

## PRÁCTICA SEIS DE VISUAL BASIC CUARTO PERIODO

### Metas de comprensión

- Desarrolla comprensión al conocer, interpretar, socializar y argumentar diferentes temáticas relacionadas con la tecnología informática de punto (redes, web 1.0, 2.0, 3.0, comunicaciones etc).
- Interpreta y adopta el reglamento del aula de informática.
- Desarrolla comprensión al elaborar diferentes documentos que impliquen el uso hipervínculos, código, HTML y otros lenguajes. (página web)

### **METODOLOGIA DEL AREA**

- Durante la ejecución de cada una de las actividades propias en cada una de las asignaturas que componen el área se tendrán en cuenta los aspectos de asistencia, contextualización del tema, producción individual o grupal, evaluación y/o socialización de los resultados y cumplimiento de las normas.
- **O:** orden y aseo
- **T:** trato digno y respetuoso
- **A:** atención y asistencia
- **C:** cumplimiento
- **U:** útiles y uniforme

## 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 Basic 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 Basic son: potenciación ( $\wedge$ ), suma (+), resta (-), multiplicación (\*), división entera (\), 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:

Resultado =  $5 \wedge 2 \Rightarrow$  resultado = 25

X =  $2 \wedge 3 \Rightarrow$  X = 8

Potencia = X  $\wedge$  y = > potencia = X y

Quando el operador  $\wedge$  aparece más de una vez en una expresión, la operación se realiza de derecha a izquierda

$2 \wedge 2 \wedge 3 = 256$

**SUMA:** Se emplea para generar la adición de dos o más valores numéricos. Su forma de uso es la siguiente:

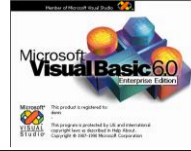
**Resultado = sumando 1 + sumando 2**

Donde resultado es una variable que almacena la respuesta generada por la suma, por ejemplo:

Respuesta = 20 + 30  $\Rightarrow$  respuesta = 50

Porcentaje = 0.35 + 1.20 + 0.50  $\Rightarrow$  porcentaje = 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:

Resp        = 30 - 5    => resp        = 25  
 Resp1     = 5 - 30   => resp1     = 25  
 -resp     =        =>            = 25

¡Atención!  
 El operador-  
 (- número) también  
 Puede ser usado como  
 negación

**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:

Prod        = 7 \* 250    => prod        = 1750  
 Prod\_1    = 6 \* 3 \* 10   => prod\_1    = 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:

**Resultado = dividendo / divisor**

Donde resultado es una variable que almacena un valor en punto flotante generado por la división, por ejemplo:

Resp        = 7 / 5        => resp        = 1.40  
 Div         = 10 / 2       => div         = 5.00  
 División   = 35 / 6       => división   = 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 \ divisor**

Donde resultado es una variable que almacena un valor entero generado por la división por ejemplo:

Div         = 7 \ 5         => div         = 1  
 Div1      = 11.85 \ 2     => div1      = 6  
 Resp      = 10 \ 2       => resp      = 5  
 División = 35 \ 6     => división = 5

Note la diferencia entre  
 los operadores para hacer  
 divisiones y obtener  
 resultados con punto  
 flotante (/) y entero (\).

**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:

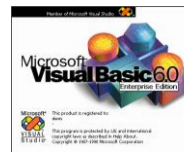
Residuo = 6 **Mod** 3   => residuo = 0  
 Res 1    = 10 **Mod** 4   => res1     = 2

Los operadores aritméticos pueden combinarse 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 expresiónB**



**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 Basic proporciona los siguientes operadores relacionales (la tabla muestra la prioridad en que son evaluados):

PRIORIDAD DE LOS OPERADORES RELACIONALES	
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.

### UNA APLICACIÓN CON OPERADORES ARITMÉTICOS Y RELACIONALES

A continuación se desarrollará un programa que simula una calculadora básica, es decir, que realiza las operaciones matemáticas:

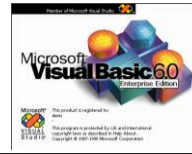
SEGUIMIENTO DEL PROGRAMA	
<p><b>OBJETIVO</b>                      Crear un programa que realice las operaciones matemáticas de suma, resta, multiplicación, división con resultado entero y con punto flotante y modulo, desde una ventana cuya apariencia sea similar a la de una calculadora.</p>	<p><b>Interfaz</b>                      Es necesario un formulario con los siguientes controles:                      Un cuadro de texto <b>TextBox</b> para visualizar los datos procesados.                      Dos marcos <b>Frame</b> para agrupar los botones por funciones.                      Diecinueve botones de comando <b>CommandBuutton</b> para los números, operadores y opciones del programa.</p>
	<p><b>Acciones</b>                      Visualizar los datos que se procesarán en el visor de la calculadora.                      Ejecutar el cálculo requerido según el operador seleccionado.                      Validar posibles errores, por ejemplo, dividir por cero.                      Mostrar el resultado de la operación en el visor.                      Presentar alternativas para nuevos cálculos.                      Finalizar la aplicación al presionar el botón <b>Salir</b>.</p>

Seleccione la opción **Nuevo Proyecto** en el menú **Archivo** para crear una nueva aplicación. Active la ventana **Propiedades** e incluya las siguientes características.

Formulario	
Propiedad	Nuevo valor
Caption	Operaciones básicas
Height	5310
Width	4590

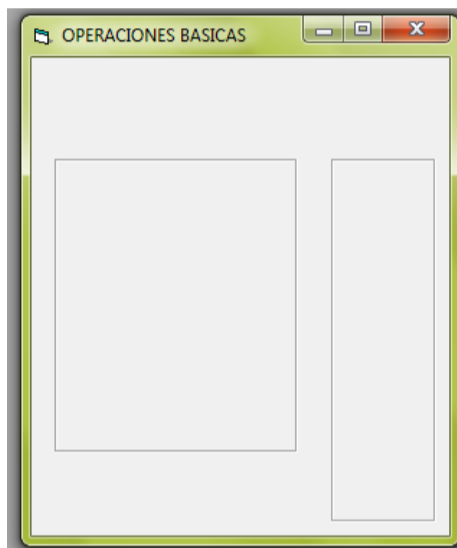
Dibuje dos controles Frame sobre el formulario para agrupar los botones de la calculadora según su funcionalidad (números y operadores) y ajuste las siguientes características:

objeto	Propiedad	valor
--------	-----------	-------



Frame 1	Caption Height Left Top Width	Eliminar el contenido 3135 240 960 2535
Frame 2	Caption Height Left Top Width	Eliminar el contenido 3855 3120 960 1095

El formulario deberá tener un aspecto como el siguiente:

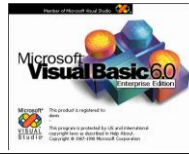


En el cuadro Frame 1 incluya diez botones de comando para formar el teclado numérico de la calculadora, y en Frame2 inserte siete botones de comando para definir cada uno de los operadores aritméticos. Ajuste sus características de ubicación y contenido siguiendo lo propuesto en el formulario.



Tenga en cuenta los siguientes valores para la propiedad **Name** para cada uno de los botones.

BOTÓN	PROPIEDAD NAME
1	n1
2	n2
3	n3

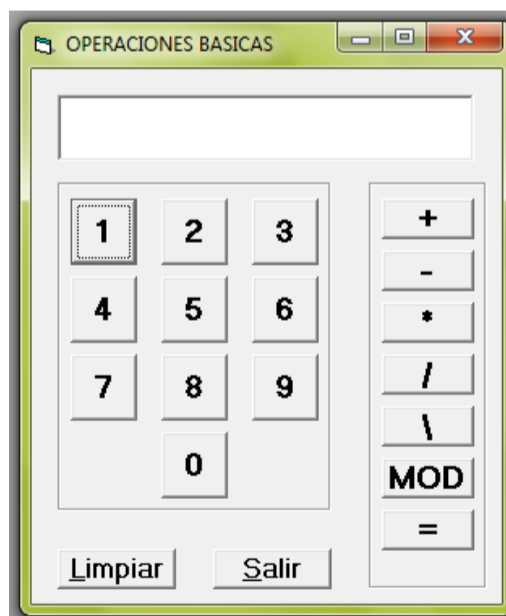


4	n4
5	n5
6	n6
7	n7
8	n8
9	n9
0	n0
+	Suma
-	Resta
*	Multi
/	DivDec
\	DivEnt
MOD	Módulo
=	Igual

Inserte un cuadro de texto TextBox para usarlo como visor de la calculadora, es decir, donde aparecieran los datos con los que se harán los cálculos y los resultados. Ajuste las propiedades del visor, así:

objetivo	propiedad	valor
<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     Para que la alineación indicada en la propiedad <b>Alignment</b> se tenga en cuenta, el valor de <b>MultiLine</b> debe estar en <b>True</b>.                 </div>	TextBox 1	Alignment
	Enabled	False
	Font	Arial Black Regular, 11
	MultiLine	True
	Name	Visor
	Text	Eliminar el contenido

Por último, para controlar la salida de la aplicación o indicar un nuevo cálculo (limpiar el visor y las variables), inserte dos botones de comando sobre el formulario y asigne los valores Limpiar y Salir respectivamente en la propiedad **Name** (**Para que la letra quede subrayada se le antepone & a la palabra en la propiedad caption**)

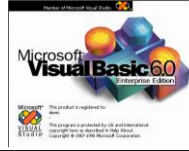


Ahora deben incluirse las instrucciones de programación que le indicarán a la calculadora las operaciones que debe realizar de acuerdo con la acción que ejecute el usuario.

Seleccione la opción **Código** en el menú **Ver** para desplegar la ventana **Código**.

Escoja la opción **[General]** en la lista Objeto:, [ declaraciones] en Proc: y defina las variables que se utilizarán en el programa.

Dim opera As Byte



Dim num1, num2 as double

La variable **opera es de tipo Byte ya** que se le asignará un valor numérico que identifica la operación que va a realizarse, num1 almacenara el contenido existente en el visor antes de seleccionar un operador y num2 contendrá el valor que hay en el visor después de presionar un operador; por ejemplo, si va a realizarse el cálculo  $50 * 23$ , al presionar el operador = la variable num1 valdrá 50 y num2 contendrá 23.

Seleccione **Form** (ventana de código) en la lista de objetos: y el evento Load para inicializar las variables num1, num2 en cero (0). Copie las siguientes líneas.

```
Num1=0  
Num2=0
```

Cada vez que el usuario haga clic en los botones con números, el valor debe agregarse en el visor, es decir, para esto escriba el siguiente código en el evento clic de los botones (objetos) n0, n1, n2, n3, n4, n5, n6, n7, n8, n9

```
Private sub n0_clic()  
Visor.text = visor.text + Str (0)  
End sub
```

```
Private sub n1_clic()  
Visor.text = visor.text + Str (1)  
End sub
```

```
Private sub n2_clic()  
Visor.text = visor.text + Str (2)  
End sub
```

```
Private sub n3_clic()  
Visor.text = visor.text + Str (3)  
End sub
```

```
Private sub n4_clic()  
Visor.text = visor.text + Str (4)  
End sub
```

```
Private sub n5_clic()  
Visor.text = visor.text + Str (5)  
End sub
```

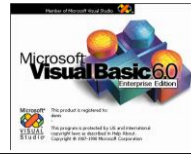
```
Private sub n6_clic()  
Visor.text = visor.text + Str (6)  
End sub
```

```
Private sub n7_clic()  
Visor.text = visor.text + Str (7)  
End sub
```

```
Private sub n8_clic()  
Visor.text = visor.text + Str 8)  
End sub
```

```
Private sub n9_clic()  
Visor.text = visor.text + Str 9)  
End sub
```

La función **Str**  
convierte un valor  
numérico en una  
cadena de  
caràcteres.



En el momento en que el usuario presione un operador, el valor contenido en el visor debe almacenarse en una variable, limpiarse el visor e identificar el operador seleccionado para realizar el cálculo correcto. Incluya el siguiente código en el **evento clic de los objetos sum, resta, multi, divdec, divEnt, y Modulo**.

```
Private sub suma_clic()  
Num1 = val (visor.text)  
Opera=1  
Visor.text = " "  
End sub
```

```
Private sub resta_clic()  
Num1 = val (visor.text)  
Opera=2  
Visor.text = " "  
End sub
```

```
Private sub multi_clic()  
Num1 = val (visor.text)  
Opera=3  
Visor.text = " "  
End sub
```

```
Private sub DivDec_clic()  
Num1 = val (visor.text)  
Opera=4  
Visor.text = " "  
End sub
```

```
Private sub DivEnt_clic()  
Num1 = val (visor.text)  
Opera=5  
Visor.text = " "  
End sub
```

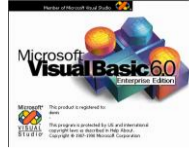
```
Private sub modulo_clic()  
Num1 = val (visor.text)  
Opera=6  
Visor.text = " "  
End sub
```

En los procedimientos anteriores se hace necesario el uso de **VAL** para convertir el contenido del Visor (texto) en un valor numérico, compatible con el tipo de datos de la variable num1.

En la ventana de código, seleccione el objeto Igual (command) y el evento clic y digite el siguiente código para asociarse el cálculo al operador seleccionado de acuerdo a los valores proporcionados.

La sentencia **IF** compara el contenido de la variable **opera** con valores entre 1 y 6 para determinar cual operación realizar. La clausula IF será trabada con más frecuencia.

```
Private sub igual_clic()
```



```
Resp = 0  
Num2 = val (visor.text)
```

```
If Opera = 1 then  
Resp = num1 + num2  
End if
```

```
If Opera = 2 then  
Resp = num1 - num2  
End if
```

```
If Opera = 3 then  
Resp = num1 * num2  
End if
```

```
If Opera = 4 then  
If num2 = 0 then  
Msgbox "error...no puede dividirse por 0"  
Exit sub (finaliza la ejecución)  
End if  
Resp = num1 / num2  
End if
```

```
If Opera = 5 then  
If num2 = 0 then  
Msgbox "error...no puede dividirse por 0"  
Exit sub (finaliza la ejecución)  
End if  
Resp = num1 \ num2  
End if
```

```
If Opera = 6 then  
If num2 = 0 then  
Msgbox "error...no puede dividirse por 0"  
Exit sub (finaliza la ejecución)  
End if  
Resp = num1 Mod num2  
End if  
Visor.text=resp (visualiza la respuesta)  
End sub
```

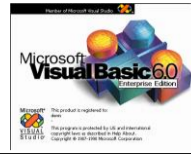
Cada vez que se ejecute el procedimiento **Igual\_Click**, el valor de la variable resp se inicia con cero para después asignarle el resultado de la operación que toma el valor de las variables num1 y num2 y se realiza el cálculo correspondiente. Cuando se va a hacer una división se está validando que el divisor sea diferente de cero (0) para que no se presenten errores en la ejecución; se muestra un mensaje con MsgBox y se sale del procedimiento sin realizar ninguna acción con la institución Exit Sub.

Para limpiar el contenido del visor y de las variables agregue el siguiente código al evento Click del objeto Limpiar.

```
Visor.text = ""  
Num1=0  
Num2=0  
resp=0
```

La instrucción **EXIT SUB** finaliza la ejecución del procedimiento, es decir salta a la siguiente línea de **END SUB**





Para finalizar la programación de eventos seleccione la opción **Salir** en la lista **Objeto:** y el evento **Click** en **Proc:**, luego introduzca las instrucciones que produce la salida del programa.

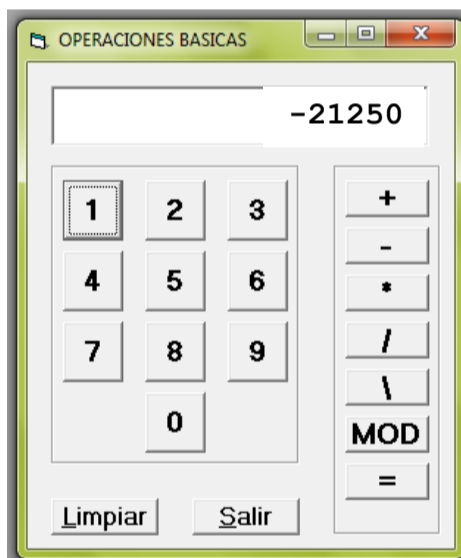
```
Private Sub Salir_Click ( )  
    End  
End Sub
```

Presione el botón **Cerrar** para desactivar la ventana **Código**.

Grabe el proyecto seleccionando la opción **Guardar proyecto** en el menú **Archivo**. Introduzca como nombre básico en las ventanas **Guardar archivo como** y **Guardar proyecto como**.

Para ejecutar el programa, seleccione la opción **Iniciar** en el menú **Ejecutar** y aparecerá la ventana de la aplicación.

Compruebe que al presionar cada uno de los botones con números aparece el valor asociado en el visor y que los cálculos corresponden según el operador seleccionado. Por ejemplo, multiplique 850 por 25 y observe el resultado.



Resultado -21250

Tenga en cuenta que cada vez que realice una operación debe presionar el botón = para obtener el resultado correcto, por ejemplo, si desea calcular  $12 - 4 + 8 * 2$  debe presionar el botón = después del 4, el 8 y el 2.