

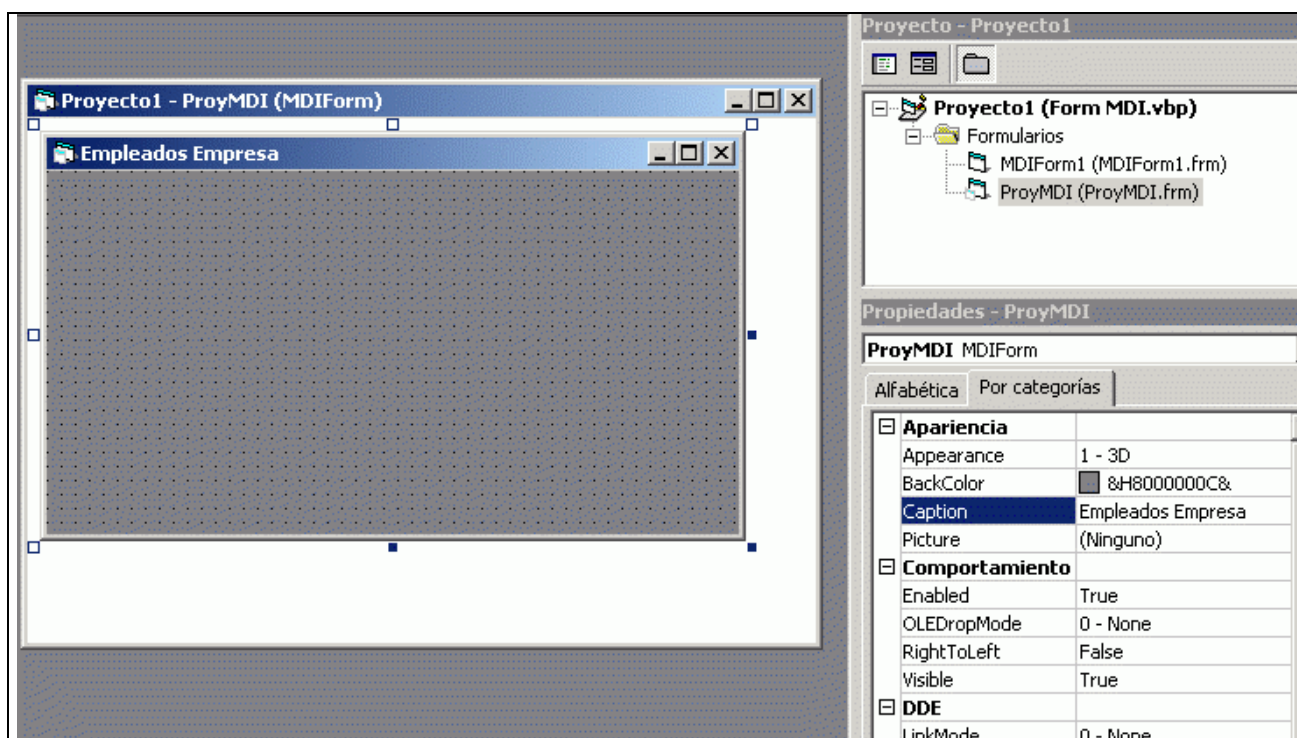
**VISUAL BASIC – Diseño de Formularios MDI y Menús - Proyectos Aplica****Formularios MDI**

Un formulario **MDI (Multiple Document Interface)** es un contenedor de otros formularios. La aplicación MDI contendrá un formulario padre MDI y los formularios hijos MDI. Un ejemplo de aplicación que se basa en este concepto es Excel, Adobe Reader también.

Un formulario MDI (padre) se crea mediante la **selección de MDI Form del menú Project**. Los formularios hijos a su vez tienen que tener asignado **True en la propiedad MDI Child**.

Una aplicación sólo puede tener un formulario padre; y un formulario padre sólo puede contener objetos de menú, cuadros de dibujo, barras de herramientas, barras de estado etc. Es decir, es en el formulario MDI donde se diseña el menú principal y las barras de herramientas de la aplicación para llamar así a los formularios hijos que son los que hacen el trabajo.

Si un formulario hijo tiene menús, la barra de menú del formulario hijo reemplaza automáticamente la barra del menú del objeto MDI mientras esté activo. Es decir que en tiempo de diseño el menú aparece en el formulario hijo pero en tiempo de ejecución se desplegará sobre el formulario padre.

**Creación de Menús**

Los menús se crean mediante el *Editor de Menú* que es una aplicación de diseño de menú que se incluye en Visual Basic y con el cual pueden crearse menús personalizados.

Puede seleccionarse de varias formas. Una de las más habituales es desde el Menú Herramientas (Tools)/Editor de Menús o también colocándose sobre el formulario y pulsando el botón derecho del ratón.

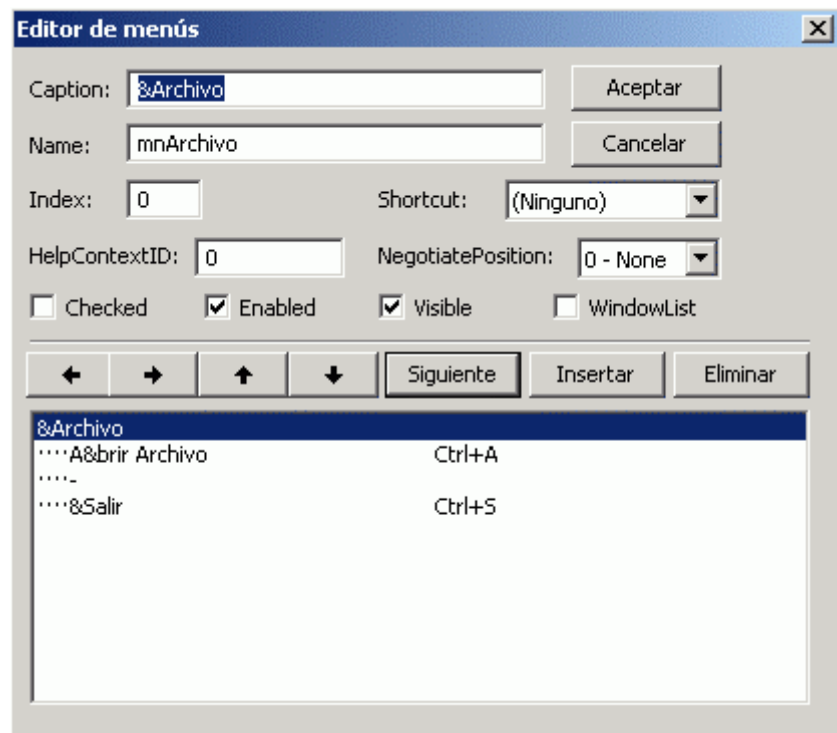
La siguiente imagen muestra el Editor de Menú. Como puede apreciarse se pueden ingresar el título de cada elemento de menú, un nombre de menú, un Index que ya vamos a explicar para qué sirve, teclas de modo abreviado, etc. (Véanse páginas 65/67 de Aprenda Visual Basic 6.0 como si estuviera en primero de la Universidad de Navarra)

La siguiente imagen nos muestra el Editor de Menús:

Se ha creado una barra de menú, el menú Archivo que por ahora tiene dos submenú: Abrir Archivo y Salir. Además con el símbolo – se dibujará una línea separadora.

Para Abrir Archivo se utilizará como teclas rápidas la combinación ctrl.+A y para Salir ctrl. +S

Los botones de esquema (flechas) los hemos usado para ajustar la jerarquía y también hemos utilizado el Index de cada submenú para poder manejar el código con mayor facilidad



El **Index sirve para establecer un Array de Menús**. Esto se hace para poder referenciar a los distintos títulos de menú en forma simple sin necesidad de tener 10 procedimientos distintos para el evento clic si tenemos diez títulos de menú. De esta manera tendremos un solo procedimiento o método clic para todos los títulos de menú que serán referenciados mediante el Index.

¿Cómo se hace? Pues para cualquier título de menú tendremos el mismo nombre de menú.

Tanto el manejo de formularios MDI como de Menús se explican y muestran en **los proyectos Aplica** desde el Aplica1 hasta el Aplica2 que son sencillos pero muestran con claridad cómo debe diseñarse tanto un conjunto de formularios MDI como un Menú para que sea fácil de programar y de mantener en el tiempo. Los ejemplos han sido realizados teniendo en cuenta una secuencia didáctica coherente al grado de complejidad que se les ha ido introduciendo.

En el último proyecto se ha utilizado el concepto de reusabilidad del código para poder llevar a cabo las ABM de la tabla Cargos.

### Proyecto Aplica

El proyecto Aplica es muy simple. Tenemos un formulario padre que se llama InicioAplica y cinco formularios hijos. En el formulario padre está definido el menú principal que sólo tiene títulos de menú de rango superior y lo hemos llamado **mnuEmpre ya que es un arreglo de menú**. De esta forma tenemos 5 valores posibles para diseñar el código del evento clic del menú (que es el único evento que utiliza el menú). Cuando el usuario pulsa el título Sistema se muestra el formulario hijo Empres0, cuando selecciona el título Cargos se muestra el formulario Emprs3 y así siguiendo. El menú está deshabilitado inicialmente. En las opciones de proyecto hemos establecido que el primer formulario que se ejecuta es Empres0. Si lo ejecutamos y cerramos la ventana ahí se habilitan todos los títulos de menú.

Este diseño no es óptimo puesto que tiene por fin sólo mostrar cómo se trabaja con los títulos de menú. No cuenta con un subsistema de seguridad mínimo y podemos tener todas las ventanas abiertas al mismo tiempo, lo cual tampoco es práctico ni bueno desde un punto de vista de diseño de sistemas.

#### Código del formulario InicioAplica:

```
Option Explicit
Dim permiso As Integer
Dim usuario As Integer
```

Por ahora no se van a utilizar.

```
Private Sub MDIForm_Load()
```

```
End Sub
```

```
Private Sub mnuEmpre_Click(Index As Integer)
```

```
Dim i As Integer
Select Case (Index)
Case 0: 'submenu sistema
    Empres0.Show
Case 1:
    Emprs3.Show
Case 2:
    Emprs2.Show
Case 3:
    Emprs1.Show
Case 4:
    Emprs4.Show
End Select
End Sub
```

El método Show muestra el formulario según el elemento de menú que se seleccione

Además cada formulario hijo tiene el siguiente código:

```
Option Explicit

Private Sub Form_Unload(Cancel As Integer)
Dim i As Integer
For i = 0 To InicioAplica.mnuEmpre.Count - 1
    InicioAplica.mnuEmpre(i).Enabled = True
Next i
End Sub
```

Cuando se carga el formulario todos los elementos de menú se establecen como habilitados

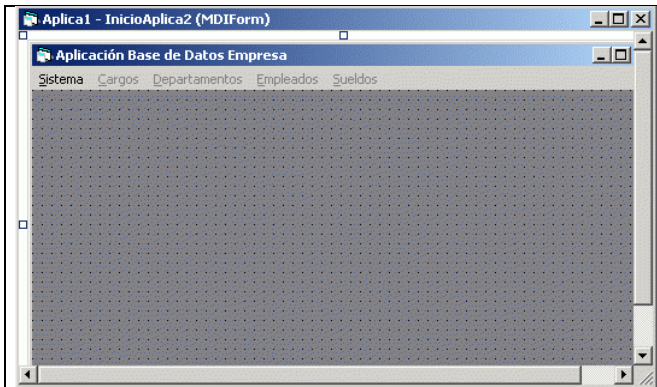
### Proyecto Aplica1

Este proyecto es más elaborado. En primer lugar el Menú también está deshabilitado inicialmente para todos los títulos salvo el de Sistema que además inicia con la ventana que permite validar el usuario y contraseña (formulario hijo Empresa0). En una palabra el formulario principal se llama IniciaAplica2 y tiene 5 formularios hijos, como en el caso anterior. El primer formulario se llama Empresa0 y permite que el usuario ingrese su nombre de usuario y su contraseña, que valida el sistema. Para ello se ha agregado una nueva tabla en la base de datos que contemple a los usuarios y sus claves. Asimismo el usuario puede modificar la clave.

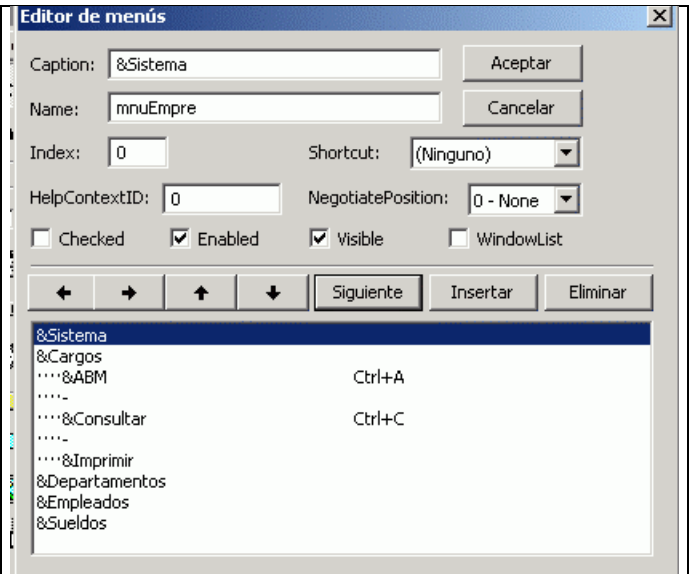
Una vez que el sistema validó el usuario en forma correcta se cierra el formulario hijo y se habilitan todos los títulos de menú.

El diseño del Menú principal es similar al caso anterior salvo que el título de menú Cargos tiene un submenú

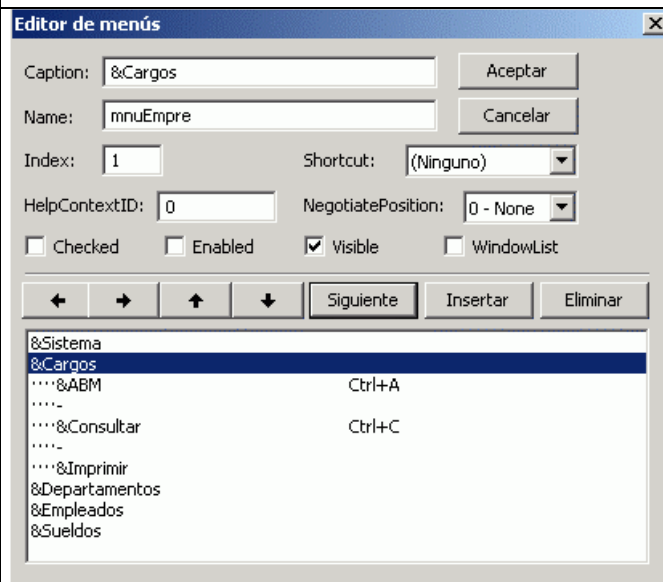
que también es un arreglo de menú.



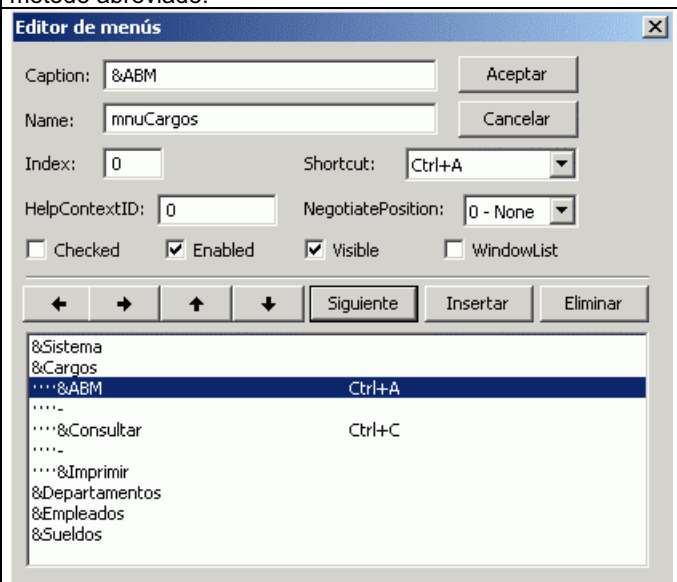
Puede apreciarse que sólo el título Sistema está habilitado.



Opciones de configuración del título Sistema: primer elemento del array de menú (index = 0), habilitado y visible. Como es un título de nivel superior no puede tener ninguna tecla de método abreviado.



Es la segunda opción del array de menu llamado mnuEmpre.



Es la primera opción del menú mnuCargos que es a su vez un array de menu.

### Código del formulario InicioAplica2:

Option Explicit  
Dim gcValido As Boolean

Public Sub habilitar(valor As Boolean)  
gcValido = valor  
End Sub

Private Sub MDIForm\_Load()

End Sub

```
Private Sub mnuEmpre_Click(Index As Integer)
```

```
If (gcValido) Then
Select Case (Index)
Case 0: 'submenu sistema
    Empresa0.Show
Case 1: 'submenú cargos
    'Empresa3.Show
```

Variable gcValido: si se ha validado el nombre de usuario y contraseña cuando el usuario pulse sobre uno de los títulos se mostrará el formulario correspondiente

```
Case 2: 'submenu departamentos
    Empresa2.Show
Case 3: 'submenu empleados
    Empresa1.Show
Case 4: ' submenu sueldos
    Empresa4.Show
End Select
End If
End Sub
```

```
Private Sub mnuCargos_click(Index As Integer)
```

```
Select Case (Index) 'el índice del submenu
Case 0: Empresa3.Show
Case 2: MsgBox ("Todavía no está realizado el formulario de consultas")
Case 4: MsgBox ("Todavía no está realizado el formulario de impresión")
End Select
End Sub
```

#### Código del formulario empresa0:

```
Option Explicit
Dim dbBase As Database
Dim rsUsuarios As Recordset
```

```
Private Sub ocultar()
```

```
Dim i As Integer
'primero se deja los cuadros de texto sin datos
For i = 0 To txtUsuario.Count - 1
    txtUsuario(i).Text = ""
Next i
Empresa0.Hide
For i = 0 To InicioAplica2.mnuEmpre.Count - 1
    InicioAplica2.mnuEmpre(i).Enabled = True
Next i
End Sub
```

Este método en primer lugar deja los cuadros de texto sin datos. Después habilita todos los títulos de menú del formulario principal.

```
Private Sub cmdUsuarios_Click(Index As Integer)
```

```
rsUsuarios.Index = "usuario"
rsUsuarios.Seek "=", txtUsuario(0).Text
Select Case (Index)
Case 0:
    If Not rsUsuarios.NoMatch Then
        If (txtUsuario(1).Text = rsUsuarios.Fields("contraseña")) Then
            MsgBox ("Acceso Permitido. Se cerrará la ventana")
            Call ocultar
            Call InicioAplica2.habilitar(True)
        End If
    Else
        MsgBox ("El usuario o contraseña son incorrectos")
    End If
Case 1:
```

Es la lógica de los botones de comando.

El primer botón valida el usuario y contraseña. Si existe el usuario y la clave es la correcta. Se oculta este formulario y luego se llama al método del formulario padre habilitar.

```
If (cmdUsuarios(Index).Caption = "&Modificar") Then
    If Not rsUsuarios.NoMatch Then
        txtUsuario(2).Visible = True
        cmdUsuarios(0).Enabled = False
        cmdUsuarios(Index).Caption = "&Grabar"
    Else
        MsgBox ("El usuario no es correcto")
    End If
Else
    rsUsuarios.Edit
    rsUsuarios.Fields("contraseña") = txtUsuario(2).Text
    rsUsuarios.Update
    txtUsuario(2).Visible = False
    cmdUsuarios(0).Enabled = True
    cmdUsuarios(Index).Caption = "&Modificar"
End If
End Select
End Sub
```

Este botón se ocupa del cambio de contraseña del usuario.

Cuando se carga el formulario se abre la base de datos y el recordset.

**Private Sub Form\_Load()**

```
Set dbBase = OpenDatabase(App.Path & "\bdEmpresas.mdb")
Set rsUsuarios = dbBase.OpenRecordset("Usuarios")
End Sub
```

**Private Sub txtUsuario\_Validate(Index As Integer, Cancel As Boolean)**

```
Select Case (Index)
Case 0:
    If (txtUsuario(Index).Text <> "" And Len(txtUsuario(Index)) <= 20) Then
        Cancel = False
    Else
        MsgBox ("Debe completar el campo " & txtUsuario(Index).Tag & "con no más de 20 caracteres")
        Cancel = True
    End If
Case 1, 2:
    If txtUsuario(Index).Text <> "" And Len(txtUsuario(Index)) <= 8 Then
        Cancel = False
    Else
        MsgBox ("Debe completar el campo " & txtUsuario(Index).Tag & "con hasta 8 dígitos")
        Cancel = True
    End If
End Select
End Sub
```

Valida el texto del nombre de usuario como la contraseña.  
El nombre de usuario no tiene que tener más de 20 dígitos ni estar vacío.  
La contraseña debe tener hasta 8 dígitos y no estar vacía

## Proyecto Aplica2

Es más elaborado que el anterior. Permite trabajar con las ABM de Cargos. Las modificaciones están indicadas en las llamadas de atención celestes. Además para realizar el formulario correspondiente a las ABM Cargos se ha reutilizado el código del proyecto Cargos6.

### Código de InicioAplica3:

Option Explicit



```
Dim gcValido As Boolean  
Dim j As Integer
```

```
Public Sub habilitar(valor As Boolean)  
gcValido = valor  
End Sub
```

```
Private Sub mnuEmpre_Click(Index As Integer)  
If (gcValido) Then  
Select Case (Index)  
Case 0: 'submenu sistema  
For j = 0 To mnuEmpre.Count - 1  
mnuEmpre(j).Enabled = False  
Next j  
Empresa10.Show  
Case 1: 'submenu cargos  
'Empresa3.Show  
'mnuCargos_click (0)  
Case 2: 'submenu departamentos  
Empresa12.Show  
Case 3: 'submenu empleados  
Empresa11.Show  
Case 4: ' submenu sueldos  
Empresa14.Show  
End Select  
End If  
End Sub
```

Si se pulsa este elemento se deshabilitan todos los títulos de menú.

```
Private Sub mnuCargos_click(Index As Integer)  
If (gcValido) Then  
Select Case (Index) 'el índice del submenu  
Case 0: Empresa13.Show  
Case 2: MsgBox ("Todavía no está realizado el formulario de consultas")  
Case 4: MsgBox ("Todavía no está realizado el formulario de impresión")  
End Select  
End If  
End Sub
```

#### Código del formulario Empresa10:

```
Option Explicit  
Dim dbBase As Database  
Dim rsUsuarios As Recordset  
Dim i As Integer
```

```
Private Sub ocultar()  
'primero se deja los cuadros de texto sin datos  
For i = 0 To txtUsuario.Count - 1  
txtUsuario(i).Text = ""  
Next i  
Empresa10.Hide  
For i = 0 To InicioAplica3.mnuEmpre.Count - 1  
InicioAplica3.mnuEmpre(i).Enabled = True  
Next i  
End Sub
```

```
Private Sub Form_Load()  
Set dbBase = OpenDatabase(App.Path & "\bdEmpresas.mdb")  
Set rsUsuarios = dbBase.OpenRecordset("Usuarios")  
End Sub
```

```
Private Sub cmdUsuarios_Click(Index As Integer)
```

```
rsUsuarios.Index = "usuario"
```

```
rsUsuarios.Seek "=", txtUsuario(0).Text
```

```
Select Case (Index)
```

```
Case 0:
```

```
    If Not rsUsuarios.NoMatch Then
```

```
        If (txtUsuario(1).Text = rsUsuarios.Fields("contraseña")) Then
```

```
            MsgBox ("Acceso Permitido. Se cerrará la ventana")
```

```
            Call ocultar
```

```
            Call InicioAplica3.habilitar(True)
```

```
        Else
```

```
            MsgBox ("Acceso Denegado")
```

```
            Call ocultar
```

```
            Call InicioAplica3.habilitar(False)
```

```
        End If
```

```
    Else
```

```
        MsgBox ("El usuario o contraseña son incorrectos")
```

```
    End If
```

```
Case 1:
```

```
    If (cmdUsuarios(Index).Caption = "&Modificar") Then
```

```
        If Not rsUsuarios.NoMatch Then
```

```
            txtUsuario(2).Visible = True
```

```
            cmdUsuarios(0).Enabled = False
```

```
            cmdUsuarios(Index).Caption = "&Grabar"
```

```
        Else
```

```
            MsgBox ("El usuario no es correcto")
```

```
        End If
```

```
    Else
```

```
        rsUsuarios.Edit
```

```
        rsUsuarios.Fields("contraseña") = txtUsuario(2).Text
```

```
        rsUsuarios.Update
```

```
        txtUsuario(2).Visible = False
```

```
        cmdUsuarios(0).Enabled = True
```

```
        cmdUsuarios(Index).Caption = "&Modificar"
```

```
    End If
```

```
End Select
```

```
End Sub
```

Si se ingresó mal el usuario o contraseña se deshabilitan todos los títulos de menú

```
Private Sub txtUsuario_Validate(Index As Integer, Cancel As Boolean)
```

```
Select Case (Index)
```

```
Case 0:
```

```
    If (txtUsuario(Index).Text <> "" And Len(txtUsuario(Index)) <= 20) Then
```

```
        Cancel = False
```

```
    Else
```

```
        MsgBox ("Debe completar el campo " & txtUsuario(Index).Tag & "con no más de 20 caracteres")
```

```
        Cancel = True
```

```
    End If
```

```
Case 1, 2:
```

```
    If txtUsuario(Index).Text <> "" And Len(txtUsuario(Index)) <= 8 Then
```

```
        Cancel = False
```

```
    Else
```

```
        MsgBox ("Debe completar el campo " & txtUsuario(Index).Tag & "con hasta 8 dígitos")
```

```
        Cancel = True
```

```
    End If
```

```
End Select
```

```
End Sub
```



**Código del formulario Empresa13:**

Option Explicit  
Dim i As Integer

'PARA EL ARRAY DE CONTROLES de los botones  
\_de comandos

Private Sub cmdComandos\_Click(Index As Integer)

If Index = 0 Or Index = 1 Then

    'para habilitar los cuadros de texto

    For i = 0 To txtTexto.Count - 1

        txtTexto(i).Enabled = True

    Next i

Else

    'para deshabilitar los cuadros de texto

    For i = 0 To txtTexto.Count - 1

        txtTexto(i).Enabled = False

    Next i

End If

    'usamos un select case

    Select Case (Index)

    Case 0: 'para insertar

        txtTexto(0).SetFocus

        dataCargos.Recordset.AddNew

        dataCargos.Caption = "Proceso de inserción"

        cmdComandos(0).Enabled = False

        cmdComandos(1).Enabled = False

        cmdComandos(4).Enabled = False

    Case 1: 'para modificar un registro existente

        txtTexto(1).SetFocus

        dataCargos.Recordset.Edit

        dataCargos.Caption = "Proceso de actualización"

        cmdComandos(0).Enabled = False

        cmdComandos(1).Enabled = False

        cmdComandos(4).Enabled = False

    Case 2: 'Se utiliza para confirmar una inserción o modificación

        If (cmdComandos(1).Enabled = False Or cmdComandos(0).Enabled = False) Then

            dataCargos.Recordset.Update

            dataCargos.Caption = "Proceso de confirmación de inserción o actualización"

            cmdComandos(1).Enabled = True

            cmdComandos(0).Enabled = True

            cmdComandos(4).Enabled = True

        End If

    Case 3: 'Se utiliza para deshacer una inserción o modificación

        If (cmdComandos(1).Enabled = False Or cmdComandos(0).Enabled = False) Then

            dataCargos.Recordset.CancelUpdate

            dataCargos.Caption = "Proceso de reversión de inserción o actualización"

            cmdComandos(1).Enabled = True

            cmdComandos(0).Enabled = True

            cmdComandos(4).Enabled = True

        End If

    Case Else: 'corresponde al índice 4 que es para eliminar un registro existente

        If (dataCargos.Recordset.EOF = False) Then

            dataCargos.Recordset.Delete

            dataCargos.Caption = "Proceso de eliminación de registro"

        End If

    End Select

End Sub

Es casi el mismo proyecto Cargos6. Se ha ocultado el control Data y se ha agregado que el array txtTexto esté deshabilitado salvo para cuando se van a agregar o modificar registros.

Código para habilitar/deshabilitar la edición en los cuadros de texto.

Este código se podría sacar

```
'para validar las entradas realizadas por el usuario en cuanto _  
a los campos que son obligatorios utilizaremos el método Validate  
Private Sub txtTexto_GotFocus(Index As Integer)  
'sirve para darle formato al ingreso de datos del usuario  
Select Case (Index)  
Case 2, 3:  
    txtTexto(Index).Text = Format(txtTexto(Index).Text, "$#,###,###.00")  
End Select  
End Sub
```

```
'el usuario en IdCargo sólo puede ingresar numeros  
Private Sub txtTexto_KeyPress(Index As Integer, KeyAscii As Integer)  
Dim i As Integer  
  
If (Index = 0) Then  
    If (KeyAscii >= 48 And KeyAscii <= 57) Then  
        KeyAscii = KeyAscii  
    Else  
        i = MsgBox("Debe Ingresar sólo valores numéricos", vbCritical, "Error")  
        KeyAscii = 0  
    End If  
End If  
  
End Sub
```

```
'así se valida que el usuario ingrese sólo 2 números  
Private Sub txtTexto_Validate(Index As Integer, Cancel As Boolean)  
Dim i As Integer  
Select Case (Index)  
Case 0:  
'primero se verifica que el campo no esté vacío  
'luego se verifica para cada campo a considerar la longitud de la cadena  
If txtTexto(Index).Text = "" Or txtTexto(Index).Text = 0 Then  
    i = MsgBox("Debe ingresar los datos requeridos", vbInformation, "Alerta")  
    Cancel = True  
Else  
    If (Len(txtTexto(Index)) > 2) Then  
        i = MsgBox("Debe ingresar hasta dos números", vbInformation, "Alerta")  
        Cancel = True  
    Else  
        Cancel = False  
    End If  
End If  
Case 1:  
If txtTexto(Index).Text = "" Then  
    i = MsgBox("Debe ingresar los datos requeridos", vbInformation, "Alerta")  
    Cancel = True  
Else  
    If (Len(txtTexto(Index).Text) > 12) Then  
        i = MsgBox("Debe ingresar hasta doce cacteres", vbInformation, "Alerta")  
        Cancel = True  
    Else  
        Cancel = False  
        'sirve para cambiar toda la cadena a mayúsculas  
        txtTexto(Index).Text = UCase(txtTexto(Index).Text)  
    End If  
End If
```

Case 2, 3:

```
If (Len(txtTexto(Index).Text) > 10) Then  
    i = MsgBox("Debe ingresar un valor no mayor a 9.999.999,99", vbInformation, "Alerta")  
    Cancel = True  
Else  
    Cancel = False
```

```
End If
```

```
End Select
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Dim i As Integer
```

```
For i = 0 To InicioAplica3.mnuEmpre.Count - 1
```

```
    InicioAplica3.mnuEmpre(i).Enabled = True
```

```
Next i
```

```
End Sub
```