

Ejercicios Lección 2

Interfaz Básica de Usuario



1.- Ejemplo centrar un formulario:

Se trata de hacer un formulario de forma que este se visualice redimensionado con las siguientes características:

- ✓ Una anchura igual a la mitad de la anchura de la pantalla.
- ✓ Una altura igual a la mitad de la altura de la pantalla.
- ✓ Lo sitúe centrado centrado en la pantalla.

```
Private Sub Form_Load()
```

```
    ScaleMode = vbTwips
```

```
    ' la nueva anchura del Formulario es igual a la mitad de la anchura de la  
    ' pantalla usamos la propiedad Width
```

```
    Width = Screen.Width / 2
```

```
    ' la nueva altura del formulario es igual a la mitad de la altura de la  
    ' pantalla usamos la propiedad Height
```

```
    Height = Screen.Height / 2
```

```
    ' Centrado horizontal = (Ancho pantalla - Ancho Formulario)/2
```

```
    Left = (Screen.Width - Width)/2
```

```
    ' Centrado vertical
```

```
    Top = (Screen.Height - Height) / 2
```

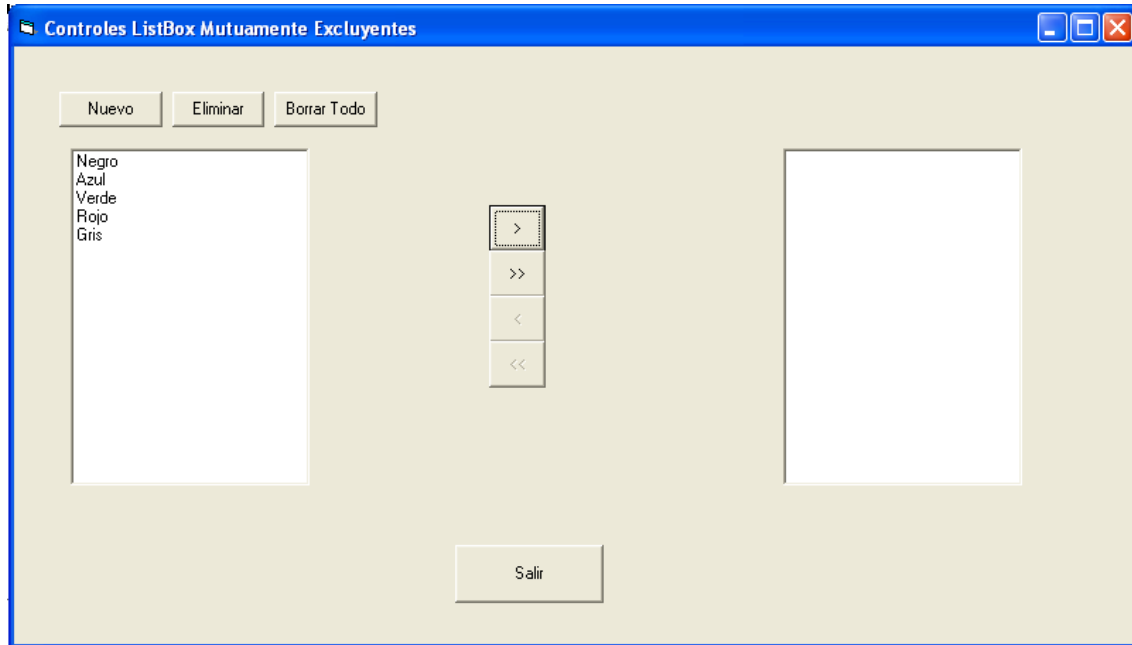
```
End Sub
```



2.- Ejemplo de Listas

A continuación, proponemos la resolución por pasos de un ejemplo práctico sobre el manejo de listas mediante el control ListBox.

Queremos llegar a tener este formulario:



Este ejemplo práctico consiste en programar una aplicación con las siguientes características:

- Que contenga un formulario con dos listas (lstDcha y lstIzda) y cuatro botones.
- Dos de los botones sirven para pasar los valores de lstDcha a lstIzda.
- Los otros dos botones sirven para pasar los valores de lstIzda a lstDcha.

PASOS	FUNCIONES A REALIZAR										
1	Nuevo Proyecto con un formulario										
2	Añadimos dos ListBox: lstIzda y lstDcha										
3	Les modificamos la propiedad MultiSelect: 1 - Simple										
4	Añadimos los cuatro botones y cambiamos las propiedades <i>Name</i> y <i>Caption</i> según los siguientes valores <table><thead><tr><th><i>Name</i></th><th><i>Caption</i></th></tr></thead><tbody><tr><td><i>cmdMover</i></td><td>></td></tr><tr><td><i>cmdMoverTodo</i></td><td>>></td></tr><tr><td><i>cmdAtras</i></td><td><</td></tr><tr><td><i>cmdAtrasTodo</i></td><td><<</td></tr></tbody></table>	<i>Name</i>	<i>Caption</i>	<i>cmdMover</i>	>	<i>cmdMoverTodo</i>	>>	<i>cmdAtras</i>	<	<i>cmdAtrasTodo</i>	<<
<i>Name</i>	<i>Caption</i>										
<i>cmdMover</i>	>										
<i>cmdMoverTodo</i>	>>										
<i>cmdAtras</i>	<										
<i>cmdAtrasTodo</i>	<<										
5	Escribimos el evento Load Private Sub Form_Load() Dim i As Long, mat As Variant ' carga una lista de nombres										

	<pre> mat = Array("Negro", "Azul", "Verde", "Rojo", "Gris") For i = 0 To UBound(mat) lstIzda.AddItem mat(i) Next DefEstadoBoton End Sub </pre>
6	<pre> Escribimos el evento Click del botón cmdMover ' mueve un elemento de izquierda a derecha ' dos comportamientos diferentes, para cuadros de lista regulares ' o multiseleccionables If lstIzda.MultiSelect = False Or lstIzda.SelCount = 0 Then ' cuadro de lista de selección única If lstIzda.ListIndex >= 0 Then lstDcha.AddItem lstIzda.Text lstIzda.RemoveItem lstIzda.ListIndex End If Else ' cuadro de lista de selección múltiple ' observe cómo avanzamos el contador Dim i As Long Do While i < lstIzda.ListCount If lstIzda.Selected(i) Then lstDcha.AddItem lstIzda.List(i) lstIzda.RemoveItem i Else i = i + 1 End If Loop End If DefEstadoBoton End Sub </pre>
7	<pre> Escribimos el evento Click del botón cmdMoverTodo Private Sub cmdMoverTodo_Click() ' mueve todos los elementos de izquierda a derecha Do While lstIzda.ListCount lstDcha.AddItem lstIzda.List(0) lstIzda.RemoveItem 0 Loop DefEstadoBoton End Sub </pre>
8	<pre> Escribimos el evento Click del botón cmdAtras ' mueve un elemento de derecha a izquierda ' dos comportamientos diferentes, para cuadros de lista </pre>

	<p>regulares o multiseleccionables</p> <pre> If IstIzda.MultiSelect = False Or IstDcha.SelCount = 0 Then ' cuadro de lista de selección única If IstDcha.ListIndex >= 0 Then ' cuadro de lista de selección única IstIzda.AddItem IstDcha.Text IstDcha.RemoveItem IstDcha.ListIndex End If Else ' cuadro de lista de selección múltiple ' observe cómo avanzamos el contador Dim i As Long Do While i < IstDcha.ListCount If IstDcha.Selected(i) Then IstIzda.AddItem IstDcha.List(i) IstDcha.RemoveItem i Else i = i + 1 End If Loop End If DefEstadoBoton End Sub </pre>
9	<p>Escribimos el evento Click del botón cmdAtrasTodo</p> <pre> Private Sub cmdAtrasTodo_Click() ' mueve todos los elementos de derecha a izquierda Do While IstDcha.ListCount IstIzda.AddItem IstDcha.List(0) IstDcha.RemoveItem 0 Loop DefEstadoBoton End Sub </pre>
10	<p>Programamos una función que:</p> <p>Si una lista no tiene elementos el botón correspondiente para pasar valores aparezca deshabilitado.</p> <p>Si una lista tiene elementos los botones aparecen habilitados.</p> <p>Llamamos a esta función cada vez que pasamos valores para refrescar el estado de los botones y también al cargar el formulario.</p> <pre> Private Sub IstIzda_Click() DefEstadoBoton End Sub Private Sub IstIzda_DblClick() ' simula una pulsación en el botón cmdMover.Value = True End Sub Private Sub IstDcha_Click() </pre>

	<pre> DefEstadoBoton End Sub Private Sub lstDcha_DbIclick() ' simula una pulsación en el botón cmdAtras.Value = True End Sub Private Sub DefEstadoBoton() ' define el estado activado para todos los botones ' este código tiene en cuenta tanto a los cuadros de lista ' regulares como multiseleccionables cmdMover.Enabled = (lstIzda.ListCount > 0 And lstIzda.ListIndex <> -1 Or lstIzda.SelCount > 0) cmdMoverTodo.Enabled = (lstIzda.ListCount > 0) cmdAtras.Enabled = (lstDcha.ListCount > 0 And lstDcha.ListIndex <> -1 Or lstDcha.SelCount > 0) cmdAtrasTodo.Enabled = (lstDcha.ListCount > 0) End Sub </pre>
11	<p>Añadimos un botón para Salir cmdOk</p> <p>Y en su evento Click</p> <pre> Private Sub cmdOK_Click() Unload Me End Sub </pre>
12	<p>Ahora mejoramos un poco el programa añadiendo tres botones en la lista de la izquierda para añadir un nuevo elemento (cmdNuevo), eliminar uno (cmdBorrar) y borrar todo (cmdBorrarTodo)</p>
13	<p>Escribimos los eventos click de estos botones</p> <pre> Private Sub cmdBorrar_Click() If lstIzda.SelCount = 0 Then MsgBox "No ha seleccionado nada. Seleccione el elemento a borrar", vbExclamation, "Eliminar elemento" Exit Sub End If If lstIzda.MultiSelect = False Or lstIzda.SelCount = 0 Then ' cuadro de lista de selección única </pre>

```
    If lstIzda.ListIndex >= 0 Then
'        lstDcha.AddItem lstIzda.Text
        lstIzda.RemoveItem lstIzda.ListIndex
    End If

Else
' cuadro de lista de selección múltiple
' observe cómo avanzamos el contador

    Dim i As Long
    Do While i < lstIzda.ListCount
        If lstIzda.Selected(i) Then
'            lstDcha.AddItem lstIzda.List(i)
            lstIzda.RemoveItem i
        Else
            i = i + 1
        End If
    Loop
End If

DefEstadoBoton

End Sub

Private Sub cmdBorrarTodo_Click()
If lstIzda. ListCount= 0 Then
    MsgBox "No hay elementos"
    Exit Sub
End If
```

```
Me.lstIzda.Clear
```

```
DefEstadoBoton
```

```
End Sub
```

```
Private Sub cmdNuevo_Click()
```

```
Dim nuevo As String
```

```
nuevo = InputBox("Escribe el valor", "Nuevo valor de la lista")
```

```
If nuevo <> "" Then
```

```
    lstIzda.AddItem nuevo
```

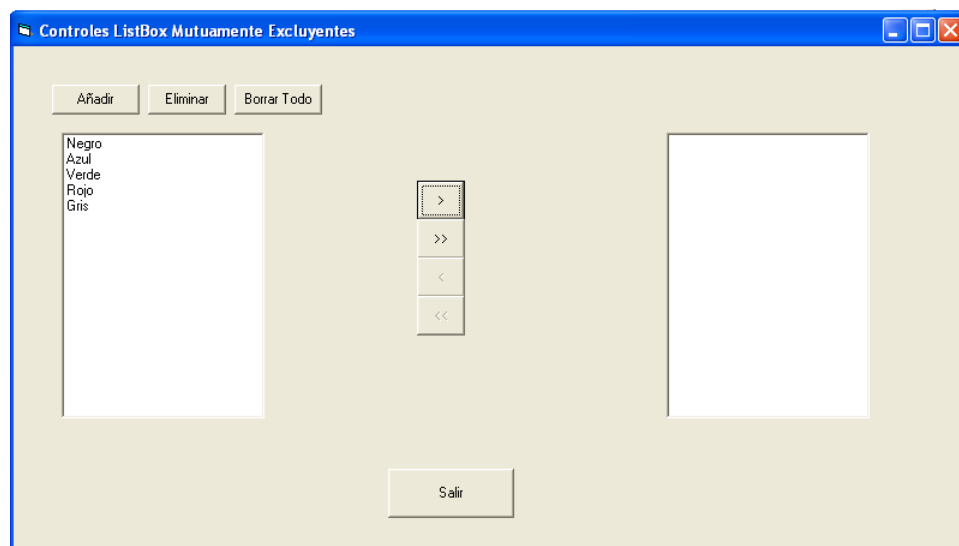
```
End If
```

```
    DefEstadoBoton
```

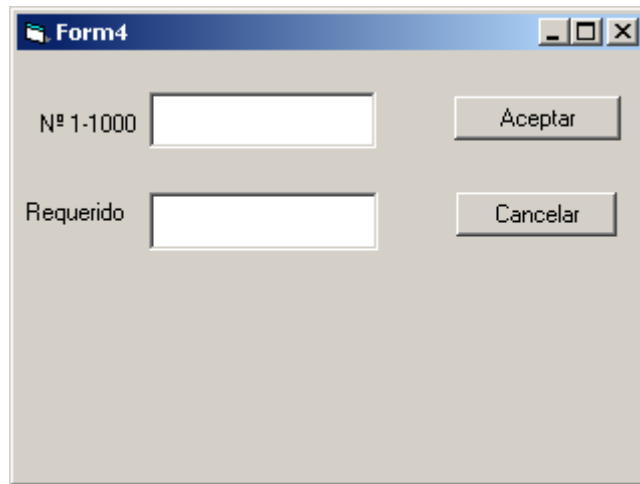
```
End Sub
```

14

Ejecutamos para probar todo



3.- Ejemplo para validar los campos de un formulario:



- Que contenga un formulario, con nombre *form4*, con dos cajas de texto (*txtNecesario*) y (*txtNumerico*) y dos botones (*cmdOk* y *cmdCancel*).
- Al pulsar el botón *cmdOk* validamos:
 - Que la caja de texto *TxtNecesario* no esté vacía
 - Que la caja de texto *TxtNumerico* contenga un valor numérico, en caso contrario exigimos al usuario que introduzca estos datos.
- Al pulsar el botón *cmdCancel* descargamos el formulario sin validar el contenido de la caja de texto.
- Al cerrar el formulario pulsando el botón cerrar de la barra de título validamos el contenido de las cajas de texto.

Pasos	Funciones a realizar												
1	<p data-bbox="507 1256 1406 1317">Añadimos un formulario, y agregamos dos controles TextBox y dos botones y dos etiquetas.</p> <p data-bbox="408 1357 1437 1417">En la Ventana de Propiedades cambiamos las propiedades <i>Name</i> y <i>Caption</i> y <i>TabIndex</i> de los controles según los siguientes valores.</p> <table border="1" data-bbox="309 1458 1299 1588"> <thead> <tr> <th data-bbox="309 1458 539 1491">Control</th> <th data-bbox="715 1458 799 1491">Name</th> <th data-bbox="948 1458 1062 1491">Caption</th> <th data-bbox="1155 1458 1299 1491">TabIndex</th> </tr> </thead> <tbody> <tr> <td data-bbox="309 1525 539 1559">CommandButton</td> <td data-bbox="715 1525 799 1559">cmdOK</td> <td data-bbox="948 1525 1062 1559">Aceptar</td> <td data-bbox="1206 1525 1230 1559">2</td> </tr> <tr> <td data-bbox="309 1559 539 1592">CommandButton</td> <td data-bbox="715 1559 863 1592">cmdCancel</td> <td data-bbox="948 1559 1062 1592">Cancelar</td> <td data-bbox="1206 1559 1230 1592">3</td> </tr> </tbody> </table>	Control	Name	Caption	TabIndex	CommandButton	cmdOK	Aceptar	2	CommandButton	cmdCancel	Cancelar	3
Control	Name	Caption	TabIndex										
CommandButton	cmdOK	Aceptar	2										
CommandButton	cmdCancel	Cancelar	3										

	<p>TextControl txtNecesario 1 TextControl txtNumerico 0 Label Label1 Nº 1 - 1000 Label Label2 Requerido</p>															
2	<p>En la Ventana de Propiedades cambiamos la propiedad CausesValidation para que salte el evento Validate si está a True CausesValidation.</p> <table border="1"> <thead> <tr> <th>Control</th> <th>Name</th> <th>Propiedad:CausesValidation</th> </tr> </thead> <tbody> <tr> <td>CommandButton</td> <td>cmdOK</td> <td>True</td> </tr> <tr> <td>CommandButton</td> <td>cmdCancel</td> <td>False</td> </tr> <tr> <td>TextControl</td> <td>txtNumerico</td> <td>True</td> </tr> <tr> <td>TextControl</td> <td>txtNecesario</td> <td>True</td> </tr> </tbody> </table>	Control	Name	Propiedad:CausesValidation	CommandButton	cmdOK	True	CommandButton	cmdCancel	False	TextControl	txtNumerico	True	TextControl	txtNecesario	True
Control	Name	Propiedad:CausesValidation														
CommandButton	cmdOK	True														
CommandButton	cmdCancel	False														
TextControl	txtNumerico	True														
TextControl	txtNecesario	True														
3	<p>Programamos el suceso <i>Validate</i> de los dos controles TextBox para que cumplan las restricciones enunciadas.</p> <pre> Private Sub txtNecesario_Validate(Cancel As Boolean) ' verifica que el campo no está vacío If txtNecesario.Text = "" Then MsgBox "Por favor, escriba algo aquí", vbExclamation Cancel = True End If End Sub Private Sub txtNumerico_Validate(Cancel As Boolean) If Not IsNumeric(txtNumerico.Text) Then Cancel = True ElseIf CDbI(txtNumerico.Text) < 1 Or CDbI(txtNumerico.Text) > 1000 Then Cancel = True End If If Cancel Then MsgBox "Por favor, escriba un número perteneciente al rango[1-1000]",vbExclamation End If End Sub </pre>															
4	<p>Añadimos las siguiente función al modulo.bas del proyecto. En ella vamos situando el foco en cada control del formulario y con el método ValidateControls del objeto formulario hacemos saltar el evento Validate de cada control que ya hemos programado</p> <pre> Public Function validar(frm As Form) As Boolean On Error GoTo Trat_error validar = True Dim ctrl As Control ' proporciona el foco a cada control del formulario luego lo valida With frm For Each ctrl In .Controls .ctrl.SetFocus .ValidateControls Next End With Exit Function </pre>															

	<pre>Trat_error: Select Case Err.Number 'Si se produce el error 380 devuelve el valor <i>false</i> y termina la función Case 380 validar = False 'Si se produce el error 438 la función no termina se obvia el error y se ejecuta la siguiente instrucción de la función. Este error se produce porque las etiquetas Label no admiten que se les asigne el foco con SetFocus. Case 438 Resume Next 'Si se produce cualquier otro error termina la función y se propaga el error a la pila de llamadas. Case Else Err.Raise Err.Number, "validar", Err.Description End Select End Function</pre>
5	<p>Programamos los botones aceptar y cancelar</p> <pre>Private Sub cmdcancelar_Click() Unload Me End Sub Private Sub cmdaceptar_Click() Dim frm As Form4 Set frm = Me txtNecesario.SetFocus If validar(frm) Then Unload Me End If</pre>