



Anexos

## ANEXOS

(En apoyo a las actividades de aprendizaje de los temas 4 y 5)

### 4.1 Introducción a clases

- Agregar el generador de clases **Class Builder** al proyecto
  1. Iniciar un proyecto **Standard EXE Project**.
  2. En el menú **Add-Ins**, dar seleccionar a la opción **Add-In Manager**.
  3. De la lista disponible, seleccionar **VB 6 ClassBuilder Utility**.
  4. De la opción **Load Behavior**, seleccionar el check box **Loaded/Unloaded** y presionar el botón **OK**.
  5. En el menú **Add-Ins**, seleccionar **Class Builder Utility**, esto abrirá la ventana del **Class Builder**.
  
- Generar la clase usando el Class Builder
  6. En el **Class Builder**, seleccionar menú **File**, seleccionar **New** y después **Class**.
  7. En la ventana **Class Module Builder**, escribir la palabra *HUMANO* en la caja de texto **name** para darle nombre a la clase.
  
- Agregar Propiedades, eventos y métodos con el Class Builder.
  8. En el menú File, seleccionar New →Property.
  9. En la ventana **Property Builder**, escribir la palabra *NOMBRE* en la caja de texto para darle nombre a la propiedad.
  10. En el menú File, seleccionar New →Property.



11. En la ventana **Property Builder**, escribir la palabra *EDAD* en la caja de texto para darle nombre a la propiedad.
  12. En el menú File, seleccionar New →Method.
  13. En la ventana **Method Builder**, escribir la palabra *VERNOMBRE* en la caja de texto para darle nombre al método
  14. En el menú File, seleccionar New →Event
  15. En la ventana **Event Builder**, escribir la palabra *CORRER* en la caja de texto para darle nombre al evento.
  16. Cerrar el Class Builder y actualizar el proyecto.
  17. En el módulo de código para la Clase HUMANO, agregar código al método *VERNOMBRE* con la siguiente sintaxis:
    18. Public Sub *VERNOMBRE*()
    19. MsgBox "Nombre:" & mvarnombre
    20. End Sub
  21. Nombrar el proyecto **PracticaClase** y salvar el trabajo.
- Verificar la creación de la clase con el uso del Object Browser
    1. En el menú **View**, clic sobre **Object Browser**.
    2. En la caja **Project/Library**, seleccionar **PracticaClase**.
    3. Seleccionar **HUMANO** de la lista de clases.
    4. Seleccionar propiedad Nombre.
    5. Salvar el trabajo.

#### Uso de la clase

6. En el formulario agregar dos cajas de texto y un botón
7. Dar doble click en el botón para activar la ventana de código. Se despliega el código del evento click, ahí se instancia la clase humano con las siguientes instrucciones:



Dim hmn As HUMANO

Set hmn = New HUMANO

8. En la propiedad NOMBRE del objeto hmn colocar lo que esté escrito en la caja de texto:
9. hmn.nombre = Text1
10. Invocar el método
11. hmn.VERNOMBRE
12. Salvar el proyecto y correr la aplicación.

Form2

nombre

edad

aceptar



## 4.2 Data bound class module

- Crear la ClaseOrigen.
  1. Iniciar un proyecto **Standard EXE Project**.
  2. En el menú **Project→References**, activar la referencia **Microsoft ActiveXData Object2.0**
  3. En el menú **Project**, seleccionar **Add Module Class**.
  4. En la propiedad **Name** del Módulo de clase poner el nombre **ClaseOrigen**.
  
- Colocar la propiedad **DataSourceBehavior**
  5. En la propiedad **DataSourceBehavior** de la clase **ClaseOrigen** poner el valor **vbDataSource**.

Se creará un evento que se llama **GetDataMember**.
  
- .Declarar las variables de tipo objeto *connection* y *recordset* en la zona de declaraciones del módulo de clase.
  6. Private conexión as connection.
  7. Private datos as recordset.
  
- En el evento Initialize de la clase, establecer las cadenas de conexión
  8. Dim cadena as string
  9. Cadena = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BIBLIO.MDB"
  10. Set Conexion= New connection
  11. Set datos = New recordset
  12. Conexion.open cadena
  13. Registros.open "select \* from Authors", conexion,adOpenStatic
  14. Escribir el siguiente código en el evento GetData Member

Set Data = datos



- Agregar funcionalidad a la clase con la creación del método SigAuthor

```
Private Sub SigAuthor()  
datos.MoveNext  
End Sub
```

- Ligar la Clase a un proveedor de datos en un formulario

15. Agregar un formulario al proyecto.

16. Agregar una caja de texto y dos botones al formulario.

17. Colocar la referencia a Microsoft Data Binding Collection



1. Declara una variable de tipo objeto para BindingCollection  
Dim Bc as BindingCollection
2. Declara una variable tipo objeto para la clase ClaseOrigen
3. Dim clase as ClaseOrigen

- Agregar funcionalidad y ligar la clase con los objetos

En el evento clic del botón Activar:

4. Instanciar los dos objetos

```
Set Bc = New BindingCollection
```



Set clase = New ClaseOrigen

5. Ligar la propiedad DataSource al objeto Binding Collection.

Set Bc.DataSource = clase

6. Ligar el control agregando un binding al objeto Binding Collection.

Bc.Add text1, "Text", "Campo"

- Dar movilidad a los datos. En el evento Clic del botón Siguiente

7. Activar el método SigAutor de la claseOrigen.

clase.SigAutor

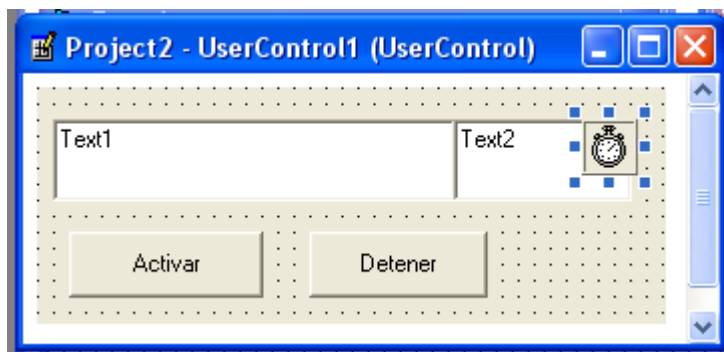
8. Salvar el proyecto y correr la aplicación.



## 5.1 Construcción de aplicaciones con Active-X Control

- Crear el control ActiveX
  1. Iniciar un proyecto **Active X control**.
  2. Nombrar al proyecto como MiControl.
  3. Cambiar la propiedad **NAME** del **UserControl** y poner el nombre **RELOJ**.
  4. En la plantilla **User Control** agregar los siguientes objetos:

Objeto	Propiedad	Valor
<b>Text</b>	<b>Text</b>	“”
<b>Text</b>	<b>Text</b>	“”
<b>Timer</b>	<b>Interval</b>	<b>1000</b>
<b>Command</b>	<b>Caption</b>	<b>Activar</b>
	<b>Name</b>	<b>Activar</b>
<b>Command</b>	<b>Caption</b>	<b>Detener</b>
	<b>Name</b>	<b>Detener</b>



- Crear propiedades y procedimientos para el nuevo control
  5. En el evento **Timer()** del control Timer, escribir el siguiente código

```
Private Sub Timer1_Timer()
```

```
Text2 = Time()
```



*End Sub*

6. En el evento clic del botón **Activar**, escribir lo siguiente:

*Private Sub Activar\_Click()*

*Timer1.Enabled = True*

*End Sub*

7. En el evento clic del botón **Detener**, escribir lo siguiente.

*Private Sub Detener\_Click()*

*Timer1.Enabled = False*

*End Sub*

8. Generar una propiedad **Letrero** usando la propiedad **text** de un objeto **text**.

*Public Property Get Letrero() As String*

*Letrero = Text1.Text*

*End Property*

*Public Property Let Letrero(ByVal NewText As String)*

*Text1.Text = NewText*

*PropertyChanged "Text"*

*End Property*

9. Salvar el proyecto MiControl.

10. Cerrar el UserControl.





Probar el control

11. Agregar un proyecto Estándar EXE.
12. De la barra de herramientas Estándar, seleccionar el objeto **Reloj** y colocarlo en el formulario.
13. En el evento Load del formulario, escribir el siguiente código:  

```
Private Sub Form_Load()  
Reloj1.Letrero = "Buen dia"  
End Sub
```
14. Ratón derecho sobre el proyecto Estándar y seleccionar del menú la opción **Set as Start Up**.
15. Correr la aplicación.