

# *GUÍA DE LABORATORIO*

## *Parte I*

*ELABORADO POR*  
*CARLOS CASTILLO PERALTA*  
[cc\\_peralta@hotmail.com](mailto:cc_peralta@hotmail.com)

## INTRODUCCIÓN

La presente Guía de Laboratorio de Microsoft Visual Basic, se elaboró con la finalidad de complementar la parte teórica con la parte práctica del curso. Los problemas presentan un grado de dificultad gradual, es decir, en cada práctica primero se comienzan con problemas muy sencillos con la finalidad de que el estudiante esclarezca dudas y quede claro en los conceptos asimilados en la teoría, y luego en los problemas posteriores se va incrementando el nivel de dificultad.

Esta guía explora una gran variedad de materias y contiene muchos problemas recopilados a lo largo de varios años de experiencia. Espero que satisfaga las necesidades de los estudiantes y colegas, ya que ha sido elaborada teniendo en cuenta los objetivos de aprendizaje del curso.

Si desean compartirla con otros colegas, me agradecería mucho que lo hicieran, con la única condición de que siempre se respete el derecho del autor.

Desde ya les deseo la mejor de las suertes en el aprendizaje de Microsoft Visual Basic.

Lima, Diciembre del 2000

*Carlos Castillo Peralta*

# Microsoft **Visual Basic**

## GUÍA DE LABORATORIO N° 1

### Objetivos

Luego de completar este laboratorio, el estudiante será capaz de:

- Identificar los elementos del entorno integrado de desarrollo de Microsoft Visual Basic 6.0.
- Añadir controles a un formulario.
- Establecer las propiedades de los controles.
- Trabajar con procedimientos de eventos.
- Crear una aplicación simple con Microsoft Visual Basic.

### Consideraciones

Para el desarrollo del presente laboratorio Ud. deberá crear una carpeta **C:\FundVB\Lab1**, para guardar sus trabajos correspondientes a este laboratorio.

### Aplicación N° 1

Elaborar una aplicación que permita calcular el salario neto de un trabajador en función del número de horas trabajadas, pago por hora de trabajo y un descuento fijo al sueldo bruto del 20 por 100. Los cálculos a efectuar para obtener el salario neto de un trabajador se muestran a continuación:

$$\text{SalarioBruto} = \text{HorasTrabajadas} * \text{PagoPorHora}$$
$$\text{Descuento} = 0.2 * \text{SalarioBruto}$$
$$\text{SalarioNeto} = \text{SalarioBruto} - \text{Descuento}$$

El diseño de la interfaz debe ser similar a la figura mostrada:

Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 2 marcos
- 6 etiquetas
- 6 cajas de texto
- 3 botones de comando

Una vez ubicados los controles, establezca las propiedades según se indica:

**Form1**

Nombre	FrmSalarioNeto
BorderStyle	3-Fixed Dialog
Caption	Cálculo del salario neto
Moveable	False
StartPosition	2-CenterScreen

**Frame1**

Nombre	FraIngreso
Caption	Ingreso de datos:

**Frame2**

Nombre	FraSalida
Caption	Salida de datos:

**Label11**

Nombre	LblApellidosNombres
AutoSize	True
Caption	Apellidos y nombres:

**Label12**

Nombre	LblHorasTrabajadas
AutoSize	True
Caption	Horas trabajadas:

**Label13**

Nombre	LblPagoPorHora
AutoSize	True
Caption	Pago por hora S/.

**Label14**

Nombre	LblSalarioBruto
AutoSize	True
Caption	Salario bruto S/.

**Label15**

Nombre	LblDescuento
AutoSize	True
Caption	Descuento S/.

**Label16**

Nombre	LblSalarioNeto
AutoSize	True
Caption	Salario neto S/.

**Text1**

Nombre	TxtApellidosNombres
Text	

**Text2**

Nombre	TxtHorasTrabajadas
Text	

**Text3**

Nombre	TxtPagoPorHora
Text	

**Text4**

Nombre	TxtSalarioBruto
Text	

**Text5**

Nombre	TxtDescuento
Text	

**Text6**

Nombre	TxtSalarioNeto
Text	

**Command1**

Nombre	CmdAceptar
Caption	&Aceptar

**Command2**

Nombre	CmdLimpiar
Caption	&Limpiar

**Command3**

Nombre	CmdSalir
Caption	&Salir

Luego de establecer las propiedades para los controles, debe añadir código a la aplicación. Para ello haga doble click sobre el botón Aceptar e ingrese el siguiente código:

```
Private Sub CmdAceptar_Click()  
    Dim HorasTrabajadas As Integer, PagoPorHora As Double  
    Dim SalarioBruto As Double  
    Dim Descuento As Double  
    Dim SalarioNeto As Double  
    HorasTrabajadas = Val(TxtHorasTrabajadas)  
    PagoPorHora = Val(TxtPagoPorHora)  
    SalarioBruto = HorasTrabajadas * PagoPorHora  
    Descuento = 0.2 * SalarioBruto  
    SalarioNeto = SalarioBruto - Descuento  
    TxtSalarioBruto = Str(SalarioBruto)  
    TxtDescuento = Str(Descuento)  
    TxtSalarioNeto = Str(SalarioNeto)  
End Sub
```

A continuación haga doble click sobre el botón Salir y añada el siguiente código:

```
Private Sub CmdSalir_Click()  
    End
```

End Sub

Guarde y luego ejecute la aplicación que acaba de crear. Pruebe ingresando diferentes valores.

Luego, haga doble click sobre el botón Limpiar y añada el siguiente código:

```
Private Sub CmdLimpiar_Click()  
    TxtApellidosNombres = ""  
    TxtHorasTrabajadas = ""  
    TxtPagoPorHora = ""  
    TxtSalarioBruto = ""  
    TxtDescuento = ""  
    TxtSalarioNeto = ""  
    TxtApellidosNombres.SetFocus  
End Sub
```

Guarde y ejecute su aplicación. ¿Cómo afecta el código añadido en el botón Limpiar a la aplicación?. Anote sus observaciones.

## **Aplicación N° 2**

El menú de un restaurante rápido se muestra a continuación:



MENÚ	PRECIO (S/.)
Hamburguesa	2.50
Cerveza	4.00
Gaseosa	3.00
Ensalada	1.50
Salchichas	2.00
Refresco	1.00
Sopa	1.50
Postre	1.50

Se desea construir una aplicación que calcule las ventas totales al final del día, así como los impuestos a pagar (18 por 100).

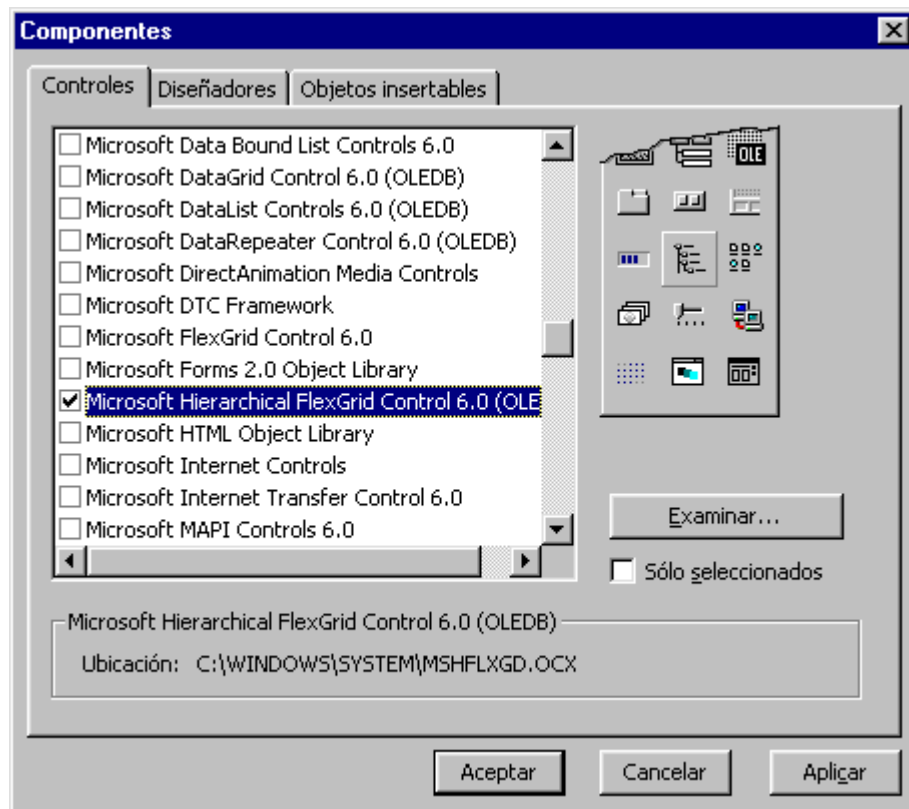
La interfaz de entrada y salida deberá ser similar a la figura mostrada a continuación:

MENÚ	PRECIO (S/.)
Hamburguesa	2.50
Cerveza	4.00
Gaseosa	3.00
Ensalada	1.50
Salchichas	2.00
Refresco	1.00
Sopa	1.50
Postre	1.50

Hamburguesa:   
 Cerveza:   
 Gaseosa:   
 Ensalada:   
 Salchichas:   
 Refresco:   
 Sopa:   
 Postre:

Venta total S/.   
 Impuesto S/.

Para el diseño del cuadro de menú utilizaremos el control Microsoft Hierarchical FlexGrid. Para tal fin, seleccione el Menú Proyecto y elija la opción Componentes:



A continuación active la casilla de verificación Microsoft Hierarchical FlexGrid Control 6.0 OLEDB y haga click sobre el botón Aceptar. En seguida este control se añadirá al Cuadro de Herramientas.

Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 10 etiquetas
- 10 cajas de texto
- 1 control MSFlexGrid
- 3 botones de comando

En seguida, elabore el diseño de entrada y salida. Para ello proceda a establecer las propiedades según se indica a continuación:

#### **Form1**

Nombre	FrmRestaurante
BorderStyle	3-Fixed Dialog
Caption	Restaurante
Moveable	False
StartPosition	2-CenterScreen

**Label11**

Nombre	LblHamburguesa
AutoSize	True
Caption	Hamburguesa:

**Label12**

Nombre	LblCerveza
AutoSize	True
Caption	Cerveza:

**Label13**

Nombre	LblGaseosa
AutoSize	True
Caption	Gaseosa:

**Label14**

Nombre	LblEnsalada
AutoSize	True
Caption	Ensalada:

**Label15**

Nombre	LblSalchichas
AutoSize	True
Caption	Salchichas:

**Label16**

Nombre	LblRefresco
AutoSize	True
Caption	Refresco:

**Label17**

Nombre	LblSopa
AutoSize	True
Caption	Sopa:

**Label18**

Nombre	LblPostre
AutoSize	True
Caption	Postre:

**Label19**

Nombre	LblVentaTotal
Caption	Venta total S/.

**Label110**

Nombre	LblImpuesto
Caption	Impuesto S/.

**MSHFlexGrid1**

Nombre	GrdMenu
Font	Arial (Negrita 10)
FontFixed	Arial (Negrita 10)

**Text1**

Nombre	TxtHamburguesa
Text	

**Text2**

Nombre	TxtCerveza
--------	------------

Text	
------	--

**Text3**

Nombre	TxtGaseosa
Text	

**Text4**

Nombre	TxtEnsalada
Text	

**Text5**

Nombre	TxtSalchichas
Text	

**Text6**

Nombre	TxtRefresco
Text	

**Text7**

Nombre	TxtSopa
Text	

**Text8**

Nombre	TxtPostre
Text	

**Text9**

Nombre	TxtVentaTotal
Locked	True
Text	

**Text10**

Nombre	TxtImpuesto
--------	-------------

Locked	True
Text	

### Command1

Nombre	CmdAceptar
Caption	&Aceptar

### Command2

Nombre	CmdLimpiar
Caption	&Limpiar

### Command3

Nombre	CmdSalir
Caption	&Salir
Picture	C:\Archivos de programa\Microsoft Visual Studio\Common\Graphics\Icons\Arrows\Point04.ico
Style	1-Graphical

En primer lugar debemos cargar los datos a la cuadrícula. Esto lo vamos a realizar en tiempo de ejecución al momento de cargarse en memoria el formulario. Para ello, haga doble click sobre el formulario y añada el siguiente código:

```
Private Sub Form_Load()
    GrdMenu.Cols = 2
    GrdMenu.Rows = 9
    GrdMenu.FixedCols = 0
    GrdMenu.FixedRows = 1
    GrdMenu.TextArray(0) = "Menú"
    GrdMenu.TextArray(1) = "Precio"
    GrdMenu.TextArray(2) = "Hamburguesa"
    GrdMenu.TextArray(3) = "2.50"
    GrdMenu.TextArray(4) = "Cerveza"
```

```

GrdMenu.TextArray(5) = "4.00"
GrdMenu.TextArray(6) = "Gaseosa"
GrdMenu.TextArray(7) = "3.00"
GrdMenu.TextArray(8) = "Ensalada"
GrdMenu.TextArray(9) = "1.50"
GrdMenu.TextArray(10) = "Salchichas"
GrdMenu.TextArray(11) = "2.00"
GrdMenu.TextArray(12) = "Refresco"
GrdMenu.TextArray(13) = "1.00"
GrdMenu.TextArray(14) = "Sopa"
GrdMenu.TextArray(15) = "1.50"
GrdMenu.TextArray(16) = "Postre"
GrdMenu.TextArray(17) = "1.50"

```

End Sub

Luego debemos añadir el código que se encargará de realizar los cálculos. Para tal fin haga doble click sobre el botón Aceptar y proceda a ingresar lo siguiente:

```

Private Sub CmdAceptar_Click()
    Dim Hamburguesa As Integer, Cerveza As Integer
    Dim Gaseosa As Integer, Ensalada As Integer
    Dim Salchichas As Integer, Refresco As Integer
    Dim Sopa As Integer, Postre As Integer
    Dim VentaTotal As Double, Impuesto As Double
    Hamburguesa = Val(TxtHamburguesa)
    Cerveza = Val(TxtCerveza)
    Gaseosa = Val(TxtGaseosa)
    Ensalada = Val(TxtEnsalada)
    Salchichas = Val(TxtSalchichas)
    Refresco = Val(TxtRefresco)
    Sopa = Val(TxtSopa)
    Postre = Val(TxtPostre)
    VentaTotal = Hamburguesa * 2.5 + Cerveza * 4.0 _

```

```

        + Gaseosa * 3.0 + Ensalada * 1.5 + Salchichas * 2.0 _
          + Refresco * 1.0 + Sopa * 1.5 + Postre * 1.5
    Impuesto = 0.18 * VentaTotal
    TxtVentaTotal = Str(VentaTotal)
    TxtImpuesto = Str(Impuesto)
End Sub

```

Luego, haga doble click sobre el botón Limpiar y añada el siguiente código:

```

Private Sub CmdLimpiar_Click()
    TxtHamburguesa = "" : TxtCerveza = ""
    TxtGaseosa = "" : TxtEnsalada = ""
    TxtSalchichas = "" : TxtRefresco = ""
    TxtSopa = "" : TxtPostre = ""
    TxtVentaTotal = "" : TxtImpuesto = ""
    TxtHamburguesa.SetFocus
End Sub

```

A continuación haga doble click sobre el botón Salir y añada el siguiente código:

```

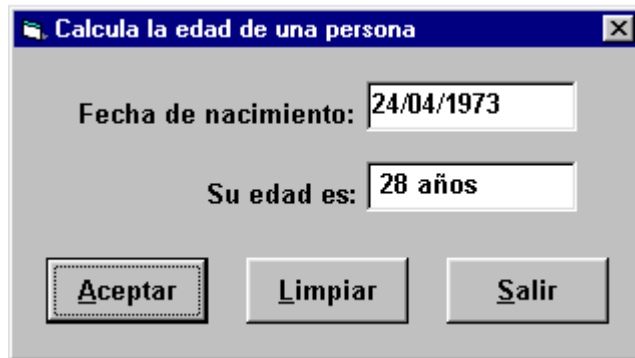
Private Sub CmdSalir_Click()
    End
End Sub

```

### **Aplicación N° 3**

Elaborar una aplicación que permita calcular la edad de una persona a partir de su fecha de nacimiento. El diseño de la interfaz debe ser similar a la figura mostrada:





Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 2 etiquetas
- 2 cajas de texto
- 3 botones de comando

Luego, proceda a establecer las propiedades según se indica a continuación:

#### Form1

Nombre	FrmEdad
BorderStyle	3-Fixed Dialog
Caption	Calcula la edad de una persona

#### Label1

Nombre	LblFecNac
AutoSize	True
Caption	Fecha de nacimiento:

#### Label2

Nombre	LblEdad
AutoSize	True
Caption	Su edad es:

#### Text1

Nombre	TxtFecNac
Text	

### Text2

Nombre	TxtEdad
Locked	True
Text	

### Command1

Nombre	CmdAceptar
Caption	&Aceptar
Default	True

### Command2

Nombre	CmdLimpiar
Caption	&Limpiar

### Command3

Nombre	CmdSalir
Cancel	True
Caption	&Salir

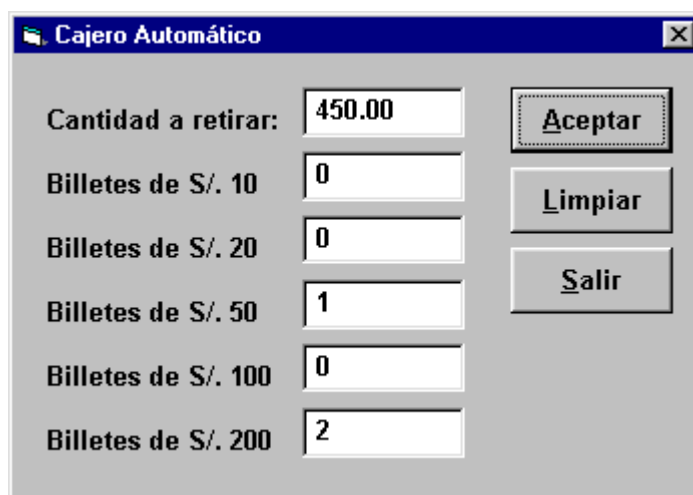
En seguida proceda a ingresar el siguiente código:

```
Private Sub CmdAceptar_Click()
    Dim FecNac As Date, Edad As Integer
    FecNac = CDate(TxtFecNac)
    Edad = CInt((Date - FecNac) / 365)
    TxtEdad = Str(Edad) & " años"
End Sub
```

Se deja como ejercicio para el estudiante el código asociado con los botones Limpiar y Salir, respectivamente.

#### Aplicación N° 4

Se tiene un cajero automático el cual permite manipular cierta cantidad de dinero para lo cual dispone de los siguientes billetes: S/. 10.00, S/. 20.00, S/. 50.00, S/. 100.00 y S/. 200.00. Elaborar una aplicación que permita la lectura de la cantidad a retirar e indique el menor número de billetes a utilizar.



Cantidad a retirar:	450.00	Aceptar
Billetes de S/. 10	0	Limpiar
Billetes de S/. 20	0	Salir
Billetes de S/. 50	1	
Billetes de S/. 100	0	
Billetes de S/. 200	2	

Creo que a estas alturas Ud. ya entendió como funciona esto. Por ello, esta aplicación se deja como ejercicio para el estudiante.

## Microsoft **Visual Basic**

#### GUÍA DE LABORATORIO N° 2

##### Objetivos

Luego de completar este laboratorio, el estudiante será capaz de:

- Implementar aplicaciones utilizando estructuras de control selectivas.

- Usar casillas de verificación y botones de opción para alternar entre uno o más valores.
- Utilizar cajas de diálogo predefinidas para visualizar mensajes.

### **Consideraciones**

Para el desarrollo del presente laboratorio Ud. deberá crear una carpeta **C:\FundVB\Lab2**, para guardar sus trabajos correspondientes a este laboratorio.

### **Aplicación N° 1**

Se desea elaborar la boleta de pago de los trabajadores de una fábrica. Para ello se cuenta con los siguientes datos de entrada:

Apellidos y nombres del trabajador

Sueldo básico

Horas extras trabajadas

Además, se sabe que los trabajadores laboran en dos turnos: diurno y nocturno. Y que la tarifa por las horas extras diurnas es de 10 soles y por las horas extras nocturnas es de 15 soles.

Los descuentos a aplicar son sobre el sueldo básico y se obtienen de la siguiente manera:

$$\text{Renta} = \text{SueldoBasico} * 0.1$$

$$\text{Fonavi} = \text{SueldoBasico} * 0.07$$

$$\text{AFP} = \text{SueldoBasico} * 0.03$$

Finalmente el sueldo neto del trabajador se calcula mediante las siguientes expresiones:

$\text{Ingresos} = \text{SueldoBasico} + \text{HorasExtras} * \text{PagoHoraExtra}$

$\text{Egresos} = \text{Renta} + \text{Fonavi} + \text{AFP}$

$\text{SueldoNeto} = \text{Ingresos} - \text{Egresos}$

El diseño de la interfaz deberá ser similar a la figura mostrada:

Trabajador:	Rojas Babilonia, Julian
Sueldo básico S/.	800.00
Horas extras:	12
Pago hora extra S/.	15
Turno:	<input type="radio"/> Diurno <input checked="" type="radio"/> Nocturno
Descuentos:	
<input checked="" type="checkbox"/> Renta	80
<input type="checkbox"/> Fonavi	0
<input checked="" type="checkbox"/> AFP	24
Sueldo neto S/.	876

Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 2 marcos
- 5 etiquetas
- 8 cajas de texto
- 2 botones de opción
- 3 casillas de verificación
- 3 botones de comando

Luego, proceda a establecer las propiedades según se indica a continuación:

#### **Form1**

Nombre	FrmBoletaDePago
BorderStyle	3-Fixed Dialog
Caption	Boleta de pago
Moveable	False

#### **Label1**

Nombre	LblTrabajador
AutoSize	True
Caption	Trabajador:

#### **Label2**

Nombre	LblSueldoBasico
AutoSize	True
Caption	Sueldo Bruto S/.

#### **Label3**

Nombre	LblHorasExtras
AutoSize	True
Caption	Horas extras:

#### **Label4**

Nombre	LblPagoHoraExtra
AutoSize	True
Caption	Pago hora extra S/.

#### **Label5**

Nombre	LblSueldoNeto
AutoSize	True
Caption	Sueldo neto S/.

**Text1**

Nombre	TxtTrabajador
Text	

**Text2**

Nombre	TxtSueldoBasico
Text	

**Text3**

Nombre	TxtHorasExtras
Text	

**Text4**

Nombre	TxtPagoHoraExtra
BackColor	&H80000004&
Text	

**Text5**

Nombre	TxtRenta
BackColor	&H80000004&
Locked	True
Text	

**Text6**

Nombre	TxtFonavi
BackColor	&H80000004&
Locked	True
Text	

**Text7**

Nombre	TxtAFP
--------	--------

BackColor	&H80000004&
Locked	True
Text	

**Text8**

Nombre	TxtSueldoNeto
BackColor	&H80000004&
Locked	True
Text	

**Option1**

Nombre	OptDiurno
Caption	Diurno
Value	True

**Option2**

Nombre	OptNocturno
Caption	Nocturno
Value	False

**Check1**

Nombre	ChkRenta
Caption	Renta

**Check2**

Nombre	ChkFonavi
Caption	Fonavi

**Check3**

Nombre	ChkAFP
Caption	AFP

**Command1**



Nombre	CmdAceptar
Caption	&Aceptar
Default	True

### Command2

Nombre	CmdLimpiar
Caption	&Limpiar

### Command3

Nombre	CmdSalir
Cancel	True
Caption	&Salir

Una vez establecidas las propiedades, proceda a ingresar el código que se indica a continuación:

```
Private Sub CmdAceptar_Click()
    Dim SueldoBasico As Double
    Dim HorasExtras As Integer, PagoHoraExtra As Double
    Dim Renta As Double, Fonavi As Double, AFP As Double
    Dim Ingresos As Double, Egresos As Double
    Dim SueldoNeto As Double
    SueldoBasico = Val(TxtSueldoBasico)
    HorasExtras = Val(TxtHorasExtras)
    If OptDiurno Then
        PagoHoraExtra = 10
    End If
    If OptNocturno Then
        PagoHoraExtra = 15
    End If
    If ChkRenta.Value Then
        Renta = SueldoBasico * 0.1
    End If
End Sub
```

```

Else
    Renta = 0
End If
If ChkFonavi.Value Then
    Fonavi = SueldoBasico * 0.07
Else
    Fonavi = 0
End If
If ChkAFP.Value Then
    AFP = SueldoBasico * 0.03
Else
    AFP = 0
End If
Ingresos = SueldoBasico + HorasExtras * PagoHoraExtra
Egresos = Renta + Fonavi + AFP
SueldoNeto = Ingresos - Egresos
TxtPagoHoraExtra = Str(PagoHoraExtra)
TxtRenta = Str(Renta)
TxtFonavi = Str(Fonavi) : TxtAFP = Str(AFP)
TxtSueldoNeto = Str(SueldoNeto)
End Sub

Private Sub CmdSalir_Click()
    End
End Sub

```

## **Aplicación N° 2**

Cuatro notas entre 0 y 20 representan las calificaciones de un estudiante de un curso de programación. Elaborar una aplicación para obtener el promedio de esas calificaciones y visualizar su puntuación de acuerdo al siguiente cuadro:

Promedio	Puntuación
----------	------------

19-20	A
16-18	B
11-15	C
6-10	D
0-5	E

El diseño de la interfaz deberá ser similar a la figura mostrada:

Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 7 etiquetas
- 7 cajas de texto
- 3 botones de comando

Luego, proceda a establecer las propiedades según se indica a continuación:

**Form1**

Nombre	FrmNotas
BorderStyle	3-Fixed Dialog
Caption	Puntuación de un estudiante

**Label11**

Nombre	LblAlumno
AutoSize	True
Caption	Alumno:

**Label12**

Nombre	LblN1
AutoSize	True
Caption	Nota 1:

**Label13**

Nombre	LblN2
AutoSize	True
Caption	Nota 2:

**Label14**

Nombre	LblN3
AutoSize	True
Caption	Nota 3:

**Label15**

Nombre	LblN4
AutoSize	True
Caption	Nota 4:

**Label16**

Nombre	LblPromedio
AutoSize	True
Caption	Promedio:

**Label17**

Nombre	LblPuntuacion
AutoSize	True
Caption	Puntuación:

**Text1**

Nombre	TxtAlumno
Text	

**Text2**

Nombre	TxtN1
Text	

**Text3**

Nombre	TxtN2
Text	

**Text4**

Nombre	TxtN3
Text	

**Text5**

Nombre	TxtN4
Text	

**Text6**

Nombre	TxtPromedio
BackColor	&H80000004&
Locked	True
Text	

**Text7**

Nombre	TxtPuntuacion
BackColor	&H80000004&
Locked	True
Text	

**Command1**

Nombre	CmdAceptar
Caption	&Aceptar
Default	True

**Command2**

Nombre	CmdLimpiar
Caption	&Limpiar

**Command3**

Nombre	CmdSalir
Cancel	True
Caption	&Salir

Una vez establecidas las propiedades, proceda a ingresar el código que se indica:

```
Private Sub CmdAceptar_Click()
    Dim N1 As Double, N2 As Double, N3 As Double, N4 As Double
    Dim Promedio As Integer
    N1 = Val(TxtN1) : N2 = Val(TxtN2)
    N3 = Val(TxtN3) : N4 = Val(TxtN4)
    Promedio = CInt((N1 + N2 + N3 + N4) / 4)
    TxtPromedio = Str(Promedio)
    If Promedio >= 19 And Promedio <= 20 Then
        TxtPuntuacion = "A"

```

```

ElseIf Promedio >= 16 And Promedio <= 18 Then
    TxtPuntuacion = "B"
ElseIf Promedio >= 11 And Promedio <= 15 Then
    TxtPuntuacion = "C"
ElseIf Promedio >= 6 And Promedio <= 10 Then
    TxtPuntuacion = "D"
ElseIf Promedio >= 0 And Promedio <= 5 Then
    TxtPuntuacion = "B"
Else: MsgBox "Error de datos", vbCritical, "Mensaje"
End If
End Sub

```

### Aplicación N° 3

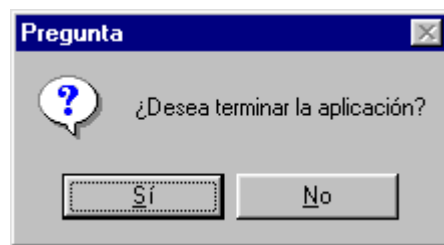
Escribir un aplicación que acepte fechas como tres números (dd, mm, aaaa) y las visualice del modo usual. A manera de ejemplo considere lo siguiente:



En caso de que el usuario ingrese algún dato incorrecto (por ejemplo 13 como un número de mes), se debe visualizar el siguiente mensaje:



Cuando el usuario haga click en el botón Salir, se debe observar el siguiente mensaje:



En caso de que el usuario elija la opción Si, la aplicación debe terminar. En caso contrario, es decir si el usuario elige la opción No, se debe proseguir con la aplicación.

Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 4 etiquetas
- 3 cajas de texto
- 1 marco
- 3 botones de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmFecha
BorderStyle	3-Fixed Dialog



Caption	Fecha en letras
---------	-----------------

**Label1**

Nombre	LblDia
AutoSize	True
Caption	Día:

**Label2**

Nombre	LblMes
AutoSize	True
Caption	Mes:

**Label3**

Nombre	LblAnno
AutoSize	True
Caption	Año:

**Label4**

Nombre	LblFecha
AutoSize	True
Caption	

**Text1**

Nombre	TxtDia
Text	

**Text2**

Nombre	TxtMes
Text	

**Text3**

Nombre	TxtAnno
--------	---------

Text	
------	--

### Command1

Nombre	CmdAceptar
Caption	&Aceptar
Default	True

### Command2

Nombre	CmdLimpiar
Caption	&Limpiar

### Command3

Nombre	CmdSalir
Caption	&Salir

Una vez establecidas las propiedades de la interfaz proceda a ingresar el siguiente código:

```
Private Sub CmdAceptar_Click()
    Dim CadMes As String, Mes As Integer
    Mes = Val(TxtMes)
    Select Case Mes
        Case 1: CadMes = "Enero"
        Case 2: CadMes = "Febrero"
        Case 3: CadMes = "Marzo"
        Case 4: CadMes = "Abril"
        Case 5: CadMes = "Mayo"
        Case 6: CadMes = "Junio"
        Case 7: CadMes = "Julio"
        Case 8: CadMes = "Agosto"
        Case 9: CadMes = "Setiembre"
        Case 10: CadMes = "Octubre"
        Case 11: CadMes = "Noviembre"
```

```

        Case 12: CadMes = "Diciembre"
        Case Else
            MsgBox "Error de datos...", vbCritical, "Mensaje"
            Call CmdLimpiar_Click
            Exit Sub
        End Select
        LblFecha = TxtDia & " de " & CadMes & " de " & TxtAnno
    End Sub

Private Sub CmdLimpiar_Click()
    TxtDia = "" : TxtMes = "" : TxtAnno = ""
    TxtDia.SetFocus
End Sub

Private Sub CmdSalir_Click()
    If MsgBox("¿Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Pregunta") = vbYes Then
        End
    Else: Call CmdLimpiar_Click
    End If
End Sub

```

#### **Aplicación N° 4**

Un restaurante ofrece un descuento del 10% para consumos entre S/. 30.00 y S/. 50.00; un descuento del 20% para consumos mayores a S/. 50.00 nuevos soles; para todos los demás casos no se aplica ningún tipo de descuento. Elaborar una aplicación que permita determinar el importe a pagar por el consumidor. El diseño de la interfaz y otras consideraciones se dejan a su criterio.

## **Microsoft Visual Basic**

## GUÍA DE LABORATORIO N° 3

### Objetivos

Luego de completar este laboratorio, el estudiante será capaz de:

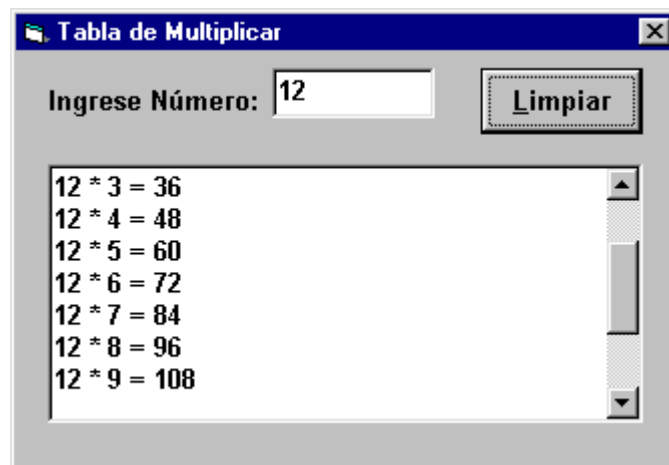
- Implementar aplicaciones utilizando estructuras de control repetitivas.
- Cambiar las propiedades de los controles en tiempo de ejecución.

### Consideraciones

Para el desarrollo del presente laboratorio Ud. deberá crear una carpeta **C:\FundVB\Lab3**, para guardar sus trabajos correspondientes a este laboratorio.

### Aplicación N° 1

Escribir una aplicación que lea un número entero y muestre la tabla de multiplicar de dicho número. El diseño de entrada y salida debe ser similar al siguiente:



Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 1 etiqueta
- 2 cajas de texto
- 1 botón de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmTabla
BorderStyle	3-Fixed Dialog
Caption	Tabla de multiplicar

**Label1**

Nombre	LblNumero
Caption	Ingrese un número:

**Text1**

Nombre	TxtNumero
Text	

**Text2**

Nombre	TxtTabla
MultiLine	True
Locked	True
ScrollBars	2-Vertical
Text	

**Command1**

Nombre	CmdLimpiar
Caption	&Limpiar

Una vez diseñada la interfaz, proceda a ingresar el código que se indica a continuación:

```

Private Sub TxtNumero_Change()
    If IsNumeric(TxtNumero) Then
        Dim N As Integer, P As Integer, I As Integer
        Dim S As String
        N = Val(TxtNumero)
        S = ""
        For I = 0 To 12
            P = N * I
            S = S & N & " * " & I & " = " & P & vbCrLf
        Next I
        TxtTabla = S
    ElseIf TxtNumero = "" Then
        Exit Sub
    Else: MsgBox "Ingrese un número", vbCritical, "Mensaje"
        TxtTabla = ""
    End If
End Sub

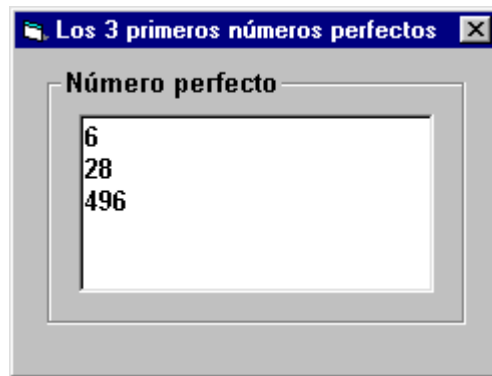
Private Sub CmdLimpiar_Click()
    TxtNumero = "" : TxtTabla = ""
    TxtNumero.SetFocus
End Sub

```

## **Aplicación N° 2**

Un número perfecto es un entero positivo, que es igual a la suma de todos los enteros positivos (excluido el mismo) que son divisores del número. El primer número perfecto es 6, ya que los divisores de 6 son 1, 2, 3 y  $1 + 2 + 3 = 6$ . Escribir una aplicación que encuentre los tres primeros números perfectos.

El diseño de la interfaz debe ser similar a la figura mostrada:



Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 1 marco
- 1 caja de texto

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmNumeroPerfecto
BorderStyle	3-Fixed Dialog
Caption	Los 3 primeros números perfectos

**Frame1**

Nombre	FraPerfecto
Caption	Número perfecto

**Text1**

Nombre	TxtPerfecto
MultiLine	True
Text	

Una vez establecidas las propiedades proceda a ingresar el código que se indica a continuación:

```

Private Sub Form_Load()
    Dim N As Long, I As Long, S As Long
    Dim K As Integer, Cad As String
    N = 1 : K = 0 : Cad = ""
    While True
        S = 0
        For I = 1 To (N - 1)
            If N Mod I = 0 Then S = S + I
        Next I
        If N = S Then
            Cad = Cad & N & vbCrLf
            K = K + 1
        End If
        If K = 3 Then
            TxtPerfecto = Cad
            Exit Sub
        End If
        N = N + 1
    Wend
End Sub

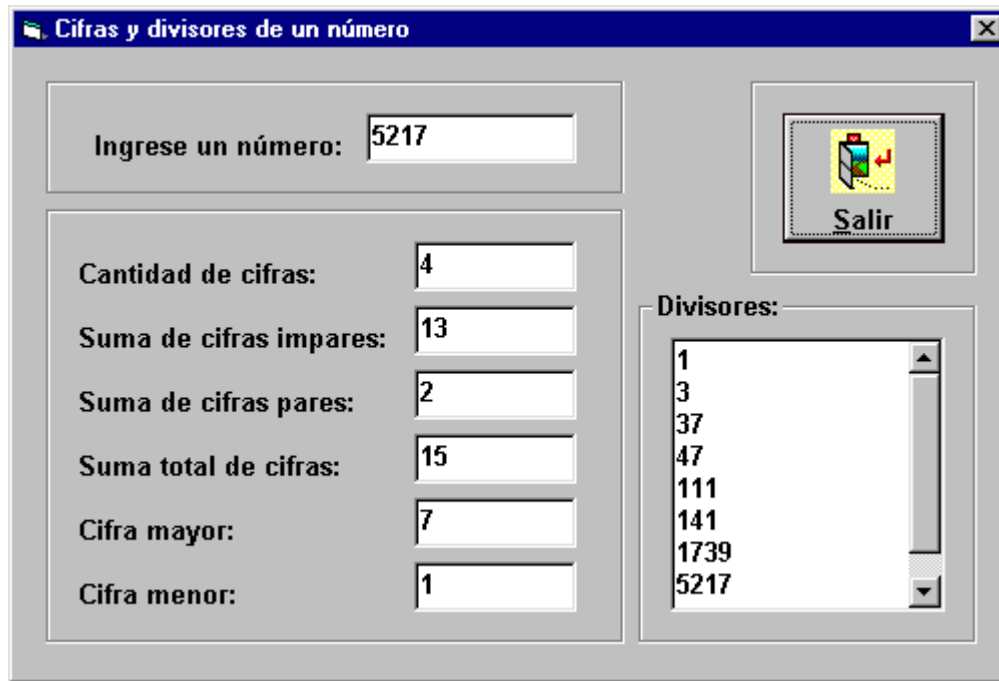
```

### **Aplicación N° 3**

Construya una aplicación que permita el ingreso de un número entero y muestre en pantalla la siguiente información: 1) Cantidad de cifras, 2) Suma de cifras impares, 3) Suma de cifras pares, 4) Suma total de cifras, 5) Cifra mayor, 6) Cifra menor y 7) Divisores de dicho número.

El diseño de la interfaz debe ser similar a la figura siguiente:





Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 4 marcos
- 7 etiquetas
- 8 cajas de texto
- 1 botón de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmNumeroPerfecto
BorderStyle	3-Fixed Dialog
Caption	Los 3 primeros números perfectos
Moveable	False
StartPosition	2-CenterScreen

**Frame1**

Nombre	FraEntrada
Caption	

**Frame2**

Nombre	FraSalida
Caption	

**Frame3**

Nombre	FraDivisores
Caption	Divisores

**Frame4**

Nombre	FraSalir
Caption	

**Label1**

Nombre	LblNumero
AutoSize	True
Caption	Ingrese un número:

**Label2**

Nombre	LblCantCifras
AutoSize	True
Caption	Cantidad de cifras:

**Label3**

Nombre	LblSumImpares
AutoSize	True
Caption	Suma de cifras impares:

**Label14**

Nombre	LblSumPares
AutoSize	True
Caption	Suma de cifras pares:

**Label15**

Nombre	LblSumTotal
AutoSize	True
Caption	Suma total de cifras:

**Label16**

Nombre	LblCifraMayor
AutoSize	True
Caption	Cifra mayor:

**Label17**

Nombre	LblCifraMenor
AutoSize	True
Caption	Cifra menor:

**Text1**

Nombre	TxtNumero
Text	

**Text2**

Nombre	TxtCantCifras
Locked	True
Text	

**Text3**

Nombre	TxtSumImpares
Locked	True

Text	
------	--

**Text4**

Nombre	TxtSumPares
Locked	True
Text	

**Text5**

Nombre	TxtSumTotal
Locked	True
Text	

**Text6**

Nombre	TxtCifraMayor
Locked	True
Text	

**Text7**

Nombre	TxtCifraMenor
Locked	True
Text	

**Text8**

Nombre	TxtDivisores
MultiLine	True
Locked	True
ScrollBars	2-Vertical
Text	

**Command3**

Nombre	CmdSalir
Caption	&Salir
Picture	C:\FundVB\Bitmaps\Exit.bmp

Style	1-Graphical
-------	-------------

Una vez establecidas las propiedades proceda a ingresar el código que se indica a continuación:

```
Private Sub CmdAceptar_Click()  
    If IsNumeric(TxtNumero) Then  
        Dim S As Integer, SI As Integer, SP As Integer  
        Dim May As Integer, Min As Integer  
        Dim Cad As String  
        Dim I As Integer, J As Integer  
        N = CLng(TxtNumero)  
        M = CLng(TxtNumero)  
        Cad = ""  
        I = 0  
        J = 1  
        S = SP = SI = 0  
        For J = 1 To N  
            If (N Mod J = 0) Then  
                Cad = Cad & J & vbCrLf  
            End If  
        Next J  
        While (N > 0)  
            If ((N Mod 10) Mod 2) = 0 Then  
                SP = SP + (N Mod 10)  
            Else  
                SI = SI + (N Mod 10)  
            End If  
            S = S + (N Mod 10)  
            N = N \ 10  
            I = I + 1  
        Wend
```

```

    May = Mid(TxtNumero, 1, 1)
    Men = May
    While (M > 0)
        If May < (M Mod 10) Then
            May = M Mod 10
        End If
        If Men > (M Mod 10) Then
            Men = M Mod 10
        End If
        M = M \ 10
    Wend
    TxtCantCifras = Str(I)
    TxtSumImpares = Str(SI)
    TxtSumPares = Str(SP)
    TxtSumTotal = Str(S)
    TxtCifraMayor = Str(May)
    TxtCifraMenor = Str(Men)
    TxtDivisores = Cad
Else
    MsgBox "Debe ingresar un número", vbCritical, "Mensaje"
    TxtNumero.SetFocus
End If
End Sub

Private Sub CmdSalir_Click()
    If MsgBox("¿Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Pregunta") = vbYes Then
        End
    Else
        Cancel = True
        TxtNumero.SetFocus
    End If
End Sub

```

End Sub

#### **Aplicación N° 4**

Un centro numérico es un número que separa una lista de números enteros (comenzando en 1) en dos grupos de números, cuyas sumas son iguales. El primer centro numérico es el 6, el cual separa la lista (1 a 8) en los grupos: (1; 2; 3; 4; 5) y (7; 8) cuyas sumas son ambas iguales a 15. El segundo centro numérico es el 35, el cual separa la lista (1 a 49) en los grupos: (1 a 34) y (36 a 49) cuyas sumas son ambas iguales a 595. Se pide elaborar una aplicación que calcule los centros numéricos entre 1 y N.

El diseño de la interfaz y otras consideraciones se dejan a su criterio.

# Microsoft **Visual Basic**

## GUÍA DE LABORATORIO N° 4

### Objetivos

Luego de completar este laboratorio, el estudiante será capaz de:

- Manipular y realizar operaciones con arreglos.
- Crear una estructura y realizar operaciones con los campos de la misma.
- Implementar arreglos de estructuras.
- Trabajar con listas desplegables y cuadros combinados.

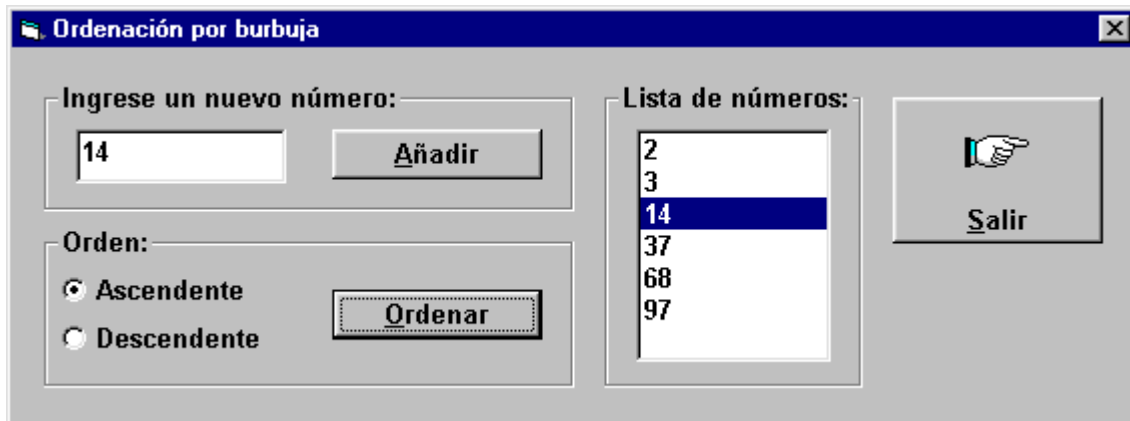
### Consideraciones

Para el desarrollo del presente laboratorio Ud. deberá crear una carpeta **C:\FundVB\Lab4** para guardar sus trabajos correspondientes a este laboratorio.

### Aplicación N° 1

Elabore una aplicación que permita leer N números de tipo entero, y a continuación los visualice ordenados en forma ascendente o descendente.





Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 3 marcos
- 1 caja de texto
- 1 control lista
- 2 botones de opción
- 3 botones de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmBurbuja
BorderStyle	3-Fixed Dialog
Caption	Ordenación por burbuja
Moveable	False

**Frame1**

Nombre	FraNumero
Caption	Ingrese un nuevo número:

**Frame2**

Nombre	FraLista
Caption	Lista de números:

**Frame3**

Nombre	FraOrden
Caption	Orden:

**Text1**

Nombre	TxtNumero
Text	

**List1**

Nombre	LstNumero
List	

**Option1**

Nombre	OptAscendente
Caption	Ascendente
Value	True

**Option2**

Nombre	OptDescendente
Caption	Descendente
Value	False

**Command1**

Nombre	CmdAnnadir
Caption	&Añadir
Default	True

**Command2**

Nombre	CmdOrdenar
Caption	&Ordenar

**Command3**

Nombre	CmdSalir
Caption	&Salir
Picture	C:\Archivos de programa\Microsoft Visual Studio\Common\Graphics\Icons\Arrows\Point04.ico
Style	1-Graphical

Una vez establecidas las propiedades proceda a ingresar el código que se indica a continuación:

```
Private Sub CmdAceptar_Click()
    If IsNumeric(TxtNumero.Text) Then
        LstNumero.AddItem TxtNumero.Text
        TxtNumero.Text = ""
        TxtNumero.SetFocus
    Else
        MsgBox "Ingrese un número", vbCritical, "Mensaje"
        TxtNumero.SelStart = 0
        TxtNumero.SelLength = Len(TxtNumero.Text)
        TxtNumero.SetFocus
    End If
End Sub

Private Sub CmdOrdenar_Click()
    Dim I As Integer, J As Integer, T As Integer, N As Integer
    Dim A() As Integer
    N = LstNumero.ListCount
    ReDim A(N)
    For I = 0 To N - 1
        A(I) = LstNumero.List(I)
    Next I
    If OptAscendente.Value Then
        For I = 0 To N - 2
```

```

        For J = I + 1 To N - 1
            If A(I) > A(J) Then
                T = A(I)
                A(I) = A(J)
                A(J) = T
            End If
        Next J
    Next I
End If

If OptDescendente.Value Then
    For I = 0 To N - 2
        For J = I + 1 To N - 1
            If A(I) < A(J) Then
                T = A(I)
                A(I) = A(J)
                A(J) = T
            End If
        Next J
    Next I
End If

LstNumero.Clear
For I = 0 To N - 1
    LstNumero.List(I) = A(I)
Next I
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If MsgBox("Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Pregunta") = vbYes Then
        End
    Else: Cancel = True : TxtNumero.SetFocus
    End If
End Sub

```

End Sub

```
Private Sub CmdSalir_Click()
```

```
    Unload Me
```

```
End Sub
```

## Aplicación N° 2

Elaborar una aplicación que permita seleccionar un artículo de un cuadro combinado (Combo). Apenas el usuario seleccione un artículo se debe mostrar el precio del mismo, el interés es fijo para esta ocasión.

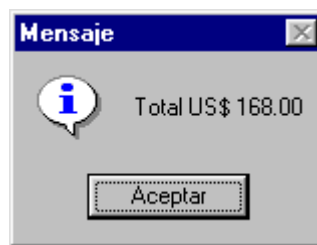
El diseño de la interfaz debe ser similar a la siguiente figura:



La venta ha realizarse es a plazos, ello condiciona la cuota mensual a pagarse. Cuando se haga click sobre el botón Cuota mensual debe mostrarse un cuadro de diálogo con los datos propuestos:



De manera similar al hacer click sobre el botón Total nos debe mostrar la cantidad total a pagar.



Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 1 marco
- 3 etiquetas
- 1 cuadro combinado
- 2 cajas de texto
- 3 botones de opción
- 2 botones de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmCotizacion
BorderStyle	3-Fixed Dialog
Caption	Pedido de cotizaciones

**Frame1**

Nombre	FraPlazo
--------	----------

Caption	Plazo:
---------	--------

**Label1**

Nombre	LblArticulo
Caption	Artículo:

**Label2**

Nombre	LblPrecio
Caption	Precio US\$

**Label3**

Nombre	LblInteres
Caption	Interés:

**Combo1**

Nombre	CboArticulo
Text	

**Text1**

Nombre	TxtPrecio
Locked	True
Text	

**Text2**

Nombre	TxtInteres
Locked	True
Text	

**Option1**

Nombre	OptPlazo
Caption	6 meses
Value	True

### Option2

Nombre	OptPlazo
Caption	12 meses
Value	False

### Option3

Nombre	OptPlazo
Caption	24 meses
Value	False

### Command1

Nombre	CmdCuotaMensual
Caption	&Cuota mensual

### Command2

Nombre	CmdTotal
Caption	&Total

Una vez establecidas las propiedades de la interfaz, haga doble click sobre el formulario e ingrese las siguientes declaraciones en la sección General del módulo de formulario:

```
Private Type Articulo
    Nombre As String * 30
    Precio As Double
End Type
Dim A(4) As Articulo, Plazo As Integer
Const Interes = 0.12
```

Recuerde que un dato declarado en la sección General de un módulo puede ser accedido por todos los procedimientos de dicho módulo. Luego, continúe ingresando el código que se muestra a continuación:



```

Private Sub Form_Load()
    A(0).Nombre = "Monitor SAMSUNG SyncMaster 3"
    A(1).Nombre = "Impresora Hewlett Packard DeskJet 930C"
    A(2).Nombre = "Impresora Epson Stylus Color 740"
    A(3).Nombre = "Microprocesador Pentium I 233 MHZ"
    A(0).Precio = 150 : A(1).Precio = 275
    A(2).Precio = 145 : A(3).Precio = 80
    Dim I As Integer
    For I = 1 To 4
        CboArticulo.AddItem A(I - 1).Nombre
    Next I
    TxtInteres = Interes : Plazo = 6
End Sub

Private Sub CboArticulo_Click()
    Dim I As Integer
    I = CboArticulo.ListIndex
    TxtPrecio = A(I).Precio
End Sub

Private Sub OptPlazo_Click(Index As Integer)
    Select Case Index
        Case 0: Plazo = 6
        Case 1: Plazo = 12
        Case 2: Plazo = 24
    End Select
End Sub

Private Sub CmdCuotaMensual_Click()
    Dim Total As Double, CuotaMensual As Double, I As Integer
    I = CboArticulo.ListIndex
    Total = A(I).Precio * (1 + Interes)
    CuotaMensual = Total / Plazo

```

```
        MsgBox "Cuota Mensual US$" & Str(CuotaMensual)
End Sub

Private Sub CmdTotal_Click()
    Dim Total As Double, I As Integer
    I = CboArticulo.ListIndex
    Total = A(I).Precio * (1 + Interes)
    MsgBox "Total US$" & Str(Total)
End Sub
```

### **Aplicación N° 3**

Se desea elaborar una aplicación que permita controlar el proceso de matrícula en un curso de computación. Para ello se deben recabar los siguientes datos: 1) Curso en que se matricula el alumno, 2) Fecha de matrícula, 3) Apellidos y nombres, 4) Sexo, 5) Dirección, y 6) Distrito de residencia.

Para el desarrollo de esta aplicación proceda a ubicar los siguientes controles en el formulario:

- 4 marcos
- 6 etiquetas
- 5 cajas de texto
- 1 cuadro combinado
- 2 controles de lista
- 4 botones de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmFichaMatricula
--------	-------------------

BorderStyle	3-Fixed Dialog
Caption	Ficha de matrícula

**Frame1**

Nombre	FraIngreso
Caption	Ingreso de datos:

**Frame2**

Nombre	FraOpciones
Caption	Opciones:

**Frame3**

Nombre	FraCursos
Caption	Cursos:

**Frame4**

Nombre	FraDistritos
Caption	Distritos:

**Label1**

Nombre	LblCurso
Caption	Curso:

**Label2**

Nombre	LblFechaMat
Caption	Fecha de matrícula:

**Label3**

Nombre	LblAlumno
Caption	Alumno:

**Label4**

Nombre	LblSexo
--------	---------

Caption	Sexo:
---------	-------

**Label15**

Nombre	LblDirección
Caption	Dirección:

**Label16**

Nombre	LblDistrito
Caption	Distrito:

**Combo1**

Nombre	CboSexo
Text	

**List1**

Nombre	LstCursos
Text	

**List2**

Nombre	LstDistrito
Text	

**Text1**

Nombre	TxtCurso
Text	

**Text2**

Nombre	TxtFechaMat
Text	

**Text3**

Nombre	TxtAlumno
Text	

**Text4**

Nombre	TxtDireccion
Text	

**Text5**

Nombre	TxtDistrito
Text	

**Command1**

Nombre	CmdGuardar
Caption	&Guardar

**Command2**

Nombre	CmdCancelar
Caption	&Cancelar

**Command3**

Nombre	CmdLimpiar
Caption	&Limpiar

**Command4**

Nombre	CmdSalir
Caption	&Salir
Picture	C:\FundVB\Bitmaps\Exit.bmp
Style	1-Graphical

Una vez establecidas las propiedades de la interfaz, proceda a ingresar el código que se indica a continuación:

```
Private Sub Form_Load()
```

```
LstCursos.AddItem "Borland C++ Nivel I"
LstCursos.AddItem "Borland C++ Nivel II"
LstCursos.AddItem "Microsoft Visual Basic Nivel I"
LstCursos.AddItem "Microsoft Visual Basic Nivel II"
LstCursos.AddItem "Microsoft Visual FoxPro Nivel I"
LstCursos.AddItem "Microsoft Visual FoxPro Nivel II"
LstCursos.AddItem "Microsoft Visual C++ Nivel I"
LstCursos.AddItem "Microsoft Visual C++ Nivel II"
LstCursos.AddItem "Microsoft Visual J++ Nivel I"
LstCursos.AddItem "Microsoft Visual J++ Nivel II"
LstCursos.AddItem "Microsoft SQL Server Nivel I"
LstCursos.AddItem "Microsoft SQL Server Nivel II"
LstCursos.AddItem "Microsoft Power Builder Nivel I"
LstCursos.AddItem "Microsoft Power Builder Nivel II"
LstDistrito.AddItem "Callao"
LstDistrito.AddItem "Bellavista"
LstDistrito.AddItem "Carmen de la Legua"
LstDistrito.AddItem "La Perla"
LstDistrito.AddItem "La Punta"
LstDistrito.AddItem "Ventanilla"
LstDistrito.AddItem "Cercado de Lima"
LstDistrito.AddItem "Ancón"
LstDistrito.AddItem "Ate"
LstDistrito.AddItem "Barranco"
LstDistrito.AddItem "Breña"
LstDistrito.AddItem "Carabayllo"
LstDistrito.AddItem "Comas"
LstDistrito.AddItem "Chaclacayo"
LstDistrito.AddItem "Chorrillos"
LstDistrito.AddItem "El Agustino"
LstDistrito.AddItem "Jesús María"
LstDistrito.AddItem "La Molina"
```

```
LstDistrito.AddItem "La Victoria"  
LstDistrito.AddItem "Lince"  
LstDistrito.AddItem "Lurigancho"  
LstDistrito.AddItem "Lurín"  
LstDistrito.AddItem "Magdalena del Mar"  
LstDistrito.AddItem "Miraflores"  
LstDistrito.AddItem "Pachacamac"  
LstDistrito.AddItem "Pucusana"  
LstDistrito.AddItem "Pueblo Libre"  
LstDistrito.AddItem "Puente Piedra"  
LstDistrito.AddItem "Punta Negra"  
LstDistrito.AddItem "Rimac"  
LstDistrito.AddItem "San Bartolo"  
LstDistrito.AddItem "San Isidro"  
LstDistrito.AddItem "Independencia"  
LstDistrito.AddItem "San Juan de Miraflores"  
LstDistrito.AddItem "San Luis"  
LstDistrito.AddItem "San Martín de Porres"  
LstDistrito.AddItem "San Miguel"  
LstDistrito.AddItem "Santiago de Surco"  
LstDistrito.AddItem "Villa María del Triunfo"  
LstDistrito.AddItem "San Juan de Lurigancho"  
LstDistrito.AddItem "Santa María del Mar"  
LstDistrito.AddItem "Santa Rosa"  
LstDistrito.AddItem "Los Olivos"  
LstDistrito.AddItem "Cieneguilla"  
LstDistrito.AddItem "San Borja"  
LstDistrito.AddItem "Villa el Salvador"  
LstDistrito.AddItem "Santa Anita"  
CboSexo.AddItem "Masculino" : CboSexo.AddItem "Femenino"
```

End Sub

Private Sub Form\_Unload(Cancel As Integer)



```

    If MsgBox("¿Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Pregunta") = vbYes Then
        End
    Else
        Cancel = True
        Call CmdLimpiar_Click
    End If
End Sub

Private Sub LstCursos_Click()
    TxtCursos = LstCursos
End Sub

Private Sub LstDistrito_Click()
    TxtDistrito = LstDistrito
End Sub

Private Sub CmdGuardar_Click()
    LstCursos.Enabled = False
    LstDistrito.Enabled = False
    TxtCursos.Locked = True
    TxtFechaMat.Locked = True
    TxtAlumno.Locked = True
    CboSexo.Locked = True
    TxtDireccion.Locked = True
    TxtDistrito.Locked = True
    MsgBox "Alumno matriculado", vbInformation, "Mensaje"
End Sub

Private Sub CmdLimpiar_Click()
    LstCursos.Enabled = True
    LstDistrito.Enabled = True
    TxtCursos.Locked = False
    TxtFechaMat.Locked = False

```

```

    TxtAlumno.Locked = False
    CboSexo.Locked = False
    TxtDireccion.Locked = False
    TxtDistrito.Locked = False
    TxtCursos = ""
    TxtFechaMat = ""
    TxtAlumno = ""
    CboSexo = ""
    TxtDireccion = ""
    TxtDistrito = ""
    TxtCursos.SetFocus
End Sub

Private Sub CmdCancelar_Click()
    If MsgBox("¿Desea modificar algún dato?", _
        vbQuestion + vbYesNo, "Mensaje") = vbYes Then
        LstCursos.Enabled = True
        LstDistrito.Enabled = True
        TxtCursos.Locked = False
        TxtFechaMat.Locked = False
        TxtAlumno.Locked = False
        CboSexo.Locked = False
        TxtDireccion.Locked = False
        TxtDistrito.Locked = False
    End If
End Sub

Private Sub CmdSalir_Click()
    Unload Me
End Sub

```

#### **Aplicación N° 4**

Desarrollar una aplicación que permita realizar consultas acerca de un determinado curso, los cuales se mostraran en una lista. El usuario debe seleccionar un curso y en seguida se debe presentar el nombre del profesor encargado del curso (teoría), el nombre del jefe de práctica (laboratorio), así como los horarios de teoría y de laboratorio. El diseño de la interfaz deberá ser similar al siguiente:

Consulta de cursos

Profesor de teoría: Castillo Peralta, Carlos

Horario de teoría: Sa 10-13

Jefe de práctica: Córdoba Saavedra, Javier

Horario de laboratorio: Do 08-10

Cursos:

- Microsoft Visual Basic Nivel II
- Microsoft Visual FoxPro Nivel I
- Microsoft Visual FoxPro Nivel II
- Microsoft Visual C++ Nivel I
- Microsoft Visual C++ Nivel II
- Microsoft Visual J++ Nivel I
- Microsoft Visual J++ Nivel II
- Microsoft SQL Server Nivel I
- Microsoft SQL Server Nivel II

# Microsoft **Visual Basic**

## GUÍA DE LABORATORIO N° 5

### Objetivos

Luego de completar este laboratorio, el estudiante será capaz de:

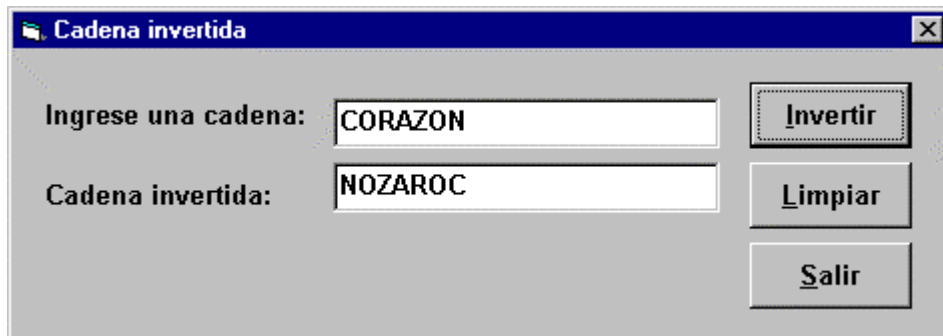
- Crear procedimientos y funciones definidos por el usuario.
- Añadir formularios creados anteriormente a un proyecto.
- Cambiar el formulario de arranque de un proyecto.
- Utilizar formularios MDI.
- Manejar diferentes métodos y propiedades de los formularios.
- Crear y utilizar menús en una aplicación.

### Consideraciones

Para el desarrollo del presente laboratorio Ud. deberá crear una carpeta **C:\FundVB\Lab5**, para guardar sus trabajos correspondientes a este laboratorio.

### Aplicación N° 1

Escriba una función que reciba como argumento una cadena de caracteres y la devuelva en forma inversa, por ejemplo si se ingresa la cadena CORAZON deberá retornar NOZAROC.



Para el desarrollo de esta aplicación, proceda a ubicar los siguientes controles en el formulario:

- 2 etiquetas
- 2 cajas de texto
- 3 botones de comando

En seguida proceda a establecer las propiedades según se indica:

**Form1**

Nombre	FrmCadInvertida
Caption	Cadena invertida
BorderStyle	3-Fixed Dialog

**Label1**

Nombre	LblCadena
AutoSize	True
Caption	Ingresa una cadena:

**Label2**

Nombre	LblInvertida
AutoSize	True
Caption	Cadena invertida:

**Text1**

Nombre	TxtCadena
Text	

**Text2**

Nombre	TxtInvertida
Locked	True
Text	

**Command1**

Nombre	CmdAceptar
Caption	&Aceptar

**Command2**

Nombre	CmdLimpiar
Caption	&Limpiar

**Command3**

Nombre	CmdSalir
Caption	&Salir

Una vez establecidas las propiedades de la interfaz, proceda a ingresar el siguiente código:

```
Function CadInvertida(Cadena As String) As String
    Dim Invertida() As String * 1
    Dim I As Integer
    Dim J As Integer
    N = Len(Cadena)
    ReDim Invertida(N)
    For I = 1 To N
        Invertida(I - 1) = Mid(Cadena, I, 1)
    Next I
    For J = (N - 1) To 0 Step -1
        CadInvertida = CadInvertida & Invertida(J)
    Next J
End Function

Private Sub CmdInvertir_Click()
    TxtInvertida = CadInvertida(TxtCadena)
End Sub
```

```

Private Sub CmdLimpiar_Click()
    TxtCadena = ""
    TxtInvertida = ""
    TxtCadena.SetFocus
End Sub

```

```

Private Sub CmdSalir_Click()
    End
End Sub

```

## Aplicación N° 2

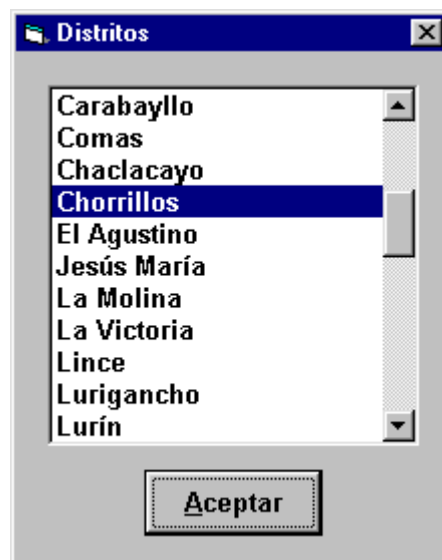
Se desea elaborar una aplicación que permita controlar el proceso de matrícula en un curso de computación. Para ello se deben recabar los siguientes datos: 1) Curso en que se matricula el alumno, 2) Fecha de matrícula, 3) Apellidos y nombres, 4) Sexo, 5) Dirección, y 6) Distrito de residencia. El diseño de la interfaz debe ser similar a la figura mostrada:

Para seleccionar un curso el usuario deberá hacer click en el botón punteado que se encuentra al lado de la caja de texto. En seguida se presentará un menú de selección por realce en el cual se presenta la relación de todos los cursos disponibles.

El curso quedará seleccionado al hacer click en el botón Aceptar.



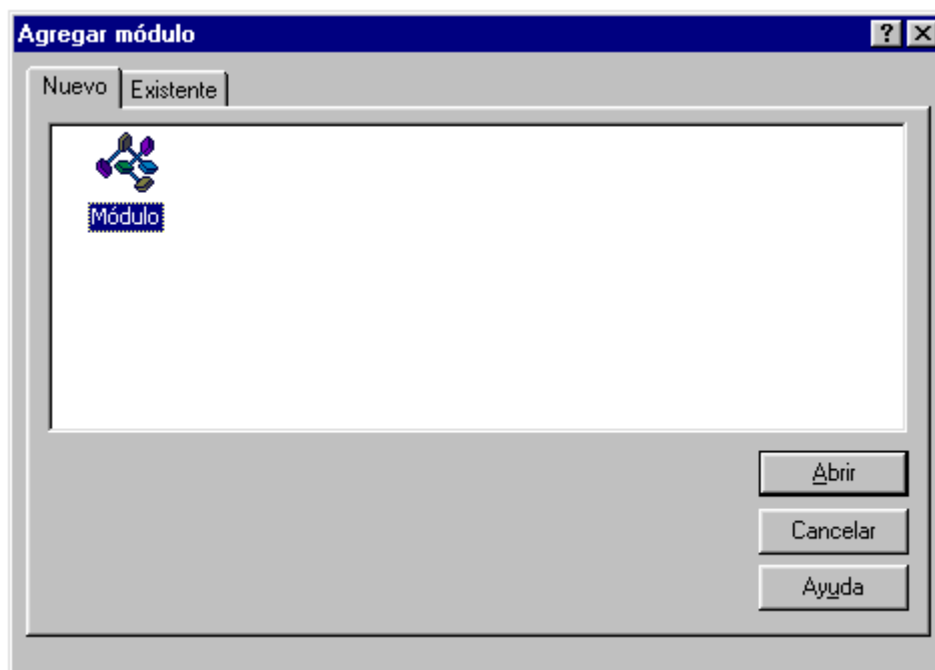
El mismo tipo de selección deberá realizarse al momento de ingresar el nombre del distrito.



Para el desarrollo de esta aplicación necesitamos tres formularios y un módulo. En primer lugar proceda a añadir un módulo de código al proyecto. Seleccione el Menú Proyecto y



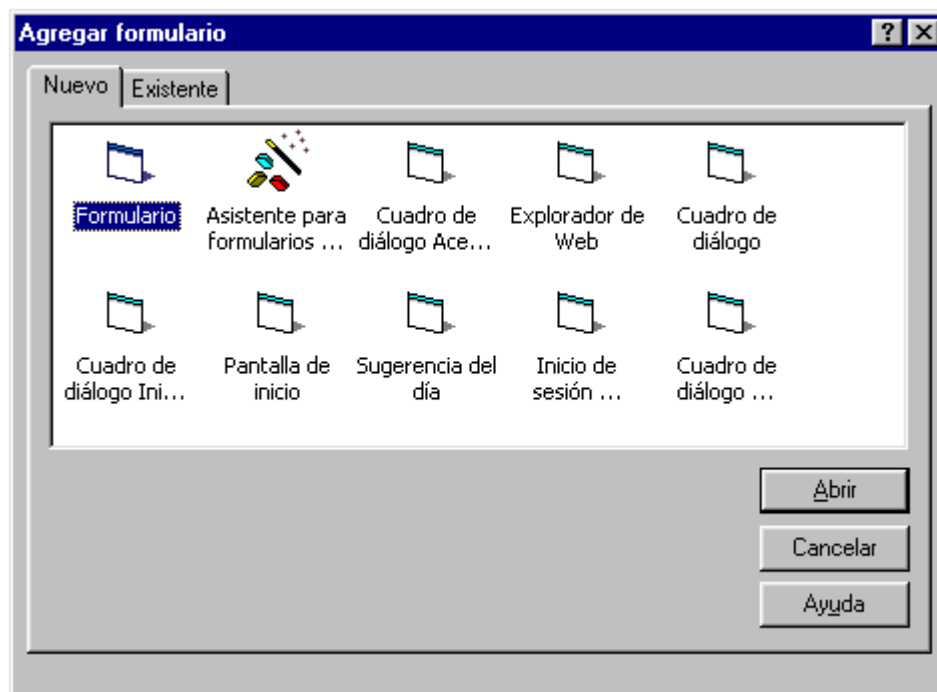
elija la opción Agregar módulo, se debe presentar un cuadro de diálogo similar a la siguiente figura:



Del cuadro de diálogo Agregar módulo, en la ficha Nuevo, haga click en el botón Abrir. Luego ingrese el siguiente código en la sección de Declaraciones del módulo que acabamos de añadir:

```
Public Curso As String  
Public Distrito As String
```

En seguida proceda a cambiar el nombre formulario principal por FrmFichaMatricula. Luego debe añadir los demás formularios necesarios para construir la aplicación. Para tal fin, seleccione el menú Proyecto y elija la opción Agregar formulario. Del cuadro de diálogo Agregar formulario, en la ficha Nuevo, elija la opción Formulario y haga click en el botón Abrir. Se debe presentar un cuadro de diálogo similar a la siguiente figura:



En ese instante se añadirá un nuevo formulario al proyecto. Cambie el nombre del nuevo formulario por FrmCurso. Repita el procedimiento anterior para añadir el formulario FrmDistrito.

A continuación copie los pasos de la pág. 55 a la pág. 62 de la Guía de Laboratorio N° 4 (Aplicación N° 3) con los siguientes cambios:

En la pág. 55, añadir sólo 2 marcos. No añadir ningún control de lista. En vez de 4 botones de comando, añadir 6 botones de comando. Luego, establecer las siguientes propiedades para los dos nuevos botones:

#### Command5

Nombre	CmdCurso
Caption	. . .

## Command6

Nombre	CmdDistrito
Caption	. . .

En la pág. 59, reemplazar el código del evento Load del formulario por el siguiente:

```
Private Sub Form_Load()  
    CboSexo.AddItem "Masculino"  
    CboSexo.AddItem "Femenino"  
End Sub
```

En la pág. 61, suprimir el código asociado al evento Click de los controles de lista LstCursos y LstDistrito. En su lugar añadir lo siguiente:

```
Private Sub CmdCurso_Click()  
    Load FrmCurso  
    FrmCurso.Show vbModal  
    TxtCurso = Curso  
End Sub  
  
Private Sub CmdDistrito_Click()  
    Load FrmDistrito  
    FrmDistrito.Show vbModal  
    TxtDistrito = Distrito  
End Sub
```

A continuación active el formulario FrmCurso, para ello haga click sobre el mismo. En seguida proceda a ubicar los siguientes controles en el formulario:

- 1 control de lista
- 1 botones de comando

Luego proceda a establecer las propiedades según se indica:

**Form2**

Nombre	FrmCurso
BorderStyle	3-Fixed Dialog
Caption	Cursos

**List1**

Nombre	LstCursos
Text	

**Command1**

Nombre	CmdAceptar
Caption	&Aceptar

Una vez establecidas las propiedades proceda a ingresar el código que se indica a continuación:

```
Private Sub Form_Load()  
    LstCursos.AddItem "Borland C++ Nivel I"  
    LstCursos.AddItem "Borland C++ Nivel II"  
    LstCursos.AddItem "Microsoft Visual Basic Nivel I"  
    LstCursos.AddItem "Microsoft Visual Basic Nivel II"  
    LstCursos.AddItem "Microsoft Visual FoxPro Nivel I"  
    LstCursos.AddItem "Microsoft Visual FoxPro Nivel II"  
    LstCursos.AddItem "Microsoft Visual C++ Nivel I"  
    LstCursos.AddItem "Microsoft Visual C++ Nivel II"  
    LstCursos.AddItem "Microsoft SQL Server Nivel I"  
    LstCursos.AddItem "Microsoft SQL Server Nivel II"  
    LstCursos.AddItem "Microsoft Power Builder Nivel I"  
    LstCursos.AddItem "Microsoft Power Builder Nivel II"  
End Sub
```

```

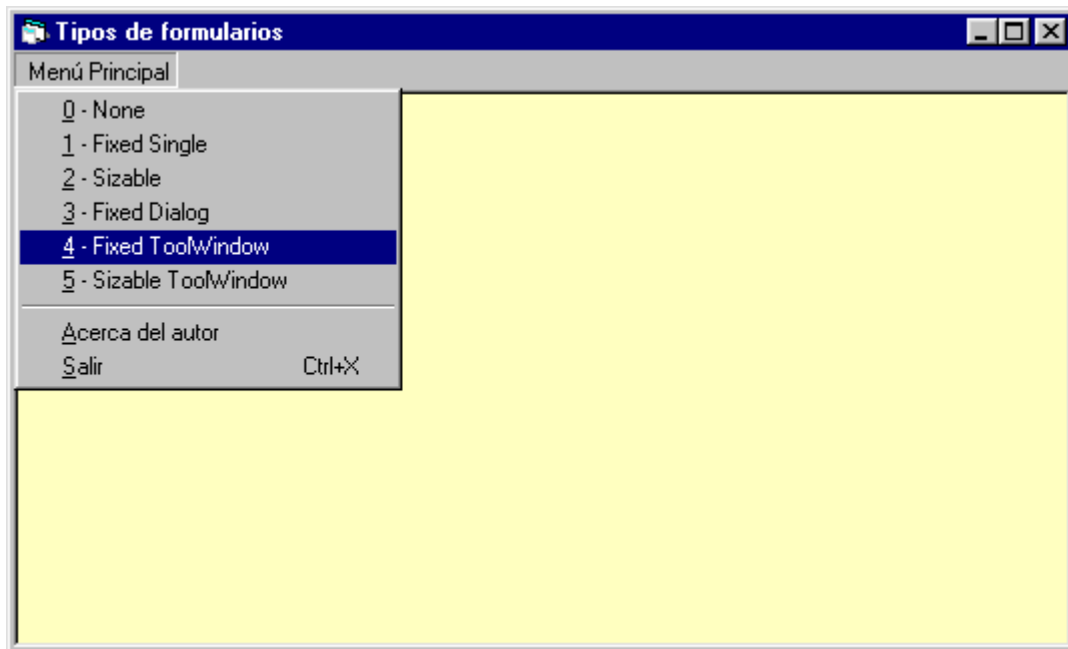
Private Sub CmdAceptar_Click()
    Curso = LstCursos.Text
    FrmCurso.Hide
End Sub

```

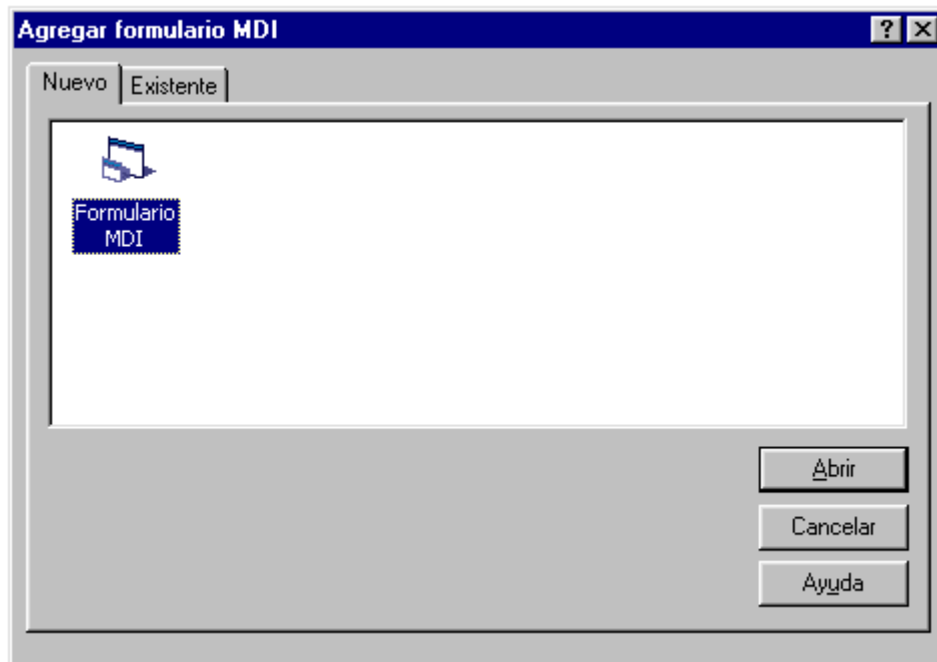
Por último, proceda Ud. a desarrollar el código respectivo para el formulario FrmDistrito.

### Aplicación N° 3

Elaborar una aplicación que permita presentar los diferentes tipos de formularios de Visual Basic. Utilice como contenedor principal un formulario MDI, tal como se muestra en la figura:



Para el desarrollo de esta aplicación necesitamos utilizar un formulario MDI (interfaz de múltiples documentos). Para ello seleccione el Menú Proyecto y elija la opción Agregar formulario MDI, se debe presentar un cuadro de diálogo similar a la siguiente figura:



Del cuadro de diálogo Agregar formulario MDI, en la ficha Nuevo, haga click en el botón Abrir. En seguida cambie el nombre del formulario MDI por MDIPrincipal.

A continuación proceda a añadir los formularios para las diferentes opciones del menú. Cambie los nombres de los formularios según se indica:

Formulario	Nombre
Form1	FrmNone
Form2	FrmFixedSingle
Form3	FrmSizable
Form4	FrmFixedDialog
Form5	FrmFixedToolWindow
Form6	FrmSizableToolWindow
Form7	FrmAcercaDe

Luego proceda a diseñar el menú de opciones. Para ello haga click derecho sobre el formulario MDI y elija la opción Editor de menús. Establezca las propiedades según:

<b>Caption</b>	<b>Name</b>	<b>ShortCut</b>
&Menú Principal	MnuPrincipal	Ninguno
&0-None	MnuNone	Ninguno
&1-Fixed Single	MnuFixedSingle	Ninguno
&2-Sizable	MnuSizable	Ninguno
&3-Fixed Dialog	MnuFixedDialog	Ninguno
&4-Fixed ToolWindow	MnuFixedToolWindow	Ninguno
&5-Sizable ToolWindow	MnuSizableToolWindow	Ninguno
-	MnuLinea	Ninguno
&Acerca del autor	MnuAcercaDe	Ninguno
&Salir	MnuSalir	Ctrl + X

A continuación haga click en el botón Aceptar del Editor de menús. Luego proceda a ingresar el siguiente código para el formulario MDI:

```
Private Sub MDIForm_Unload(Cancel As Integer)
    If MsgBox("¿Desea terminar la aplicación?", _
        vbQuestion + vbYesNo, "Mensaje") = vbYes Then
        End
    Else: Cancel = True
    End If
End Sub

Private Sub MnuNone_Click()
    Load FrmNone
    FrmNone.Show
End Sub
```

```
Private Sub MnuFixedSingle_Click()  
    Load FrmFixedSingle  
    FrmFixedSingle.Show  
End Sub
```

```
Private Sub MnuSizable_Click()  
    Load FrmSizable  
    FrmSizable.Show  
End Sub
```

```
Private Sub MnuFixedDialog_Click()  
    Load FrmFixedDialog  
    FrmFixedDialog.Show  
End Sub
```

```
Private Sub MnuFixedToolWindow_Click()  
    Load FrmFixedToolWindow  
    FrmFixedToolWindow.Show  
End Sub
```

```
Private Sub MnuSizableToolWindow_Click()  
    Load FrmSizableToolWindow  
    FrmSizableToolWindow.Show  
End Sub
```

```
Private Sub MnuAcercaDe_Click()  
    Load FrmAcercaDe  
    FrmAcercaDe.Show  
End Sub
```

```
Private Sub MnuSalir_Click()  
    Unload Me  
End Sub
```



A continuación proceda a activar el formulario FrmNone y ubique un botón de comandos sobre el mismo. En seguida establezca las propiedades según se indica a continuación:

**Form1**

Nombre	FrmNone
BorderStyle	0-None
Caption	None
MDIChild	True

**Command1**

Nombre	CmdVolver
Caption	&Volver

Una vez establecidas las propiedades, proceda a ingresar el código que se muestra a continuación:

```
Private Sub CmdVolver_Click()
    Unload Me
End Sub
```

Repita el procedimiento anterior para los demás tipos de formularios.

**Aplicación N° 4**

Elaborar una aplicación que acepte como entrada la reserva de agua de un depósito y los litros que se consumen a la semana. Utilizando una función definida por el usuario determinar como resultado las cantidades de agua que quedan al final de cada semana. El proceso finalizará cuando no quede agua suficiente para una semana. Utilizar otro formulario para mostrar la salida.

**FIN PARTE I**