



PRÁCTICA CINCO DE VISUAL BASIC

OPERADORES

Un operador es un signo cadena de caracteres o combinación de éstos que realiza cálculos o comparaciones sobre diferentes tipos de expresiones (variables constantes o valores). Visual provee varios operadores para realizar cálculos matemáticos comparaciones numéricas y lógicas de los datos.

OPERADORES ARITMÉTICOS

Son utilizados para ejecutar cálculos con expresiones numéricas. Las expresiones numéricas son aquellas que permiten ser evaluadas como números, pueden incluir palabras reservas, constantes, variables y operadores que generan como respuesta un número.

Los operadores aritméticos considerados por el lenguaje de programación de Visual son: potenciación (\wedge), suma (+), resta (-), multiplicación (*), división entera (\backslash), división con respuesta decimal (/) y el módulo (Mod) para establecer el residuo de las divisiones.

POTENCIACIÓN Se utiliza para genera el resultado de un número elevado a un exponente y su sintaxis es: Resultado = base \wedge exponente Donde resultado es la variable en la que se asigna la respuesta de la operación de potenciación, por ejemplo:

$5 \wedge 2 \Rightarrow \text{resultado} = 25$
 $2 \wedge 3 \Rightarrow X = 8$

SUMA: Se emplea para generar la adición de dos o más valores numéricos. Su forma de uso es la siguiente: Resultado = sumando 1 + sumando 2 Donde resultado es una variable que almacena la respuesta generada por la suma, por ejemplo:

$20 + 30 \Rightarrow \text{respuesta} = 50$
 $0.35 + 1.20 + 0.50 = 2.05$

RESTA: Este operador indica la diferencia entre dos valores numéricos o la negación de un número. La sintaxis es la siguiente: Resultado = num1 - num2 Donde resultado es la variable a la que se asigna la diferencia de los números, por ejemplo:

$30 - 5 \Rightarrow \text{resp} = 25$
 $5 - 30 \Rightarrow \text{resp1} = -25$

MULTIPLICACIÓN: Se emplea para generar el producto de dos o más expresiones numéricas, su forma de uso es: Resultado = factor1 * factor2 Donde resultado es una variable que forma la respuesta de la operación, por ejemplo:

$7 * 250 = 1750$
 $6 * 3 * 10 = 180$

DIVISIÓN CON PUNTO FLOTANTE: Tiene como propósito dividir dos expresiones numéricas y obtener un resultado con punto flotante es decir con dígitos decimales después del punto. La sintaxis de este operador es: Cuando el operador \wedge aparece más de una vez en una expresión, la operación se realiza de derecha a izquierda $2 \wedge 2 \wedge 3 = 256$ $2 \wedge 8 = 256$ ¡Atención! El operador- (- número) también Puede ser usado como negación Resultado = dividendo / divisor Donde resultado es una variable que almacena un valor en punto flotante generado por la división, por ejemplo:

$7 / 5 = 1.40$
 $10 / 2 = 5.00$
 $35 / 6 = 5.833333$

DIVISIÓN CON RESULTADO ENTERO: Se usa para dividir dos expresiones numéricas y obtener el resultado sin decimales. La sintaxis de esta operación es: Resultado = dividendo \backslash divisor Donde resultado es una variable que almacena un valor entero generado por la división, por ejemplo:

$7 \backslash 5 = 1$
 $11.85 \backslash 2 = 6$
 $10 \backslash 2 = 5$
 $35 \backslash 6 = 5$

RESIDUO DE UNA DIVISIÓN: Se simboliza mediante la palabra reservada Mod y se utiliza para obtener el residuo de una división, la forma de uso es la siguiente: Resultado = dividendo Mod divisor Donde resultado es una variable en la que se asigna un valor entero (residuo de la división), por ejemplo:

$6 \text{ Mod } 3 = 0$
 $10 \text{ Mod } 4 = 2$

Los operadores aritméticos pueden combatirse para realizar cálculos más complejos, por ejemplo:

Total = precio + (IVA * 12 / 100) - descuento
Porcentaje = base * porcen / 100



OPERADORES RELACIONALES: Estos operadores se emplean para comparar dos expresiones y generar un valor verdadero o falso. Sintaxis: Expresión A operador_ relaciones expresión B Expresión A y expresión B pueden ser una combinación de palabras reservadas variables, constantes, números y operadores que generan un valor verdadero o falso por medio de un operador _ relacional. Visual proporciona los siguientes operadores relacionales (la tabla muestra la prioridad en que son evaluados):

PRIORIDAD DE LOS OPERADORES RELACIONALES OPERADOR DESCRIPCIÓN

OPERADOR	DESCRIPCIÓN
=	igualdad
<>	diferencia
<	Menor que
>	Mayor que
<=	Menor o igual que
>=	Mayor o igual que

La prioridad de los operadores relacionales puede modificarse al utilizar paréntesis para agrupar las expresiones que deben ejecutarse en primera instancia.

Note la diferencia entre los operadores para hacer divisiones y obtener resultados con **punto flotante (/)** y **entero (\)**. La prioridad de los operadores relacionales puede modificarse al utilizar paréntesis para agrupar las expresiones que deben ejecutarse en primera instancia

SENTENCIAS

- RANDOMIZE:** la instrucción randomize utiliza number para inicializar el generador de números aleatorios de la función rnd asignándole un nuevo valor de inicialización. si se omite number, el valor devuelto por el temporizador del sistema se utilizará como nuevo valor de inicialización. si no se emplea randomize, la función rnd (sin argumentos) utilizará el mismo número como valor de inicialización la primera vez que se le llame y más adelante utilizará como valor de inicialización el último número generado.

Ejemplo: en este ejemplo se utiliza la instrucción randomize para inicializar el generador de números aleatorios. al haberse omitido el argumento de número, randomize utiliza el valor devuelto por la función timer como nuevo valor de inicialización. copiar:

```
randomize() ('initialize the random-number generator).
dim value as integer = cint(int((6 * rnd()) + 1)) ('generate random value between 1 and 6).
```

- EL CONTROL TIMER:** es un temporizador que nos permite ejecutar instrucciones de código, rutinas, funciones etc..., cada cierto intervalo de tiempo. este control es invisible en tiempo de ejecución, esto quiere decir que no tiene interfaz gráfica, solo es visible cuando lo agregamos a un formulario y estamos en modo de diseño, la propiedad más importante de este control es la propiedad Interval. Propiedad Interval: devuelve o establece el número de milisegundos entre dos llamadas al evento timer de un control timer. en castellano, esto quiere decir que la propiedad es la que determina el intervalo en el que ejecutará las instrucciones que estén en el evento llamado timer, que es el único evento que posee el control.

Ejemplo: utilizando la propiedad interval para mostrar la hora coloca un timer llamado timer1, selecciónalo, y desde la ventana de propiedades en la propiedad interval coloca el valor 1000 y la propiedad enable se activa a true o sea que el timer1 se va a ejecutar cada 1 segundo. Ahora coloca un control label1 en el formulario y en la propiedad visible coloca false, esto hará que el control label1 se ajuste al contenido de este. Lo que hará entonces dale doble click en el timer1 para crear el procedimiento que se describió arriba que es el que ejecuta el timer. y pega esta instrucción

```
Label1.Visible = True
End Sub
```

Ahora ejecuta el programa y puedes ver como se actualiza el label1 de acuerdo con la hora del sistema, usando claro la función time de visual que devuelve la hora del sistema operativo.

otra propiedad importante y que se utiliza mucho en este control es la propiedad enabled. la propiedad enabled lo que hace es habilitar o deshabilitar un control. si está en true funciona, si está en false no funciona.



3. **MSGBOX:** la función msgbox de vbscript, nos permite crear alertas altamente personalizadas en las que podemos configurar no solo el texto, sino que también el título, el icono, los botones, el icono de la barra de título y el botón por defecto. Valores en la siguiente tabla

miembro	valor	descripción
okonly	0	muestra sólo el botón aceptar.
okcancel	1	muestra los botones aceptar y cancelar.
abortretryignore	2	muestra los botones anular, reintentar y omitir.
yesnocancel	3	muestra los botones sí, no y cancelar.
yesno	4	muestra los botones sí y no.
retrycancel	5	muestra los botones reintentar y cancelar.
critical	16	muestra el icono mensaje crítico.
question	32	muestra el icono consulta de advertencia.
exclamation	48	muestra el icono mensaje de advertencia.
information	64	muestra el icono mensaje de información.
defaultbutton1	0	el primer botón es el predeterminado.
defaultbutton2	256	el segundo botón es el predeterminado.
defaultbutton3	512	el tercer botón es el predeterminado.
applicationmodal	0	aplicación modal: el usuario debe responder al cuadro de mensaje antes de continuar trabajando en la aplicación actual.
systemmodal	4096	sistema modal: se suspenden todas las aplicaciones hasta que el usuario responda al cuadro de mensaje.
msgboxsetforeground	65536	especifica la ventana del cuadro de mensaje como ventana de primer plano.
msgboxright	524288	texto alineado a la derecha.
msgboxrtlreading	1048576	especifica que el texto debe aparecer para ser leído de derecha a izquierda en los sistemas árabe y hebreo.

4. **CICLO FOR:** este ciclo es uno de los más usados para repetir una secuencia de instrucciones, sobre todo cuando se conoce la cantidad exacta de veces que se quiere que se ejecute una instrucción simple o compuesta. su formato general es: for varciclo=valorinicial to valorfinal [step incr o decr]
5. **IF:** es utilizada para ejecutar, o no, un bloque de instrucciones de acuerdo con el valor lógico con que fue evaluada la condición.

Sintaxis:
 If condición then
 Instrucción 1
 Instrucción 2
 .
 .
 Instrucción n
 End if

Su funcionamiento comienza al evaluar la condición (es), si son verdaderas se ejecutan las instrucciones iniciales y si estas se cumplen el programa salta la línea end if, la cual indica el final de la sentencia if. En caso contrario se evalúan las otras condiciones que se establecen.



6. **SELECT CASE:** estructura de control para ejecutar un bloque de instrucciones solo cuando el resultado de la comparación de dos expresiones coincida. Es importante mencionar que las instrucciones asociadas al bloque CASE de la expresión coincidente se ejecutaran una vez, y que el control del programa pasara a las siguientes líneas. END SELECT.

Sintaxis:

Select case Expresión X (variable numérica, cadena de caracteres, condiciones)

Case expresión 1

Instruccionloque1

Case expresión 2

Instruccionloque1

Case expresión 2

Instruccionloque1 (lista de posibilidades que puede tomar X)

Case Else

Instrucción lo que Else (serie de instrucciones que permite ejecutar las instrucciones a X en caso de no cumplir con las primeras)

END SELECT

7. **Sentencia DO:** estructura de control que, al igual que la sentencia FOR, genera un ciclo repetitivo; la diferencia radica en que para ejecutar las instrucciones debe evaluarse un condición.

Sintaxis:

Do While condición (variable numérica, cadena de caracteres, condiciones)

Instruccionloque1

Do Until condición (variable numérica, cadena de caracteres, condiciones)

Instruccionloque1

Exit Do (proporciona una salida para el ciclo sin necesidad de cumplir la condición.)

Instruccionloque1

Loop 8.

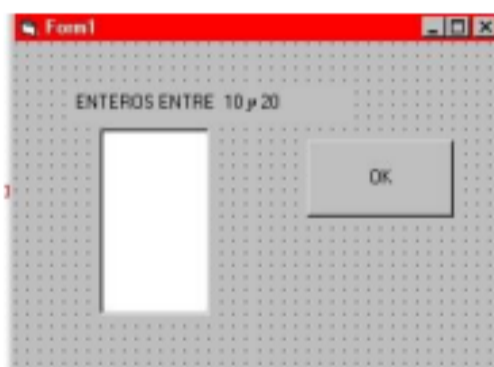
8. **La función Val:** deja de leer la cadena de caracteres inscritos en un text y lo convierte en un valor numérico en caso de que el primer carácter que no puede reconocer como parte de un número se debe tener en cuenta los símbolos y caracteres que se suelen considerar parte de valores numéricos, como signos de moneda y comas, no se reconocen. Sin embargo, la función reconoce los prefijos de base &O (para octal) y &H (para hexadecimal). Los espacios en blanco, los tabuladores y los avances de línea se eliminan del argumento. La función Val sólo reconoce el punto (.) como separador decimal válido.

Sintaxis:

Val (text) (se puede combinar con los operadores aritméticos que se desee)

Elaboraremos algunas aplicaciones con algunas de las sentencias explicadas anteriormente, manos a la obra.

1. Desplegar los números enteros, comprendidos entre el 1 y el 20. Para este problema se debe ubicar en form1, un componente Button (ok) que en su evento click contiene el for y la carga del componente (números) se muestren en un listbox; pantalla de diseño.



Digite el siguiente código en el Button (ok)

```
Dim X As Integer
For X = 10 To 20
    ListBox1.Items.Add(X)
Next X
```



2. Creando una animación sencilla:

1. Baja tres figuras que representen una secuencia.
2. Ingrese al programa visual
3. En el formulario ingrese tres timer y tres picturebox.
4. En cada picturebox inserte cada una de las imágenes que descargo para la secuencia. (debe quedar así)

NOTA: LOS timer se deben activar la propiedad de enabled y en interval digitar en cada uno 2500



5. Los picturebox deben quedar igual de grandes (ajustar el height, width, top y el left).

6. Vamos a programar cada uno de los timer.

7. Haga doble clic en Timer1 y escriba:

```
Picturebox1.Visible = True  
Picturebox3.Visible = False  
Picturebox2.Visible = False  
Timer2.Enabled = True  
Timer1.Enabled = False
```

8. Haga doble clic en Timer2 y escriba:

```
Picturebox2.Visible = True  
Picturebox1.Visible = False  
Picturebox3.Visible = False  
Timer3.Enabled = True  
Timer2.Enabled = False
```

9. Haga doble clic en Timer3 y escriba

```
Picturebox2.Visible = False  
Picturebox3.Visible = True  
Picturebox1.Visible = False  
Timer1.Enabled = True  
Timer3.Enabled = False
```

10. Haga doble clic en el formulario y en el evento LOAD escriba

```
Timer2.Enabled = False  
Timer3.Enabled = False
```

11. Ubica un picturebox encima del otro.

12. Presione F5 y vera una pequeña animación

3. Ingresando el nombre:

En el mismo formulario inserta 2 label, dos textbox y un button así:

Nombre

Apellido

registrar



Vamos a trabajar la propiedad vbInformation del msgbox que nos hace aparecer un mensaje con los datos digitados, y el botón aceptar.

4. Haga doble clic en el botón Registrar y digite las siguientes líneas.

```
MsgBox("El nombre y apellido digitados son: " + Trim(TextBox1.Text) + " " + Trim(TextBox2.Text), vbInformation)
```

4. Promedio de Notas: En el mismo formulario inserte cuatro label, cuatro textbox y un button como aparece a continuación.

Promedio de notas

Primer periodo

Segundo periodo

Tercer periodo

1. Haga doble clic en el botón calcular y digite las siguientes líneas.

```
TextBox6.Text = (Val(TextBox3.Text) + Val(TextBox4.Text) + Val(TextBox5.Text)) / 3
If TextBox6.Text >= 3 Then
    MsgBox("Felicitaciones Aprobo")
End If

If TextBox6.Text < 3 Then
    MsgBox("dedica mas tiempo a estudiar Reprobo")
End If
End Sub
```

2. Verifique el número del textbox para evitar errores en los procedimientos.

3. Presione F5 y corra el programa y vera el cálculo de acuerdo a las notas que usted digite, así como el mensaje correspondiente

Promedio de notas

1 Periodo

2 Periodo

3 Periodo

Proyecto1

Felicitaciones Aprobo

Promedio de notas

1 Periodo

2 Periodo

3 Periodo

Proyecto1

dedicate a estudiar Reprobo

4. Suma de números (serie consecutiva):

1. En el mismo formulario inserte tres label, tres textbox y un button así:

Desde: Hasta: =

2. Limpie los textbox

3. Haga doble clic en el formulario y en el objeto GENERAL evento declaraciones digite las siguientes líneas que como ya se explicó se están declarando las variables necesarias para el programa.

```
Dim n1, n2 As Integer
Dim suma As Double
```




4. Cierre la ventana de código
5. Haga doble clic en el botón calcular y escriba las siguientes líneas.

```
n1 = TextBox7.Text  
n2 = TextBox8.Text  
For n1 = n1 To n2  
    suma = suma + n1  
Next n1  
TextBox9.Text = suma
```

Las aplicaciones se deben ver así:

