

Acceso Desconectado a Base de Datos (Utilizando DataBinding)

Descripción:

- Realizar una aplicación de ABM contra la base de datos ***bd_telefonos.mdb***.
- Se visualizará en el formulario cada uno de los campos del registro seleccionado, permitiendo modificar cualquiera de ellos.
- Los campos nombre, dirección y teléfono se visualizarán en TextBoxes y el campo observaciones en un TextBox multilínea.
- Permitirá moverse de un registro a otro con el fin de visualizarlos o editarlos.

Componentes:

- DataSet
- Formulario
- Textboxes, Buttons y Labels

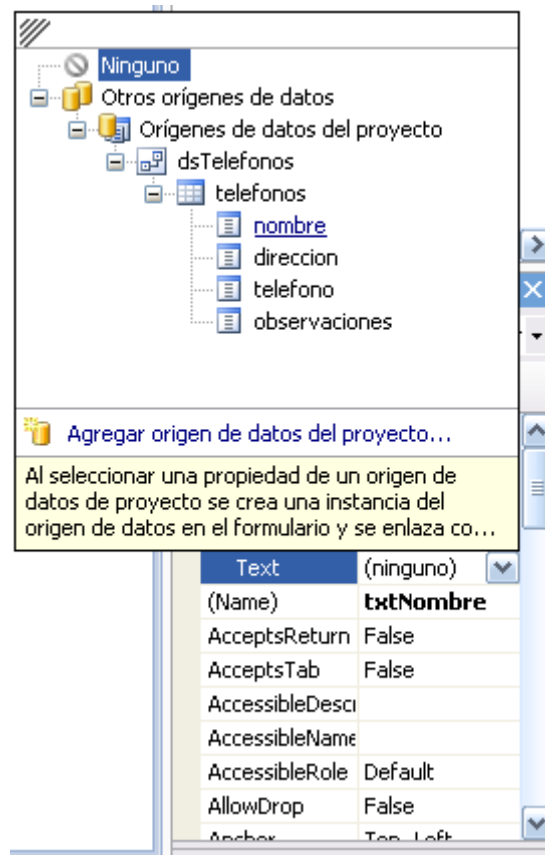
Pasos:

1. Ejecutar ***Mostrar Orígenes de datos*** del menú ***Datos*** y haga clic en ***Agregar nuevo origen de datos***.

Una vez creada la conexión, cree un conjunto de datos (DataSet): dsTelefonos. Para crear este conjunto de datos elija:

- Tipo de origen de base de datos: Base de Datos.
- Conexión de datos: bd_telefonos.mdf
- Objetos de base de datos: Tabla telefonos
- Nombre del DataSet: dsTelefonos

2. El siguiente paso será vincular los TextBoxes con el DataSet. Para ello hay que seleccionar un TextBox y en su propiedad DataBindings vincular el campo nombre de la tabla telefonos del conjunto de datos dsTelefonos con la propiedad Text de txtNombre.



Esta operación añadirá a Form1 un DataSet DsTelefonos de tipo dsTelefonos, un adaptador TelefonosTableAdapter para acceder a la tabla de *teléfonos* de la base de datos y un componente telefonosBindingSource de la clase BindingSource conectado al origen de datos.

3. Repita el paso 2 para el resto de los TextBoxes.
4. Para los botones de desplazamiento (Primero, Anterior, Siguiente y Ultimo) hay que utilizar los métodos del objeto BindingSource TelefonosBindingSource MoveFirst, MoveNext, MovePrevious y MoveLast respectivamente.
El objeto BindingSource hace de puente entre el control y el conjunto de datos, proporcionando acceso a los datos actualmente mostrados por el control en forma indirecta, incluyendo navegación, ordenación, filtrado y actualización.
Este objeto se encargará de sincronizar los cuatro controles para que juntos muestren nombre, dirección, teléfono y observaciones del registro que está en esa posición.

```
Private Sub btPrimero_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnPrimero.Click

    'TelefonosBindingSource.Position = 0
    TelefonosBindingSource.MoveFirst()
    MostrarPosicion()

End Sub
```

```

Private Sub btAnterior_Click(ByVal sender As System.Object, _
                             ByVal e As System.EventArgs) Handles btnAnterior.Click

    'TelefonosBindingSource.Position -= 1
    TelefonosBindingSource.MovePrevious()
    MostrarPosicion()

End Sub

```

```

Private Sub btSiguiente_Click(ByVal sender As System.Object, _
                               ByVal e As System.EventArgs) Handles btnSiguiente.Click

    'TelefonosBindingSource.Position += 1
    TelefonosBindingSource.MoveNext()
    MostrarPosicion()

End Sub

```

```

Private Sub btUltimo_Click(ByVal sender As System.Object, _
                            ByVal e As System.EventArgs) Handles btnUltimo.Click

    'TelefonosBindingSource.Position = TelefonosBindingSource.Count - 1
    TelefonosBindingSource.MoveLast()
    MostrarPosicion()

End Sub

```

Añadir, borrar y buscar datos.

5. Para poder agregar un registro nuevo a nuestro DataSet colocaremos el siguiente código:

```

Private Sub btAñadir_Click(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles btnAñadir.Click

    REM CREO UNA TABLA DE TIPO DSTELEFONOS
    Dim miTabla As DataTable = DsTelefonos.telefonos
    REM CREO UNA COLECCION DE FILAS
    Dim colFilas As DataRowCollection = miTabla.Rows
    REM CREO LA NUEVA FILA
    Dim nuevaFila As DataRow

    Try
        REM LE DOY EL ESQUEMA DE MI TABLA
        nuevaFila = miTabla.NewRow()

        REM GENERO DATOS POR DEFECTO
        nuevaFila(0) = "Nombre"
        nuevaFila(1) = "Dirección"
        nuevaFila(2) = "Teléfono"
        nuevaFila(3) = "Observaciones"

        REM AGREGO LA FILA A LA COLECCION
        colFilas.Add(nuevaFila)

        REM GENERA UN CLICK SOBRE UN BOTON
        btnUltimo.PerformClick()

        REM MUESTRO LA NUEVA POSICION
        MostrarPosicion()

        REM PONGO EN FOCO AL NOMBRE
        txtNombre.Focus()

    Catch ex As System.Data.ConstraintException
        MessageBox.Show(ex.Message)
    End Try

End Sub

```

6. En el caso que se desee borrar algún registro, se debe borrar la fila del DataGridView.

```

Private Sub btBorrar_Click(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles btnBorrar.Click

    REM CREO UNA FILA
    Dim vistaFilaActual As DataRowView
    Dim NL As String = Environment.NewLine

    REM SI SE DESEA BORRAR, LO MARCO COMO DELETE
    If (MessageBox.Show("¿Desea borrar este registro?" & NL, _
        "Buscar", MessageBoxButtons.YesNo, _
        MessageBoxIcon.Question) = DialogResult.Yes) Then

        REM CASTEO DE OBJECT A DATAROWVIEW
        vistaFilaActual = CType(TelefonosBindingSource.Current, DataRowView)

        REM BORRO LA FILA DEL DATAGRIDVIEW
        vistaFilaActual.Row.Delete()

        REM MUESTRO LA POSICION
        MostrarPosicion()

    End If

```

7. Finalmente, el botón Buscar permitirá al usuario buscar un registro determinado a partir del actual, utilizando el método *Find* de *DataRowCollection* o el método *Select* de *DataTable* y código SQL.

```

Private Sub btBuscar_Click(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles btnBuscar.Click

    Dim miTabla As DataTable = DsTelefonos.telefonos
    Dim colFilas As DataRowCollection = miTabla.Rows
    REM DECLARO UN ARRAY DE DATAROW
    Dim filaBuscada() As DataRow

    Dim NL As String = Environment.NewLine

    REM BUSCO EN LA COLUMNA NOMBRE DE CADA FILA
    Dim criterio As String = "Nombre Like '%" & txtBuscar.Text & "%'"

    REM UTILIZO EL METODO SELECT PARA ENCONTRAR TODAS LAS FILAS
    REM QUE PASEN EL FILTRO Y LAS ALMACENO EN EL ARRAY DE DATAROW
    filaBuscada = miTabla.Select(criterio)

    REM SI EL LIMITE SUPERIOR DEL ARRAY ES -1...
    If (filaBuscada.GetUpperBound(0) = -1) Then
        MessageBox.Show("No se encontraron registros coincidentes", "Buscar")
        Exit Sub
    End If

```

```

Dim i, j As Integer
REM RECORRO TODAS LAS COINCIDENCIAS...
For i = 0 To filaBuscada.GetUpperBound(0)
    If (MessageBox.Show("¿Es este el nombre buscado?" & NL & _
        filaBuscada(i)(0).ToString & NL, "Buscar", _
        MessageBoxButtons.YesNo) = DialogResult.Yes) Then

        REM SI ES EL NOMBRE, COMPLETO EL RESTO DE LOS CAMPOS
        For j = 0 To colFilas.Count - 1
            If (colFilas(j).Equals(filaBuscada(i))) Then
                TelefonosBindingSource.Position = j
                MostrarPosicion()
            End If
        Next j
    Exit For
End If
Next i
End Sub

```

8. Fin del ejercicio.