

PRÁCTICA TRES DE VISUAL BASIC

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

Tipos de variables en Visual Basic

Los distintos tipos de variables utilizados en Visual Basic son:

Integer	Valor Entero	2 Bytes
Long	Valor Entero Largo	4 Bytes
Single	Valor Real	4 Bytes
Double	Valor Real Doble	8 Bytes
String	Carácter (texto)	1 Byte por carácter
Byte	Byte	1 Byte
Boolean	Valor Booleano	(1/0) 2 Bytes
Currency	Monedas y Punto Fijo	8 Bytes
Date	Fecha	8 Bytes
Object	Referencias a objetos	4 Bytes
Variant	Cualquiera	16-22 Bytes

Estas formas son extensibles para las variables tipo Integer, Long, Single, Double, String y Currency.

BYTE: Indica que el valor almacenado será un numero entero positivo entre 0 y 255 por ejemplo al declarar las siguientes variables:

Dim a,b,c As byte

Algunos valores que estas aceptarían son:

A= 50
 B= 0
 C=255

BOOLEAN: Este tipo de datos es usado para señalar que el valor almacenado será un dato lógico (**TRUE** o **FALSE**). Por ejemplo si se realiza la siguiente declaración:

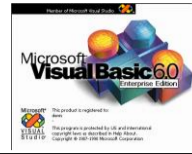
Dim activado As Boolean

La variable **Activado** solo puede tomar uno de estos valores:

Activado=true
 Activado=false

CURRENCY: Establece que la información tendrá un formato de números enteros (positivos y negativos) con un máximo de 15 dígitos a la izquierda del punto decimal y 4 después de este. Reserva 8 Bytes en la memoria y generalmente es utilizada con variables que requieren precisión.

DATE: Hace referencia a la información de fecha y hora, donde el rango para cada una es: **fecha: 01/01/100 y 31/12/9999** para día/mes/año **Hora:0:00:00 y 23:59:59** para horas/minutos/segundos. Por ejemplo, al declarar las siguientes variables:



Dim fecha_nacimiento, hora_ingreso As Date

Algunos posibles valores que pueden tomar son:

Fecha_nacimiento="15/03/1972"
 Hora_ingreso="10:30:25"

DOUBLE: Usando para indicar que un valor será un número real o flotante de doble precisión, por ejemplo, al declarar:

Dim nota As Double

Nota podría tomar uno de estos valores:

Nota=3.65
 Nota=0.18
 Nota=18532320.463

INTEGER: Indica que se guardarán números enteros (positivos y negativos) comprendidos entre -32768 y 32767. Por ejemplo, al declarar la siguiente variable:

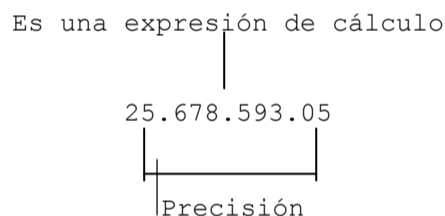
Dim numero_registro As integer

Numero_registro podría tomar los siguientes valores:

Numero_registro= 100
 Numero_registro=12500

LONG: Almacén valores numéricos (enteros positivos y negativos) que reservan 4 bytes en la memoria.

SINGLE: Define que un dato será un número flotante de precisión simple es decir que el grado de precisión está sujeto a siete dígitos por ejemplo si como resultado de un cálculo se genera el siguiente número se tendría:



STING (CADENAS): Tipo de datos para declarar variables, constante y otros que almacenen una secuencia de caracteres (letras, números, símbolos o una combinación de ellos).

Algunos ejemplos de **String** podrían ser:

Variables de tipos STRING

Nombre
 Dirección
 Edad
 Ciudad
 País

VALO

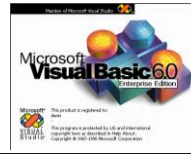
Aluissio affonso
 Av. Assembleia, 12-4240
 25
 Rio de janeiro
 Brasil

VARIANT: Almacena todos los tipos de tipos de datos, excepto cadenas de longitud fija y datos definidos por el usuario; además, efectúa automáticamente las conversiones necesarias según el tipo de datos de esta; por ejemplo, al declarar **Var1** como **Variant** podría tomar cualquier valor.

Dim Var1 As Variant

Var1 podría tomar los siguientes valores:

Var1= 8520
 Var1= "feliz navidad"
 Var1= 65386763.234



TYPE: Define un tipo de datos personalizados por el usuario. Solo puede usarse a nivel de modulo y la sintaxis de su declaración es la siguiente:

[PUBLIC | PRIVATE] TYPE nomtipo

Elemento1 **As** tipo

Elemento2 **As** tipo

..

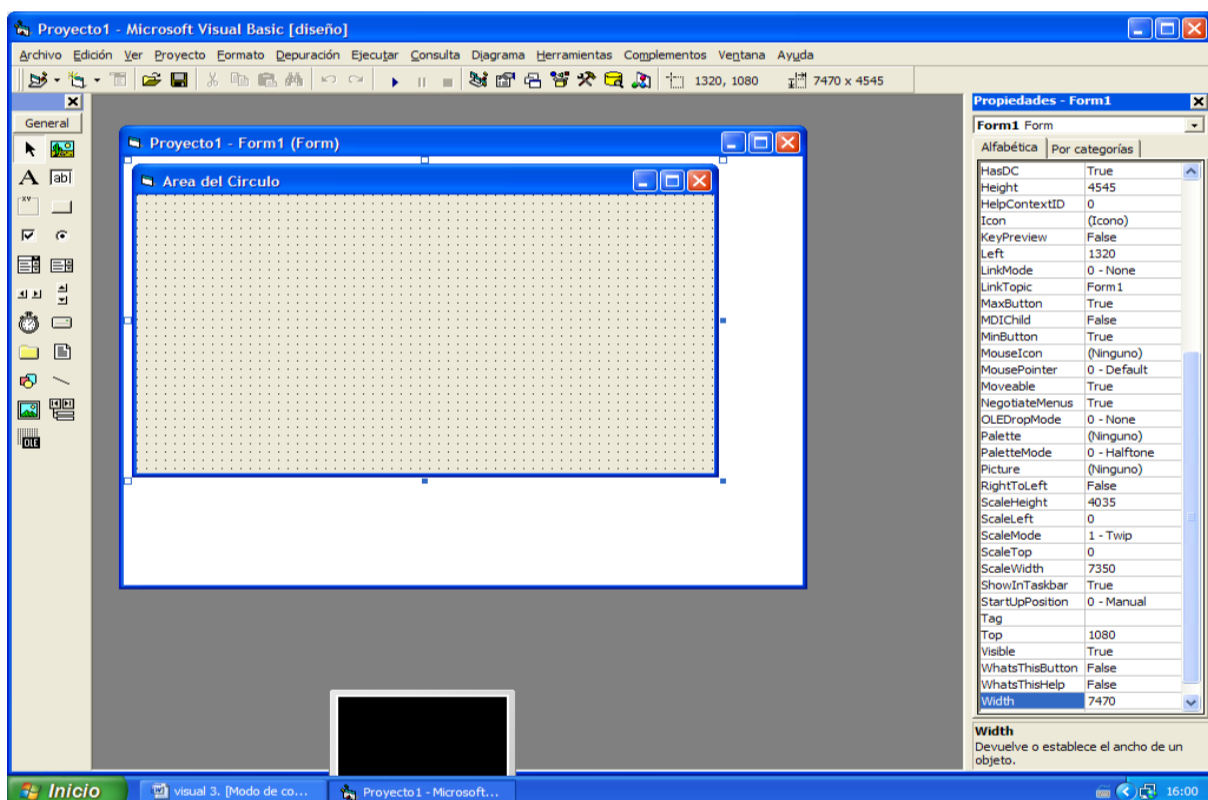
elementoN **As** tipo

End Type

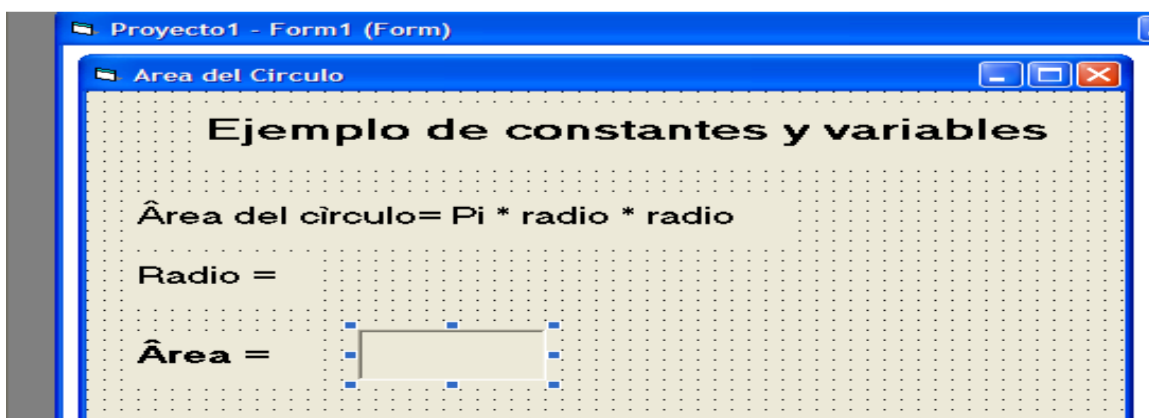
EJEMPLO CON VARIABLE Y CONSTANTE

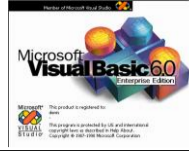
Se realizara el ejercicio sencillo como lo es calcular el área del círculo, en el cual se declara **constante Pi** y **variable el Radio**.

1. Ingrese al programa visual Basic
2. Active la ventana propiedades (formulario) y cambie los siguientes campos
 - Caption Área del círculo
 - Height 4545
 - Left 1320
 - Top 1080
 - Width 7470

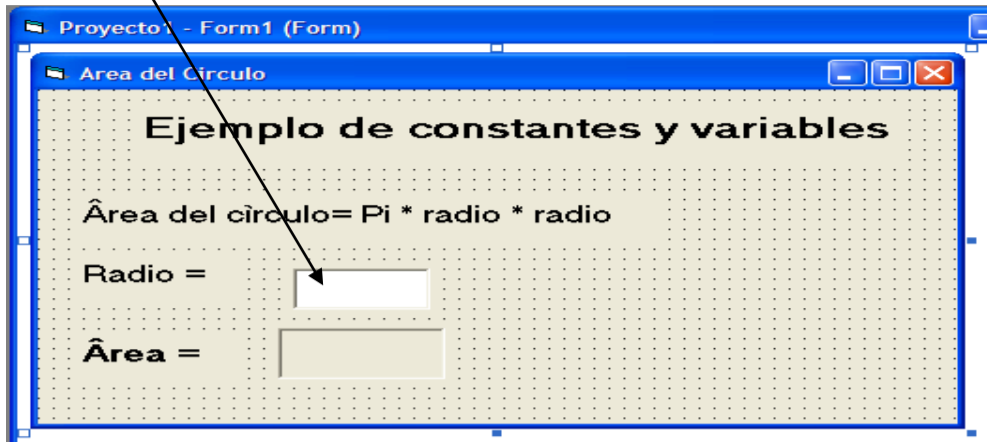


3. Inserte cinco controles LABEL sobre el formulario para que queden así (aplique los cambios que sean necesarios):



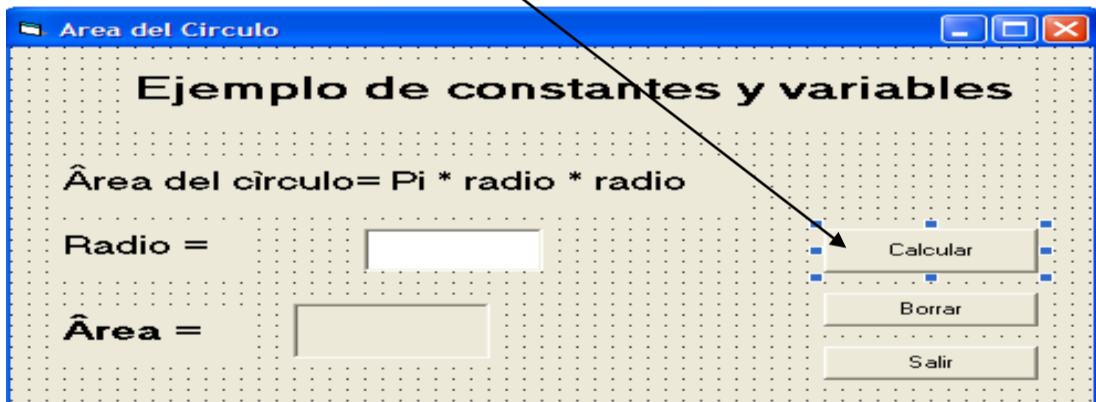


4. Dibuje un TextBox para que el usuario introduzca el valor del radio y ajuste sus características así:



- Font negrilla, 14
- Height 420
- Left 2400
- Text borrar el contenido
- Top 2040
- Width 1215

5. Inserta tres botones CommandButton en el formulario para calcular, borrar y salir del programa. Ubíquelos como se muestra en el formulario. El contenido de cada botón se cambia en la opción Caption



6. Haga doble clic sobre el formulario y active la ventana de código, seleccione en la lista la opción **General** y en el procedimiento **declaraciones para definir la variable** y escriba el siguiente código.

```
Dim radio as double
Const Pi =3.1416
```

7. Seleccione `Text1` de la lista de objetos y en los procedimientos `change` y escriba este código.

```
If text1.Text<>"" then
Radio=Text1.Text
End if
```

8. Seleccione `Text1` de la lista de objetos y en los procedimientos `KeyPress` y escriba este código. Este código permite que el usuario solo digite números en caso de una letra aparece un error.

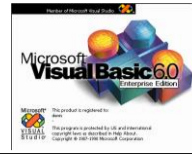
```
If KeyAscii <48 or KeyAscii>57 then    \(código Ascii para los números\)
KeyAscii=0
End if
```

9. Seleccione `Command1` de la lista de objetos y en los procedimientos `Click` y escriba este código. Este procedimiento para calcular el área del círculo.

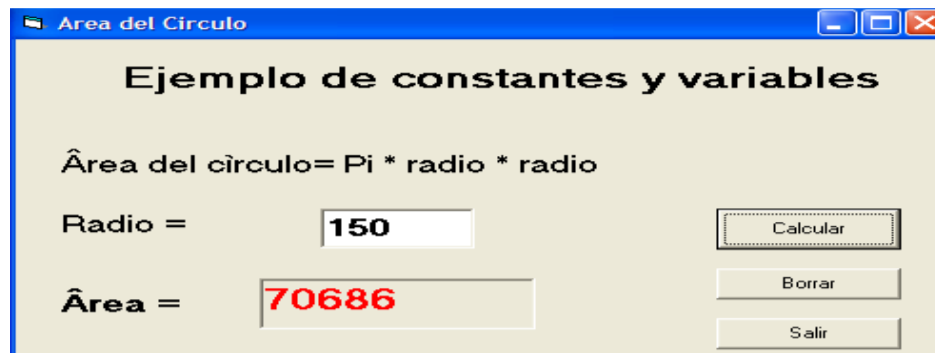
```
Label5.Caption=Pi*radio*radio
```

10. Seleccione `Command2` de la lista de objetos y en los procedimientos `Click` y escriba este código. Este procedimiento para limpiar el valor del radio y de su resultado.

```
Radio=0
Label5.Caption=""
Text1.text=""
```



11. Seleccione Command3 de la lista de objetos y en los procedimientos Click y escriba este código.
Este procedimiento para salir de la aplicación.
End.
12. Presione la tecla f5 y ejecute el programa y digite en el radio 100 la respuesta será.



Ejercicio práctico. Calcula el área de las siguientes figuras: (No olvide guardarlos en su carpeta de trabajo, siempre se relaciona el proyecto con el nombre de la aplicación)

- Cuadrado
- Triángulo
- Trapecio
- Rectángulo
- Esfera