

UNJBG ITEL CARRERA Técnico Analista Programador de Sistemas	¡LÍDER EN CAPACITACIÓN INFORMÁTICA!! Garantía del proceso Enseñanza-Aprendizaje con las últimas tecnologías, con computadoras de última generación, impresoras, escáner, multimedia, redes, Internet, material didáctico paso a paso, biblioteca y aula virtual con docentes del más alto nivel.	CURSO Programación Visual .NET II
--	--	--------------------------------------

GUÍA DE LABORATORIO N° 03

OBJETIVOS:

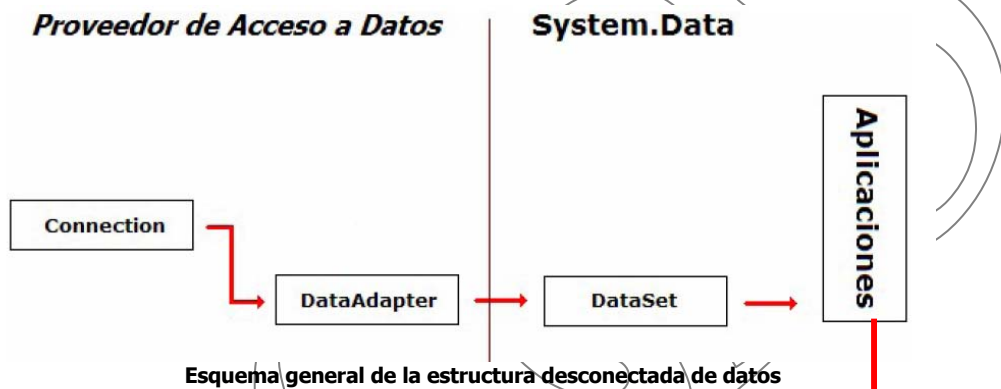
- Crear aplicaciones para actualizar y/o mantenimiento de tablas, realizando la misma sin el asistente o añadiendo objetos, todo con código para evitar las propiedades de enlace y trabajar en un ambiente desconectado

1. ENTORNO INTEGRADO DE DESARROLLO

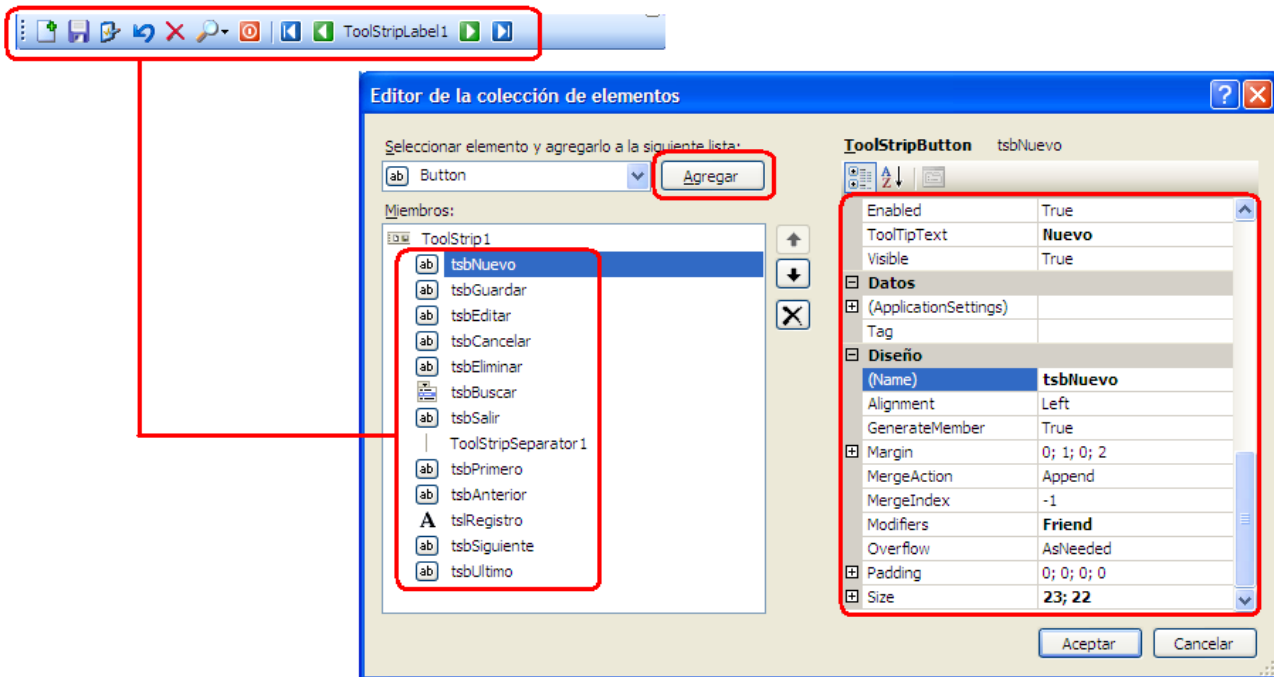
- Proceda a ingresar Visual Studio .NET y realice lo siguiente:
- Deberá Abrir la solución realizada en la práctica anterior (**SisVentas**)

2. CREACION DE FORMULARIO DE Mantenimiento de Empleados (frmActualizarEmpleado.vb)

- 2.1. Agregue un nuevo formulario a su proyecto y guárdelo con el nombre de **frmActualizarEmpleado.vb**
- 2.2. Proceda a agregar los siguientes objetos según se observa a continuación:



- 2.3. Agregando los botones del **ToolStrip** "Barra de Herramientas" para actualización y Desplazamiento
- Para ello deberá agregar un componente **ToolStrip**
 - Luego haga clic derecho sobre el y seleccione **Editar Elementos...**
 - Nos visualizará la Ventana siguiente, en ella deberás agregar los elementos "**miembros del ToolStrip**", para ello solo deberás hacer clic en el botón **Agregar** cuantas veces sea necesarios según la imagen.
 - Una vez que haz agregado todos los miembros del **ToolStrip**, procede a establecer las propiedades **Name**, **ToolTipText** e **Image** para c/u de los elementos.
 - El **Name** a establecer a cada elemento es el que se aprecia en la imagen



2.4. Ahora procederemos a establecer las propiedades de los demás objetos

Objeto	Propiedad	Valor Asignado
Form1	Name BackgroundImage Icon StartPosition	frmActualizarEmpleado Haga clic en ... y Seleccione la imagen que Ud. Desea Haga clic en ... y busque un icono de su agrado CenterScreen
Label3, label4....	Text	Agregar los textos respectivos en cada Etiqueta
TextBox1	Name	TxtCodEmp
TextBox2	Name	TxtNomEmp
TextBox3	Name	TxtApeEmp
TextBox4	Name	TxtDirEmp
GroupBox2	Name Text	grbsexo Sexo
RadioButton1	Name Text	RbtMasculino Masculino
RadioButton2	Name Text	RbtFemenino Femenino
TextBox5	Name	TxtTelEmp
Combo1	Name Text	CboCargo
Button1	Name Text	btnNuevoCargo Nuevo Cargo...

Label1	Name	LblActivo
TextBox6	Name	txtObservacion
Button2	Name Text	BtnAgregarFoto Agregar Foto...
Label10	Name	lblRutaFoto

2.5. Establecer el tamaño adecuado a cada control u objeto que esta dentro del formulario.

2.6. A todas las etiquetas establecer la propiedad Autosize = True

3. CREAR PROCEDIMIENTOS ALMACENADOS (ABRIR EL ANALIZADOR DE CONSULTAS)

Ingresar al Analizador de Consultas y Crear los siguientes **Procedimientos Almacenados** de mantenimiento de Datos:

Procedimiento para Insertar Registros:

```

CREATE PROCEDURE AgregarEmpleado
(
@codemp varchar(5),
@nomemp varchar(20),
@apeemp varchar(20),
@diremp varchar(25),
@sexemp bit,
@telemp varchar(8),
@codcar varchar(5),
@activo bit,
@observacion varchar(80),
@foto varchar(50)
)
AS
begin transaction
INSERT EMPLEADO(codemp,nomemp,apeemp,diremp,sexemp,telemp,codcar,activo,observacion,foto)
VALUES
(@codemp,@nomemp,@apeemp,@diremp,@sexemp,@telemp,@codcar,@activo,@observacion,@foto)
if @@error=0
    commit transaction
else
    rollback transaction
    
```

Para agregarlo a la BD: Sombree el Procedimiento y pulse F5, o clic en el botón ejecutar.

Procedimiento para Desactivar un Empleado:

```

CREATE PROCEDURE DesactivaEmpleado
(
@codemp varchar(5),
@activo bit
)
AS
BEGIN TRANSACTION
UPDATE EMPLEADO
SET
codemp=@codemp,
activo=@activo
WHERE codemp=@codemp
IF @@ERROR=0
    COMMIT TRANSACTION
ELSE
    ROLLBACK TRANSACTION
    
```

Procedimiento para Eliminar Registros: (Sólo de referencia)

```
CREATE PROCEDURE EliminarEmpleado
@codemp varchar(5)
AS
BEGIN TRANSACTION
DELETE FROM EMPLEADO WHERE codemp=@codemp
IF @@ERROR=0
    COMMIT TRANSACTION
ELSE
    ROLLBACK TRANSACTION
```

Procedimiento para Modificar Registros :

```
CREATE PROCEDURE ModificarEmpleado
(
@codemp varchar(5),
@nomemp varchar(20),
@apeemp varchar(20),
@diremp varchar(25),
@sexemp bit,
@telemp varchar(8),
@codcar varchar(5),
@activo bit,
@observacion varchar(100),
@foto varchar(50)
)
AS
BEGIN TRANSACTION
UPDATE EMPLEADO
SET
codemp=@codemp,
nomemp=@nomemp,
apeemp=@apeemp,
diremp=@diremp,
sexemp=@sexemp,
telemp=@telemp,
codcar=@codcar,
activo=@activo,
observacion=@observacion,
foto=@foto
WHERE codemp=@codemp
IF @@ERROR=0
    COMMIT TRANSACTION
ELSE
    ROLLBACK TRANSACTION
```

4. CODIFICACIÓN DEL FORMULARIO DE ACTUALIZACIÓN DE EMPLEADOS

No olvide que para este formulario solo se trabaja con Procedimientos Almacenados para realizar el mantenimiento de los registros de Empleados.

AHORA PROCEDEREMOS A ESCRIBIR EL CODIGO CORRESPONDIENTE:

Agregar la Biblioteca de Clases SQLClient: Sirve para poder utilizar los objetos SqlConnection, SqlCommand, SqlDataAdapter

Biblioteca de Clase:

```
Option Compare Text
Imports System.Data.SqlClient
Imports System.Data
```

En la Sección de Declaraciones Generales escriba:

```
Dim dacargo As SqlDataAdapter
Dim objDataAdapter As SqlDataAdapter
Dim objDataSet As New DataSet
Dim objDataRow As DataRow
Dim posicion, flag, resultado As Integer
```

Private Sub frmActualizarEmpleado_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

```
Conexion.Open()
objDataAdapter = New SqlDataAdapter("SELECT * FROM EMPLEADO WHERE
activo=1", Conexion)
objDataAdapter.Fill(objDataSet, "EMPLEADO")
Conexion.Close()
llenarcombos(False)
posicion = (objDataSet.Tables("EMPLEADO").Rows.Count - 1)
CargarDatos()
lblRutaFoto.Visible = False
HabilitarControles(False)
HabilitarBotones(True)
'Estableciendo etiquetas a las cajas
txtCodEmp.Tag = "Código" : txtNomEmp.Tag = "Nombres"
txtApeEmp.Tag = "Apellidos" : txtDirEmp.Tag = "dirección"
txtTelEmp.Tag = "Teléfono" : txtObservacion.Tag = "Observacion"
```

End Sub

Procedimientos de Usuario

Private Sub llenarcombos(ByVal cargo As Boolean)

```
Conexion.Open()
desconectarcombos()
dacargo = New SqlDataAdapter("SELECT * FROM cargo", ModConexion.Conexion)
If cargo = False Then
    dacargo.Fill(objDataSet, "cargo")
Else
    objDataSet.Tables("cargo").Clear()
    dacargo.Fill(objDataSet, "cargo")
    cargo = False
End If
Conexion.Close()
cboCargo.DataBindings.Add(New
System.Windows.Forms.Binding("SelectedValue", objDataSet, "Empleado.codcar"))
cboCargo.DataSource = Me.objDataSet.Tables("Cargo")
cboCargo.DisplayMember = "descar"
cboCargo.ValueMember = "codcar"
cboCargo.DropDownStyle = ComboBoxStyle.DropDownList
' si es que se ha agregado un nuevo cargo, lo seleccionamos
If cargo = False Then cboCargo.SelectedIndex = cboCargo.Items.Count - 1
```

End Sub

Private Sub CargarDatos()

```
Dim valor As Boolean
Try
    objDataRow = objDataSet.Tables("EMPLEADO").Rows(posicion)
    txtCodEmp.Text = objDataRow("codemp")
    txtNomEmp.Text = objDataRow("nomemp")
    txtApeEmp.Text = objDataRow("apeemp")
    txtDirEmp.Text = objDataRow("diremp")
    valor = objDataRow("sexemp")
```

```
If valor Then
    rbtMasculino.Checked = True
    rbtFemenino.Checked = False
Else
    rbtMasculino.Checked = False
    rbtFemenino.Checked = True
End If
txtTelEmp.Text = objDataRow("telemp")
cboCargo.Selectedvalue = objDataRow("codcar")
txtObservacion.Text = objDataRow("observacion").ToString
If objDataRow("foto").ToString = Nothing Then
    PictureBox1.Image = Nothing
    lblRutaFoto.Text = Nothing
Else
    lblRutaFoto.Text = objDataRow("foto")
End If
If objDataRow("activo") = True Then
    lblActivo.Text = "Activo" : lblActivo.ForeColor = Color.Blue
End If
If objDataRow("activo") = False Then
    lblActivo.Text = "Inactivo" : lblActivo.ForeColor = Color.Red
End If
tslRegistro.Text = "Registro: " & posicion + 1 & " de " &
    objDataSet.Tables("EMPLEADO").Rows.Count

    actualizafoto()
Catch exc As Exception
    MessageBox.Show(exc.Message)
End Try
End Sub
```

Private Sub HabilitarControles (ByVal est As Boolean)

```
Dim ctrl As Control
For Each ctrl In Me.Controls
    If TypeOf ctrl Is TextBox Then ctrl.Enabled = est
Next
txtCodEmp.ReadOnly = True
cboCargo.Enabled = est
grbSexo.Enabled = est
```

End Sub

Private Sub HabilitarBotones (ByVal est As Boolean)

```
Dim oitem As Windows.Forms.ToolStripItem
For Each oitem In ToolStrip1.Items
    If TypeOf oitem Is ToolStripButton Then
        oitem.Enabled = est
    End If
Next
tsbGuardar.Enabled = Not est
tsbCancelar.Enabled = Not est
btnAgregarFoto.Visible = Not est
btnNuevoCargo.Visible = Not est
```

End Sub

Function verificavacios () As Boolean

```
Dim objetos As Object
For Each objetos In Me.Controls
    If TypeOf objetos Is TextBox Then
        If objetos.name <> "txtTelEmp" And objetos.name <> "txtObservacion" And
            objetos.TextLength = 0 Then
            MsgBox("Debe ingresar: " & objetos.tag)
            verificavacios = True
            Exit Function
        End If
    End If
End For
```

<pre>Next If cboCargo.SelectedIndex < 0 Then MsgBox("Debe seleccionar un cargo") verificavacios = True End If End Function</pre>	
<pre>Private Sub GeneraCodigo() Dim codi As String Dim cmd As New SqlCommand("select max(codemp) from empleado",Conexion) Conexion.Open() codi = cmd.ExecuteScalar Conexion.Close() Me.txtCodEmp.Text = Format(Val(codi) + 1, "00000") End Sub</pre>	
<pre>Private Sub desconectarcombos() cboCargo.DataBindings.Clear() End Sub</pre>	<pre>` IMPORTANTE: ESTE PROC. DEBE SER CREADO EN EL MÓDULO Public Sub LimpiarTextBox(ByVal ofrm As Form) For Each oControl As Control In ofrm.Controls If TypeOf oControl Is TextBox Then oControl.Text = "" End If Next End Sub</pre>
<pre>Private Sub LimpiarControles() LimpiarTextBox(Me) PictureBox1.Image = Nothing End Sub</pre>	
<pre>Private Sub Desplazar(ByVal op As Integer) Select Case op Case 1 posicion = 0 CargarDatos() Case 2 If posicion = 0 Then MessageBox.Show("Primer Registro", "", 0, MessageBoxIcon.Information) Else posicion -= 1 CargarDatos() End If Case 3 If posicion = (objDataSet.Tables("EMPLEADO").Rows.Count - 1) Then MessageBox.Show("Último Registro", "", 0, MessageBoxIcon.Information) Else posicion += 1 CargarDatos() End If Case 4 posicion = (objDataSet.Tables("EMPLEADO").Rows.Count - 1) CargarDatos() End Select End Sub</pre>	
<pre>Private Sub actualizafoto() Dim mifoto As Bitmap If objDataRow("foto").ToString = Nothing Then PictureBox1.Image = Nothing Else mifoto = New Bitmap(Me.lblRutaFoto.Text) PictureBox1.Image = CType(mifoto, Image) End If End Sub</pre>	

```

Private Sub AgregaCargo(ByVal codi As String, ByVal nombre As String)
    'Agregando el nuevo registro a la tabla cargo
    Dim objcomando As New SqlCommand("INSERT INTO Cargo (codcar,descar) VALUES
                                     (@codcar,@descar)",Conexion)

    'Parámetros
    objcomando.Parameters.Add(New SqlParameter("@codcar", SqlDbType.VarChar))
    objcomando.Parameters("@codcar").Value = codi
    objcomando.Parameters.Add(New SqlParameter("@descar", SqlDbType.VarChar))
    objcomando.Parameters("@descar").Value = nombre
    Conexion.Open()
    objcomando.ExecuteNonQuery()
    ModConexion.Conexion.Close()
    llenarcombos(True)
End Sub

Private Sub Editar()
    flag = 2
    HabilitarControles(True)
    HabilitarBotones(False)
    txtNomEmp.Focus()
End Sub

Private Sub Actualizar()
    'Para actualizar el DataSet
    objDataSet.Clear()
    objDataAdapter.Fill(objDataSet, "EMPLEADO")
End Sub

Private Sub Nuevo()
    flag = 1
    lblRutaFoto.Text = Nothing
    HabilitarControles(True)
    HabilitarBotones(False)
    LimpiarControles()
    GeneraCodigo()
End Sub

Private Sub Cancelar()
    Conexion.Open()
    objDataSet.RejectChanges()
    Conexion.Close()
    HabilitarControles(False)
    HabilitarBotones(True)
    Desplazar(4)
End Sub

Private Sub ToolStrip1_ItemClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles ToolStrip1.ItemClicked
    Select Case e.ClickedItem.Name
        ' Actualización
        Case "tsbnuevo"      ' Botón Nuevo
            Nuevo()
        Case "tsbguardar"   ' Botón Guardar
            Guardar()
        Case "tsbeditar"    ' Botón Editar
            Editar()
        Case "tsbcancelar"  ' Botón Cancelar
            Cancelar()
        Case "tsbeliminar"  ' Botón Desactivar Empleado
            Dim rpta As MsgBoxResult
            rpta = MsgBox("Esta seguro que desea Desactivar Al Empleado : " &
                Chr(13) & Chr(10) & txtCodEmp.Text & " - " & txtNomEmp.Text & " " &
                txtApeEmp.Text, MsgBoxStyle.YesNo + MsgBoxStyle.Question, "Confirmación")
            If rpta = vbYes Then DesactivaEmpleado(txtCodEmp.Text, 0)
        Case "tsbsalir"     ' Botón Salir
            Me.Close()
        ' Desplazamiento
        Case "tsbprimero"   ' Ir a al Primero
            Desplazar(1)
        Case "tsbanterior"  ' Ir al Anterior
            Desplazar(2)
        Case "tsbsiguiente" ' Ir al Siguiente
            Desplazar(3)
        Case "tsbultimo"   ' Ir al Último
            Desplazar(4)
    End Select
End Sub
    
```


Private Sub Guardar()

```
If verificavacios() = True Then Exit Sub
Dim objComando As New SqlCommand
Dim oper As String, valor, activo As Boolean
objComando.Connection = ModConexion.Conexion
objComando.CommandType = CommandType.StoredProcedure
If flag = 1 Then 'Agregar
    objComando.CommandText = "AgregarEmpleado"
    oper = " insertados : "
Else ' Editar
    objComando.CommandText = "modificarEmpleado"
    oper = " modificados : "
End If
Try
    'Agregando Parametros
    objComando.Parameters.Add(New SqlParameter("@codemp", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@nomemp", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@apeemp", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@diremp", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@sexemp", SqlDbType.Bit))
    objComando.Parameters.Add(New SqlParameter("@telemp", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@codcar", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@activo", SqlDbType.Bit))
    objComando.Parameters.Add(New SqlParameter("@observacion", SqlDbType.VarChar))
    objComando.Parameters.Add(New SqlParameter("@foto", SqlDbType.VarChar))
    'asignando valores a parametros
    If rbtMasculino.Checked = True Then
        valor = True
    Else
        valor = False
    End If
    activo = True ' Todos los empleados nuevos son activos, ¿activar un empleado?
    objComando.Parameters("@codemp").Value = txtCodEmp.Text
    objComando.Parameters("@nomemp").Value = txtNomEmp.Text
    objComando.Parameters("@apeemp").Value = txtApeEmp.Text
    objComando.Parameters("@diremp").Value = txtDirEmp.Text
    objComando.Parameters("@sexemp").Value = valor
    objComando.Parameters("@telemp").Value = txtTelEmp.Text
    objComando.Parameters("@codcar").Value = cboCargo.SelectedValue
    objComando.Parameters("@activo").Value = activo
    objComando.Parameters("@observacion").Value = txtObservacion.Text
    objComando.Parameters("@foto").Value = lblRutaFoto.Text
    Conexion.Open()
    resultado = objComando.ExecuteNonQuery
    Actualizar()
    Conexion.Close()
    posicion = (objDataSet.Tables("EMPLEADO").Rows.Count - 1)
    MessageBox.Show("Registros " & oper & resultado, "", 0,
    MessageBoxIcon.Information)
    Catch exc As Exception
        MessageBox.Show(exc.Message)
    End Try
    llenarcombos(False)
    Desplazar(4)
    HabilitarControles(False)
    HabilitarBotones(True)
End Sub
```

```
Private Sub DesactivaEmpleado(ByVal codi As String, ByVal estado As Boolean)
    Dim objComando As New SqlCommand
    objComando.Connection = ModConexion.Conexion
    objComando.CommandType = CommandType.StoredProcedure
    objComando.CommandText = "DesactivaEmpleado"
    Try
        'Agregando Parametros
        objComando.Parameters.Add(New SqlParameter("@codemp", SqlDbType.VarChar))
        objComando.Parameters.Add(New SqlParameter("@activo", SqlDbType.Bit))
        'asignando valores a parametros
        objComando.Parameters("@codemp").Value = codi
        objComando.Parameters("@activo").Value = estado
        Conexion.Open()
        resultado = objComando.ExecuteNonQuery
        Actualizar()
        Conexion.Close()
        MessageBox.Show(resultado & " Empleado Desactivado", "", 0,
        MessageBoxIcon.Information)
        Desplazar(1)
    Catch exc As Exception
        MessageBox.Show(exc.Message)
    End Try
    llenarcombos(False)
    HabilitarControles(False)
    HabilitarBotones(True)
End Sub

Private Sub btnAgregarFoto_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAgregarFoto.Click

    Dim mifoto As Bitmap
    Dim curfilename As String = ""
    Dim opendirg As OpenFileDialog = New OpenFileDialog
    opendirg.Filter = "All Bitmapfiles|*.*"
    Dim filter As String = opendirg.Filter
    opendirg.Title = "open a bitmap file"
    If opendirg.ShowDialog() = Windows.Forms.DialogResult.OK Then
        curfilename = opendirg.FileName
        lblRutaFoto.Text = curfilename
        mifoto = New Bitmap(curfilename)
        PictureBox1.Image = CType(mifoto, Image)
    Else
        Exit Sub
    End If
End Sub

Private Sub btnNuevoCargo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNuevoCargo.Click

    'Generando código para el nuevo cargo
    Dim codi As String
    Dim cmd As New SqlCommand("select max(codcar) from cargo", ModConexion.Conexion)
    Conexion.Open()
    codi = cmd.ExecuteScalar
    codi = Format(Val(codi) + 1, "00000")
    Conexion.Close()
    ' Solicitando el nombre del nuevo cargo
    Dim nombre As String
    nombre = InputBox("Ingrese Nombre de nuevo cargo:", "Agregando Nuevo Cargo")
    If nombre.Length < 4 Then
        MsgBox("Debe ingresar mínimo 4 caracteres", MsgBoxStyle.Critical, "Error")
        Exit Sub
    Else

```

```
Dim rptA As MsgBoxResult
rptA = MsgBox("Esta seguro que desea agregar el nuevo cargo : " & _
Chr(13) & Chr(10) & "codigo: " & codi & Chr(13) & Chr(10) & _
"Nombre: " & nombre, MsgBoxStyle.YesNo + MsgBoxStyle.Question, "Confirmación")
If rptA = vbYes Then
    AgregaCargo(codi, nombre)
Else
    Exit Sub
End If
End If
End Sub

Private Sub tsbBuscarbasico_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tsbBuscarbasico.Click, tsbBuscarAvanzado.Click
Select Case sender.name
Case "tsbbuscarbasico"
    Dim strDato As String
    Dim fila As Integer
    Dim dv As New DataView
    strDato = InputBox("Ingrese codigo del Empleado: ", "Buscar")
    dv.Table = objDataSet.Tables("Empleado")
    dv.Sort = "codemp"
    fila = dv.Find(strDato)
    If fila = -1 Then
        MessageBox.Show("Codigo del Empleado no Existe", "Cuidado")
        Exit Sub
    End If
    posicion = fila
    CargarDatos()
Case "tsbbuscaravanzado"
    MessageBox.Show("Proximamente será implementado", "Busqueda Avanzada",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
End Select
End Sub
```

TAREA ENCARGADA: / /

- En base al presente formulario, proceda a realizar los formularios correspondientes para Clientes, Artículos y Categoría. Así mismo deberá enlazarlos c/u a la opción de Menú que corresponda.
- Haga los procedimientos necesarios para poder visualizar a los empleados **Inactivos**, y poder activar a uno de ellos según sea el caso.

J
L
P
S