

Acceso Desconectado a Base de Datos (Utilizando DataBinding)

Descripción:

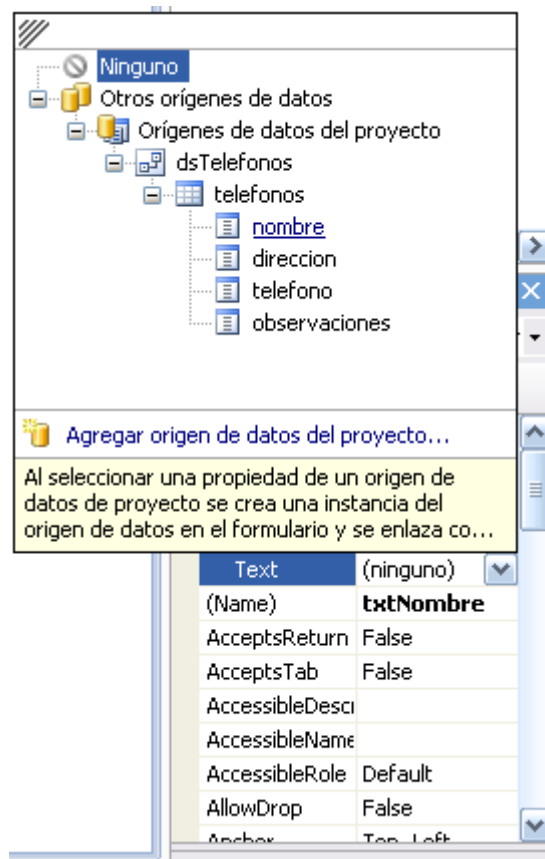
- Realizar una aplicación de ABM contra la base de datos bd_telefonos.mdb.
- Se visualizará en el formulario cada uno de los campos del registro seleccionado, permitiendo modificar cualquiera de ellos.
- Los campos nombre, dirección y teléfono se visualizarán en textBoxes y el campo observaciones en 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 void btnPrimero_Click(object sender, EventArgs e)
{
    // this.telefonosBindingSource.Position = 0;
    this.telefonosBindingSource.MoveFirst();

    this.MostrarPosicion();
}
```

```
private void btnAnterior_Click(object sender, EventArgs e)
{
    // this.telefonosBindingSource.Position -= 1;
    this.telefonosBindingSource.MovePrevious();

    this.MostrarPosicion();
}

private void btnSiguiente_Click(object sender, EventArgs e)
{
    // this.telefonosBindingSource.Position += 1;
    this.telefonosBindingSource.MoveNext();

    this.MostrarPosicion();
}

private void btnUltimo_Click(object sender, EventArgs e)
{
    //this.telefonosBindingSource.Position = this.telefonosBindingSource.Count - 1;
    this.telefonosBindingSource.MoveLast();

    this.MostrarPosicion();
}
```

Añadir, borrar y buscar datos.

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

```

private void btnAgregar_Click(object sender, EventArgs e)
{
    // Creo una tabla de tipo telefonos
    DataTable miTabla = this.bd_telefonosDataSet.telefonos;

    // Creo una colección de filas
    DataRowCollection colFilas = miTabla.Rows;

    // Creo la nueva fila
    DataRow nuevaFila;

    try
    {
        // Le doy el esquema de la tabla
        nuevaFila = miTabla.NewRow();

        // Genero datos por defecto
        nuevaFila[0] = "Alberto";
        nuevaFila[1] = "Av.Mitre 750";
        nuevaFila[2] = "1548723365";
        nuevaFila[3] = "Programador";

        // Agrego la fila a la colección
        colFilas.Add(nuevaFila);

        // Genera un click sobre un botón
        this.btnUltimo.PerformClick();

        // Muestro la nueva posición

        // Pongo en foco al nombre
        this.txtNombre.Focus();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error " + ex.Message);
    }
}

```

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

```

private void btnBorrar_Click(object sender, EventArgs e)
{
    // Creo una fila
    DataRowView vistaFilaActual;
    String NL = Environment.NewLine;

    // Si se desea borrar la marco como delete
    if (MessageBox.Show("Desea borrar este registro?" + NL, "Buscar", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
    {
        // Casteo de Object a DataRowView
        vistaFilaActual = (DataRowView)this.telefonosBindingSource.Current;

        // Borro la fila del DataGridView
        vistaFilaActual.Row.Delete();

        // Muestro la posición
    }
}

```

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 void btnBuscar_Click(object sender, EventArgs e)
{
    DataTable miTabla = this.bd_telefonosDataSet.telefonos;

    // Creo una colección de filas
    DataRowCollection colFilas = miTabla.Rows;

    // Declaro un array de DataRow
    DataRow[] filaBuscada;

    String NL = Environment.NewLine;

    // Busco en la columna nombre de cada fila
    String criterio = "nombre like '* " + this.txtBuscar.Text + " *'";

    // Utilizo el método select para encontrar todas las filas que pasen el filtro
    // y las almaceno en el array de DataRow
    filaBuscada = miTabla.Select(criterio);

    // Si el limite superior del array es -1...
    if (filaBuscada.GetUpperBound(0) == -1)
    {
        MessageBox.Show("No se encontraron registros coincidentes", "Buscar");
    }
}

```

```

// Recorro todas las coincidencias
for (int i = 0; i < filaBuscada.GetUpperBound(0); i++)
{
    if (MessageBox.Show("Este es el nombre buscado?" + NL + filaBuscada[i][0].ToString() + NL, "Buscar",
        MessageBoxButtons.YesNo) == System.Windows.Forms.DialogResult.Yes)
    {
        // Si es el nombre completo el resto de los campos
        for (int j = 0; j < colFilas.Count; j++)
        {
            if (colFilas[j].Equals(filaBuscada[i]))
            {
                this.telefonosBindingSource.Position = j;

                // MostrarPosicion();

            }
        }
    }
}
}

```

8. Fin del ejercicio.