

VISUAL BASIC – Operaciones de Altas, Modificaciones y Bajas (incluyendo búsquedas y operaciones sobre el Recordset) sin utilizar un control Data - Proyecto DMLVB

Para todos los formularios de los ejemplos los comentarios dentro del código fuente están en verde, las instrucciones propias de DML están sombreadas de celeste y las que tienen que ver con el manejo de propiedades de controles y formularios están sombreadas en amarillo.

Formulario Cargos5

Permite ver el funcionamiento de la apertura de la base de datos, del recordset y el cierre de la base de datos antes abierta.

Recomendación: si el formulario que uno ha abierto no tiene vinculado DAO hay que hacerlo llamando al **Menu Proyecto/Referencias** y buscar allí y tildar **Microsoft DAO** (la última versión).

El código fuente de este ejemplo es:

Option Explicit

Dim dbBaseDatos As Database

Dim rsCargos As Recordset

Private Sub cmdOpci_Click(Index As Integer)

Dim intMsg As Integer

Select Case (Index)

Case 0:

'Verifica si existe el archivo de la base de datos _
si existe entonces la abre y muestra el mensaje Ok _
sino muestra el mensaje de advertencia

If (VBA.FileLen(App.Path & "\bdEmpresas.mdb") >= 0) Then

Set dbBaseDatos = OpenDatabase(App.Path & "\ " & "bdEmpresas.mdb")

intMsg = MsgBox("Ha abierto la base de datos " & dbBaseDatos.Name, vbInformation, "Información")

With cmdOpci

.Item(0).Enabled = False

.Item(1).Enabled = True

.Item(2).Enabled = True

End With

Else: intMsg = MsgBox("No existe la Base de Datos", vbExclamation, "Advertencia")

End If

Case 1:

If dbBaseDatos.TableDefs.Count > 0 Then

Set rsCargos = dbBaseDatos.OpenRecordset("cargos", 1)

intMsg = MsgBox("Ha abierto la tabla Cargos", vbInformation, "Información")

Else: intMsg = MsgBox("No existe la Tabla Cargos", vbExclamation, "Advertencia")

End If

Case Else:

'Si la base de datos está abierta puede cerrarse _

si no se verifica esto lanzará un mensaje de error

If (VBA.FileLen(App.Path & "\bdEmpresas.mdb") >= 0) Then

intMsg = MsgBox("Se está cerrando la base de datos " & dbBaseDatos.Name, vbInformation, "Información")

dbBaseDatos.Close

With cmdOpci

.Item(0).Enabled = True

.Item(1).Enabled = False

.Item(2).Enabled = False

End With

Else: intMsg = MsgBox("La base de datos no está abierta, no puede cerrarse", vbExclamation, "Advertencia")

End If

End Select

End Sub

Nota: para poder utilizar los botones el código fuente tanto de la apertura de la base de datos y del recordset han sido escritos dentro del método clic. En otra circunstancia, lo correcto es incorporar en el método Load o Activate del formulario este código y en el método UnLoad o DeActivate del formulario el cierre de la base de datos.

Formulario Cargos6

Este formulario permite ver cómo se pueden pasar argumentos con las propiedades Tag de un cuadro de texto o de un label. Asimismo tiene habilitada la propiedad WhatsThisHelp en el formulario y en cada control (cuadros de texto, botones) tiene un comentario de ayuda en la propiedad ToolTipText. El código relativo a las operaciones de actualización y búsqueda ha sido sombreado de celeste y el código relativo a manejo de información entre los distintos controles con verde claro.

Option Explicit

Dim dbBase As Database

Dim rsCargos As Recordset

Dim j As Integer

Dim i As Integer

Private Sub LimpiaTexto()

For i = 0 To txtTexto.Count - 1

txtTexto(i).Text = ""

Next i

txtTexto(0).SetFocus

End Sub

Private Sub HabilitaBotones(logico As Boolean)

For i = 0 To cmdOpci.Count - 2 'todos menos el botón Open (abre la base de datos)

cmdOpci(i).Enabled = logico

Next i

End Sub

Private Sub Form_Load()

Set dbBase = OpenDatabase(App.Path & "\ & "bdEmpresas.mdb")

Set rsCargos = dbBase.OpenRecordset("Cargos")

End Sub

'ESTOS METODOS HAN SIDO REHUSADOS TAL CUAL DEL FORMULARIO CARGOS4
no se colocan aquí

'ESTOS METODOS SON NUEVOS PARA MANEJAR LAS ALTAS, MODIFICACIONES _
Y BAJAS SIN UTILIZAR UN CONTROL DATA SINO SÓLO DML DE VISUAL BASIC

Private Sub cmdOpci_Click(Index As Integer)

Select Case (Index)

Case 0:

If (cmdOpci(Index).Caption = "&AddNew") Then

rsCargos.AddNew

'CAMBIA el caption del botón y la ayuda ¿qué es esto?

cmdOpci(Index).Caption = "&Update"

cmdOpci(Index).ToolTipText = "Permite grabar la inserción en el almacén físico"

Else

For i = 0 To txtTexto.Count - 1

If txtTexto(i).Text <> "" Then

rsCargos.Fields(txtTexto(i).Tag) = txtTexto(i).Text

Else

MsgBox ("Debe Ingresar los campos")

i = txtTexto.Count

End If

Next i

rsCargos.Update

El botón cambia de **AddNew** a **Update**, por ello esta comprobación. Si estamos en **AddNew** se aplica el método AddNew sino Update

Como estamos en **Update**, a cada campo del Recordset (método field usando la propiedad Tag del control) se le pasa el valor escrito por el usuario y luego se aplica rsCargos.Update

```
'cambia el caption del botón y el mensaje de la ayuda ¿Qué es esto?
cmdOpci(Index).Caption = "&AddNew"
cmdOpci(Index).ToolTipText = "Agrega un nuevo registro al recordset"
End If
```

Case 1:

'para poder ver los datos de la tabla Cargos en los _
cuadros de texto

```
If (cmdOpci(Index).Caption = "&Edit") Then
'se puede escribir también _
rsCargos.Index = txtTexto(0).Tag
rsCargos.Index = "idcargo"
rsCargos.Seek "=", txtTexto(0).Text
```

El botón cambia de **Edit** a **Update**, por ello esta comprobación. Si estamos en **Edit** se aplica el método **rsCargos.Seek** para efectuar la búsqueda y sino **rsCargos.Edit** sino **rsCargos.Update**

```
If rsCargos.NoMatch Then
```

```
MsgBox "Registro No encontrado"
```

```
Else
```

```
'CAPTURA LA INFO DEL SEEK
```

```
MsgBox "Registro Encontrado"
```

```
For i = 0 To txtTexto.Count - 1
```

```
txtTexto(i).Text = rsCargos.Fields(i).Value
```

```
Next i
```

```
'CAMBIA el caption del botón y la ayuda ¿qué es esto?
```

```
cmdOpci(Index).Caption = "&Update"
```

```
cmdOpci(Index).ToolTipText = "Permite grabar la modificación en el almacén físico"
```

```
End If
```

Captura cada valor de cada campo del Recordset **rsCargos** en cada cuadro de texto para que el usuario pueda leer y modificar la información

```
Else
```

```
rsCargos.Edit
```

```
For i = 1 To txtTexto.Count - 1
```

```
If txtTexto(i).Text <> "" Then
```

```
rsCargos.Fields(txtTexto(i).Tag) = txtTexto(i).Text
```

```
Else
```

```
i = txtTexto.Count
```

```
End If
```

```
Next i
```

```
rsCargos.Update
```

Mediante el método **Edit** habilita la edición en el Recordset. ¿Cómo se hace esto con los eventos? Se supone que antes de apretar el botón diciendo **Update** el usuario introdujo los cambios que quería realizar. Entonces al apretar **Update** se para cada campo del Recordset se toma el valor de cada cuadro de texto. Finalmente se aplica al Recordset el método **Update**

'cambia el caption del botón y el mensaje _
de la ayuda ¿Qué es esto?

```
cmdOpci(Index).Caption = "&Edit"
```

```
cmdOpci(Index).ToolTipText = "Permite modificar, previa búsqueda, un registro del recordset"
```

```
End If
```

Case 2:

```
Dim intElimino As Integer
```

```
rsCargos.Index = "idcargo"
```

```
rsCargos.Seek "=", txtTexto(0).Text
```

```
If rsCargos.NoMatch Then
```

```
MsgBox "Registro No encontrado"
```

```
Else
```

```
'CAPTURA LA INFO DEL SEEK
```

```
MsgBox "Registro Encontrado"
```

```
For i = 0 To txtTexto.Count - 1
```

```
txtTexto(i).Text = rsCargos.Fields(i).Value
```

```
Next i
```

```
intElimino = MsgBox("¿Realmente desea eliminar el registro?", vbYesNo, "Información")
```

```
If intElimino = 1 Then
```

```
rsCargos.Delete
```

El botón **Delete** tiene una lógica similar. Primero para que el usuario pueda verificar que los datos buscados son los correctos se carga la información del Recordset a cada cuadro de texto. Luego se aplica el método **Delete**

```
        MsgBox "Registro Eliminado"
    Else
        MsgBox "El registro no se eliminó"
    End If
End If
Call HabilitaBotones(False)

Case 3:
    dbBase.Close
    cmdOpci(Index + 1).Enabled = True

    Case Else
        Call Form_Load
        cmdOpci(Index).Enabled = False
    End Select
'se deshabilitan todos los botones para empezar de vuelta
Call HabilitaBotones(False)
End Sub

Private Sub cmdPanta_Click(Index As Integer)
    If cmdPanta.Item(0) Then
        Call HabilitaBotones(True)
    Else: Call LimpiaTexto
    End If
End Sub
```

Formulario Cargos7

Este formulario permite ver cómo funcionan los métodos *MoveFirst*, *MoveLast*, *MoveNext* y *MovePrevious* del Recordset. Recordemos que permiten navegar por todo el Recordset sin ningún criterio o condición de Búsqueda. Por tratarse de métodos más fáciles de comprender se tratan antes que aquellos que permiten mover el puntero de registro en base a una condición dada (métodos Find... que se tratan en el proyecto siguiente)

```
Option Explicit
Dim dbBase As Database
Dim rsCargos As Recordset
Dim intReg As Integer
Dim i As Integer

Private Sub Form_Load()
    Set dbBase = OpenDatabase(App.Path & "\ & "bdEmpresas.mdb")
    Set rsCargos = dbBase.OpenRecordset("Cargos", dbOpenDynaset)
End Sub

Private Sub cmdOpci_Click(Index As Integer)
    Select Case (Index)
        Case 0:
            rsCargos.MoveFirst
            intReg = 1

        Case 1:
            If intReg = 1 Then
                Beep
            Else
                rsCargos.MovePrevious
                intReg = intReg - 1
            End If

        Case 2:
            If intReg = rsCargos.RecordCount Then
                Beep
            Else
```

```

rsCargos.MoveNext
intReg = intReg + 1
End If

Case 3:
rsCargos.MoveLast
intReg = rsCargos.RecordCount
End Select
For i = 0 To txtTexto.Count - 1
txtTexto(i).Text = rsCargos.Fields(i).Value
Next i
End Sub

```

Captura cada valor de cada campo del Recordset rsCargos en cada cuadro de texto para que el usuario pueda leer y modificar la información

Formulario Cargos8

Este formulario permite ver cómo funcionan los métodos *FindFirst*, *FindLast*, *FindNext* y *FindPrevious* del Recordset. Recordemos que permiten navegar por todo el Recordset pero en base a una condición. Si esta se cumple tendremos acceso a un subconjunto del Recordset sino no tendremos datos.

El formulario está pensado y diseñado de forma tal que el usuario ingresa una cadena con el criterio o condición que desea. Al final del código fuente se muestran unas ejecuciones como para que se tenga una idea más acabada de cómo se utiliza el formulario y sobretodo el control txtCondicion.

El código fuente es:

```

Option Explicit
Dim dbBase As Database
Dim rsCargos As Recordset
Dim intReg As Integer
Dim intCtos As Integer
Dim i As Integer

Private Sub Calprimero()
Dim j As Integer
intCtos = 1
j = 0

Do While j <> -1
rsCargos.FindNext (txtCondicion.Text)
If (rsCargos.NoMatch) Then
j = -1
Else
intCtos = intCtos + 1
End If
Loop
rsCargos.FindFirst (txtCondicion.Text)
End Sub

Private Sub Calultimo()
Dim j As Integer
intCtos = 1
j = 0

Do While j <> -1
rsCargos.FindPrevious (txtCondicion.Text)
If (rsCargos.NoMatch) Then
j = -1
Else
intCtos = intCtos + 1
End If
Loop
rsCargos.FindLast (txtCondicion.Text)
End Sub

Private Sub Command1_Click()

```

Calcula cuántos registros tendrá el subconjunto de datos a mostrar empezando por FindFirst (en caso de que el usuario haya elegido esta opción). Mientras que la condición coincida actualizará la variable intCtos sino sale (se le asigna a j un -1)

Una vez obtenido el máximo número de registros que el usuario puede consultar se vuelve a aplicar el método FindFirst.

Similar al anterior sólo que parte de que el usuario haya querido consultar el último registro que coincide con la condición de búsqueda.

```
cmdOpci(1).Enabled = False 'se deshabilita el botón siguiente registro
cmdOpci(2).Enabled = False 'se deshabilita el botón anterior registro
txtCondicion.Text = ""
lblRegistros(1).Caption = "0"
For i = 0 To txtTexto.Count - 1
    txtTexto(i).Text = ""
Next i
End Sub
```

```
Private Sub Form_Load()
```

```
    Set dbBase = OpenDatabase(App.Path & "\ & "bdEmpresas.mdb")
    Set rsCargos = dbBase.OpenRecordset("Cargos", dbOpenDynaset)
```

```
End Sub
```

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

```
Select Case (Index)
```

```
    Case 0:
```

```
        rsCargos.FindFirst (txtCondicion.Text)
```

```
        If rsCargos.NoMatch Then
```

```
            MsgBox ("No existe el registro")
```

```
        Else
```

```
            intReg = 1
```

```
            Call Calprimero
```

```
            lblRegistros(1).Caption = intCtos
```

```
            With cmdOpci
```

```
                .Item(1).Enabled = True 'se habilitan los restantes botones de navegacion
```

```
                .Item(2).Enabled = True
```

```
            End With
```

```
        End If
```

```
    Case 1:
```

```
        If intReg = 1 Then
```

```
            Beep
```

```
        Else
```

```
            rsCargos.FindPrevious (txtCondicion.Text)
```

```
            intReg = intReg - 1
```

```
        End If
```

```
    Case 2:
```

```
        If intReg = intCtos Then
```

```
            Beep
```

```
        Else
```

```
            rsCargos.FindNext (txtCondicion.Text)
```

```
            intReg = intReg + 1
```

```
        End If
```

Case 3:

```
rsCargos.FindLast (txtCondicion.Text)
If rsCargos.NoMatch Then
    MsgBox ("No existe el registro")
Else
    Call Calultimo
    intReg = intCtos
    lblRegistros(1).Caption = intCtos
    With cmdOpCi
        .Item(1).Enabled = True
        .Item(2).Enabled = True
    End With
End If
End Select
For i = 0 To txtTexto.Count - 1
    txtTexto(i).Text = rsCargos.Fields(i).Value
Next i
End Sub
```

Captura cada valor de cada campo del Recordset rsCargos en cada cuadro de texto para que el usuario pueda leer y modificar la información

Captura la condición escrita por el usuario. El usuario hace clic en el rótulo que desee utilizar en la condición y luego puede completar la condición en el cuadro de texto txtCondicion

```
Private Sub lblRotulos_Click(Index As Integer)
    txtCondicion.SetFocus
    txtCondicion.Text = txtCondicion.Text & lblRotulos(Index).Tag & " "
End Sub
```

Seguidamente se muestran algunas pantallas en base a la ejecución de este formulario:

Este es el formulario una vez que se inicia.

Hemos pulsado el rótulo IdCargo y automáticamente cambia el enfoque y se completa el cuadro de texto txtCondicion. Tenemos que completar la condicion

Cargos

Ingresar condición: idcargo >=30

< || < > || > Nueve Búsqueda

MoveFirst

Total de Registros Coincidentes: 0

IdCargo	Nombre Cargo	Salario Min	Salario Max

Es una condición realizada con la clave primaria; es decir que quiero conocer los registros para los cuales el idcargo es mayor o igual a treinta. Además la búsqueda se va a efectuar con FindFirst puesto que estamos

posicionados en dicho botón (propiedad ToolTipText del botón).

Cargos

Ingresar condición: idcargo >=30

< || < > || > Nueve Búsqueda

MoveFirst

Total de Registros Coincidentes: 9

IdCargo	Nombre Cargo	Salario Min	Salario Max
30	VENDEDOR	1000	3000

El resultado es la devolución del primer registro que coincide con la condición de búsqueda.

Cargos

Ingresar condición: idcargo >=30

< || < > || > Nueve Búsqueda

MoveNext

Total de Registros Coincidentes: 9

IdCargo	Nombre Cargo	Salario Min	Salario Max
31	ADMINISTR	800	2500

Nos movemos al siguiente registro (indicado en la imagen) que corresponde al idcargo = 31. También podríamos habernos movido al último o de vuelta al primero o al anterior. Es de notar también que figura el número de registros coincidentes igual a 9.

Cargos

Ingresar condición: nbrecargo LIKE 'JEFE*

< || < > || > Nueve Búsqueda

MoveLast

Total de Registros Coincidentes: 0

IdCargo	Nombre Cargo	Salario Min	Salario Max

En este caso se ha pulsado el rótulo Nombre Cargo que tiene como propiedad Tag="nbrecargo", que es el nombre del campo del Recordset sobre el cual nos interesa realizar la condición de búsqueda. Este es uno de los

casos más complicados puesto que lo que queremos es poder buscar aquellos registros cuyo nombre de cargo empiece con JEFE; es decir, se tipeó: 'JEFE*'. Para que no haya problemas hay que introducir la palabra deseada, entre comillas simples. Se ha agregado el asterisco para indicar que la búsqueda debe incluir a cualquier cadena (dentro del campo nbrecargo del Recordset) que empiece con JEFE. Si quisiéramos encontrar todos aquellos cargos cuyo nombre tiene la palabra ADM deberíamos haber escrito '*ADM*'. Además vamos a realizar la búsqueda empezando por el último registro.

The screenshot shows the 'Cargos' application window. At the top, there is a search bar labeled 'Ingresar condición' with the text 'nbrecargo LIKE 'JEFE*'' entered. Below the search bar are four navigation buttons: '< ||', '<', '>', and '|| >'. To the right of these buttons is a button labeled 'Nueva Búsqueda'. Below the navigation buttons, it says 'Total de Registros Coincidentes: 7' and a button labeled 'MoveLast'. At the bottom, there is a table with four columns: 'IdCargo', 'Nombre Cargo', 'Salario Min', and 'Salario Max'. The first row of the table contains the values: '17', 'JEFE PR02', '2000', and '8000'.

Este es el resultado de la búsqueda en base a la condición deseada.

The screenshot shows the 'Cargos' application window. At the top, there is a search bar labeled 'Ingresar condición' with the text 'nbrecargo LIKE 'JEFE*'' entered. Below the search bar are four navigation buttons: '< ||', '<', '>', and '|| >'. To the right of these buttons is a button labeled 'Nueva Búsqueda'. Below the navigation buttons, it says 'Total de Registros Coincidentes: 7' and a button labeled 'MoveFirst'. At the bottom, there is a table with four columns: 'IdCargo', 'Nombre Cargo', 'Salario Min', and 'Salario Max'. The first row of the table contains the values: '10', 'JEFE CONTA', '2000', and '8000'.

Nos fuimos al primer registro.

The screenshot shows the 'Cargos' application window. At the top, there is a search bar labeled 'Ingresar condición' with the text 'nbrecargo LIKE 'PTE.*'' entered. Below the search bar are four navigation buttons: '< ||', '<', '>', and '|| >'. To the right of these buttons is a button labeled 'Nueva Búsqueda'. Below the navigation buttons, it says 'Total de Registros Coincidentes: 0'. At the bottom, there is a table with four columns: 'IdCargo', 'Nombre Cargo', 'Salario Min', and 'Salario Max'. The first row of the table contains empty fields. A dialog box titled 'DMLVB' is open in the center of the window, displaying the message 'No existe el registro' and a button labeled 'Aceptar'.

La condición ingresada nos dio un resultado de False motivo por el cual el registro o grupo de registros no existen, cuestión que se

indica al usuario.