario

@usuario nvarchar(50),

@contraseña nvarchar(50)

as

begin

select \* from Usuario where usuario = @usuario and contraseña = @contraseña

end

create proc AgregarUsuario

@id\_empleado int,

@usuario nvarchar(50),

@contraseña nvarchar(50)

as

begin

insert into Usuario(id\_empleado,usuario,contraseña) values (@id\_empleado,@usuario,@contraseña)

end

Create Proc VerificarUsuario

@id\_empleado int

as

begin

select \* from Usuario where id\_empleado = @id\_empleado

end

**PROCEDIMIENTOS ALMACENADOS DE VENTAS**

create proc ObtenerNroDocumento

as

begin

SELECT TOP 1 nrodocumento FROM Venta ORDER BY id\_venta DESC

end

Create proc MostrarVenta

as

begin

select VE.id\_venta,VE.nrodocumento as 'Nro.Documento',(EM.nombres + ' ' + EM.apellidos) as Empleado,(CL.nombre + ' ' + Cl.apellidos) as Cliente, VE.tipodocumento as 'Tipo de Documeto', VE.fechaventa as 'Fecha Venta' from Venta VE

inner join Empleado EM on EM.id\_empleado = VE.id\_empleado

inner join Cliente CL on CL.id\_cliente = VE.id\_cliente

end

create proc AgregarVenta

@id\_empleado int,

@id\_cliente int,

@nrodocumento nvarchar(50),

@tipodocumento nvarchar(50),

@fechaventa nvarchar(30)

as

begin

insert into Venta(id\_empleado,id\_cliente,nrodocumento,tipodocumento,fechaventa) values(@id\_empleado,@id\_cliente,@nrodocumento,@tipodocumento,@fechaventa)

end

create proc DeshacerVenta

@id\_venta int

as

begin

delete from Venta where id\_venta = @id\_venta

end

**PROCEDIMIENTOS ALMACENADOS DE DETALLES DE VENTAS**

create proc ObtenerIdVenta

as

begin

SELECT TOP 1 id\_venta FROM Venta ORDER BY id\_venta DESC

end

create proc VerRegistroDetalleVenta

@id\_venta int

as

begin

select DV.id\_detalleventa,DV.id\_producto,DV.cantidad,(PR.nombre + ' ' + PR.descripcion)as descripcion,DV.preciounitario,(DV.cantidad \* DV.preciounitario) as importe from Detalleventa DV

inner join Producto PR on PR.id\_producto = DV.id\_producto where DV.id\_venta =@id\_venta

end

create proc AgregarDetalleVenta

@id\_producto as int,

@id\_venta as int,

@cantidad as int,

@preciounitario as decimal (10,2)

as

insert into Detalleventa (id\_producto, id\_venta, cantidad, preciounitario)

values (@id\_producto, @id\_venta, @cantidad, @preciounitario)

create proc GenerarComprobante

@id\_venta int

as

begin

SELECT dbo.Venta.id\_venta, dbo.Cliente.nombre, dbo.Cliente.apellidos, dbo.Cliente.RFC, dbo.Venta.fechaventa, dbo.Venta.tipodocumento, dbo.Venta.nrodocumento,

dbo.Producto.nombre AS Descripcion, dbo.Detalleventa.cantidad, dbo.Detalleventa.preciounitario,

dbo.Detalleventa.cantidad \* dbo.Detalleventa.preciounitario AS Importe

FROM dbo.Detalleventa INNER JOIN

dbo.Venta ON dbo.Detalleventa.id\_venta = dbo.Venta.id\_venta INNER JOIN

dbo.Producto ON dbo.Detalleventa.id\_producto = dbo.Producto.id\_producto INNER JOIN

dbo.Cliente ON dbo.Venta.id\_cliente = dbo.Cliente.id\_cliente

where dbo.Venta.id\_venta = @id\_venta

end

create proc AgregarDetalleVenta

@id\_producto int,

@id\_venta int,

@cantidad int,

@preciounitario decimal(10,2)

as

begin

insert into Detalleventa(id\_producto,id\_venta,cantidad,preciounitario) values (@id\_producto,@id\_venta,@cantidad,@preciounitario)

end

create proc DeshacerDetalleVenta

@id\_venta int

as

begin

delete from Detalleventa where id\_venta = @id\_venta

end

select \* from Producto

drop proc AgregarDetalleVenta

create proc Aumentarstock

@id\_producto int,

@cantidad int

as

begin

update Producto set stock = stock + @cantidad where id\_producto = @id\_producto

end

create proc EliminarDetalleVenta

@id\_detalleventa int

as

begin

delete from Detalleventa where id\_detalleventa = @id\_detalleventa

end

create proc Disminuirstock

@id\_producto int,

@cantidad int

as

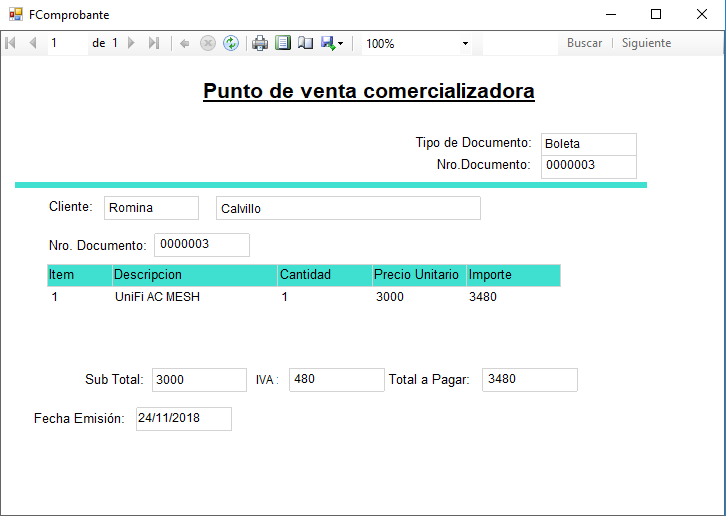
begin

update Producto set stock = stock - @cantidad where id\_producto = @id\_producto

end

**REPORTES**

Este se genera al momento de darle al botón de terminar del form venta.

****

create proc AgregarDetalleVenta

@id\_producto as int,

@id\_venta as int,

@cantidad as int,

@preciounitario as decimal (10,2)

as

insert into Detalleventa (id\_producto, id\_venta, cantidad, preciounitario)

values (@id\_producto, @id\_venta, @cantidad, @preciounitario)

create proc GenerarComprobante

@id\_venta int

as

begin

SELECT dbo.Venta.id\_venta, dbo.Cliente.nombre, dbo.Cliente.apellidos, dbo.Cliente.RFC, dbo.Venta.fechaventa, dbo.Venta.tipodocumento, dbo.Venta.nrodocumento,

dbo.Producto.nombre AS Descripcion, dbo.Detalleventa.cantidad, dbo.Detalleventa.preciounitario,

dbo.Detalleventa.cantidad \* dbo.Detalleventa.preciounitario AS Importe

FROM dbo.Detalleventa INNER JOIN

dbo.Venta ON dbo.Detalleventa.id\_venta = dbo.Venta.id\_venta INNER JOIN

dbo.Producto ON dbo.Detalleventa.id\_producto = dbo.Producto.id\_producto INNER JOIN dbo.Cliente ON dbo.Venta.id\_cliente = dbo.Cliente.id\_cliente

where dbo.Venta.id\_venta = @id\_venta

**REFERENCIAS**

**Sistema de ventas que es:**

[**https://www.monografias.com/trabajos109/monografia-sistema-ventas/monografia-sistema-ventas.shtml**](https://www.monografias.com/trabajos109/monografia-sistema-ventas/monografia-sistema-ventas.shtml)

**Conectar un nuevo origen de datos:**

[**https://msdn.microsoft.com/es-es/library/tzedkwye.aspx**](https://msdn.microsoft.com/es-es/library/tzedkwye.aspx)

**Crear una conexión con una clase en vb:**

[**https://csrlima.wordpress.com/2012/03/17/clase-conexion-visual-basic-net-a-sql-server/**](https://csrlima.wordpress.com/2012/03/17/clase-conexion-visual-basic-net-a-sql-server/)

**Hacer reportes en reportviewer:**

[**https://www.youtube.com/watch?v=lA4p3vxiBoM**](https://www.youtube.com/watch?v=lA4p3vxiBoM)