

OPERADORES EN VISUAL BASIC (I)

INTRODUCCIÓN

Incluimos una pequeña introducción al programa para aquellos profesores que quieran iniciarse en la programación a través de este entorno.

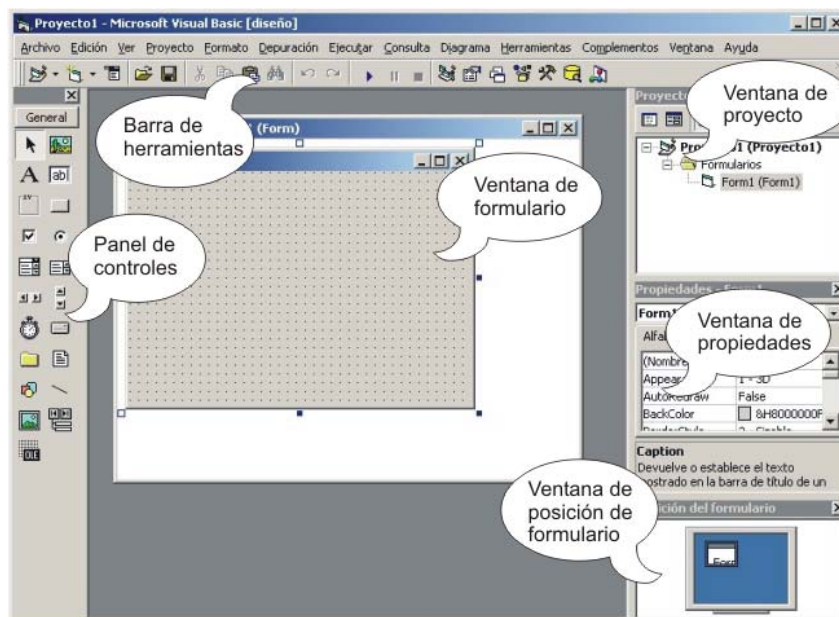
Para esto vamos a trabajar en pequeños ejemplos y explicaremos únicamente los elementos que se utilizan en los ejemplos que proponemos. Si hay cuestiones que se plantean a partir de estos experimentos, serán discutidas y aclaradas a través de los foros.

Los ejemplos de programación que propondremos están orientados al trabajo en el aula y esperamos que vuestras sugerencias y aportaciones nos ayuden a reunir un material que podamos intercambiar y compartir entre aquellos interesados en crear programas adaptados a las necesidades concretas de nuestro trabajo educativo.

ENTORNO DE PROGRAMACIÓN EN VISUAL

Interfaz de Visual

Al abrir el programa por primera vez, nos encontramos con una pantalla en la que se nos proponen distintas opciones. Aceptamos la opción por defecto para empezar.



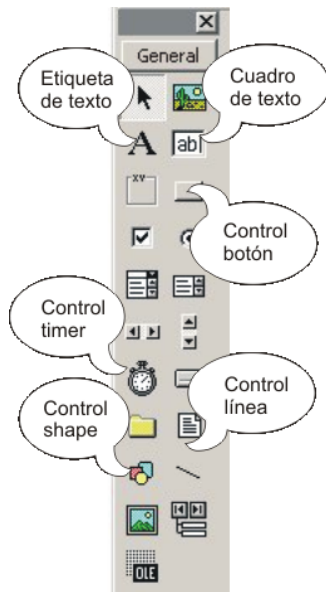
Ventana de formulario: En ella aparece el primer formulario de nuestra aplicación. Mediante los cuadros de selección podemos cambiar su tamaño. En la ventana de propiedades podemos definir la apariencia que presentará al usuario así como otras características que iremos viendo.

Ventana de proyecto: Nos muestra los elementos que contiene nuestra aplicación. Haciendo doble clic sobre ellos, se abrirán en la ventana principal.

Ventana de propiedades: Muestra las propiedades del elemento seleccionado y permite cambiarlas.

Ventana de posición de formulario: Nos permite decidir la posición que ocupará el formulario cuando se ejecute la aplicación.

Panel de controles



Contiene los objetos (controles) que podemos insertaren el formulario. Para nuestro primer ejemplo vamos a utilizar los que están señalados.

- Etiqueta de texto:** Escribe texto en el formulario
- Cuadro de texto:** Crea un espacio donde puede escribir el usuario
- Botón:** nos permite dibujar un botón
- Línea:** nos permite dibujar líneas
- Shape:** nos permite dibujar figuras geométricas
- Timer:** introduce una llamada al reloj del sistema para incluir eventos a lo largo de un espacio temporal.

OPERADORES

Abordamos la realización de aplicaciones en visual que utilizan operaciones matemáticas. Visual utiliza los siguientes símbolos para enlazar las partes de una fórmula

Operadores elementales:

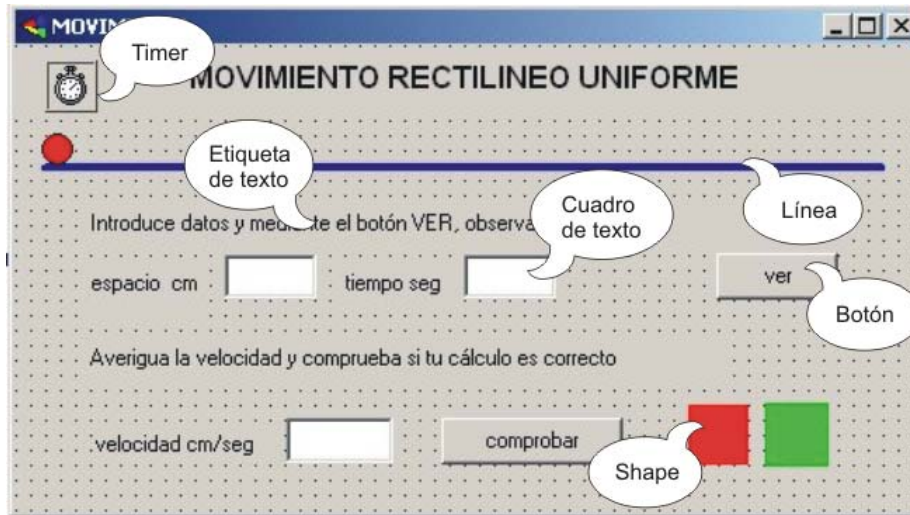
operador	Operación matemática
+	suma
-	resta
*	multiplicación
/	división
\	división entera
Mod	resto de la división entera
^	exponenciación
&	Concatenación de cadenas

Función Val

La función Val convierte una cadena de texto en un valor numérico. Esta operación es necesaria para que el programa realice las operaciones correctamente y no muestre un error o concatene los caracteres en el caso de que el operador sea + por ejemplo.

Ejemplo1: movimiento rectilíneo uniforme (velocidad.exe)

Comenzamos con un pequeño programa que incluye un solo formulario con los controles que hemos señalado en el apartado anterior y que nos permite incluir una fórmula sencilla en el código.



Al insertar cada control y mientras está seleccionado, podemos modificar las propiedades en la ventana de propiedades.

Propiedades de timer

Enabled=false
Interval=1

Propiedades de etiqueta de texto

Caption = (texto que queremos que aparezca en el formulario)
Font = (formato del texto)

Propiedades de cuadro de texto

Text = (vacío)

Propiedades de shape1 (bolita)

BackColor = (elegir el color deseado en la ventana paleta que se activa desde la propiedad)
BackStyle = 1 Opaque
Shape = 3 Circle
Visible = true

Propiedades de shape2 (aviso rojo)

BackColor = (elegir el color deseado en la ventana paleta que se activa desde la propiedad)
BackStyle = 1 Opaque
Shape = 0 Rectangle
Visible = false

Propiedades de shape3 (aviso verde)

BackColor = (elegir el color deseado en la ventana paleta que se activa desde la propiedad)
BackStyle = 1 Opaque
Shape = 0 Rectangle
Visible = false

Propiedades de línea

BorderColor = (elegir el color deseado en la ventana paleta que se activa desde la propiedad)
BorderStyle=1 Solid
BorderWidth = (elegir el ancho de la línea en puntos)

Propiedades de botón

Caption = (etiqueta que queremos que lleve el botón).

Para el resto de las propiedades podemos dejar los valores por defecto. La ubicación y tamaño de los controles dentro del formulario podemos definirla seleccionando y arrastrando.

CÓDIGO

Para insertar código hacemos doble clic en el objeto al que queremos aplicar el código. Visual escribe por defecto la entrada y el final del código y nos sitúa en el espacio en el que debemos escribir.



En las listas desplegables podemos seleccionar el objeto y el evento cambiando aquellos que el programa nos da como predeterminados.

El código para nuestro programa es el siguiente.

Botón ACCIÓN

<pre>Private Sub Command1_Click() Timer1.Enabled = True End Sub</pre>	<p>Al hacer clic en el botón Se activa el control timer (tiempo) fin</p>
---	--

Control Timer

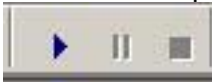
<pre>Private Sub Timer1_Timer() On Error GoTo ver espacio = Val(Text1.Text) tiempo = Val(Text2.Text) If Shape1.Left < 6720 Then Shape1.Left = Shape1.Left + espacio \ tiempo Else Timer1.Enabled = False Shape1.Left = 120 End If Exit Sub</pre>	<p>Al iniciar el control timer (tiempo) Si hay un error ir a ver Definimos espacio como el valor numérico del contenido del cuadro de texto(1) Definimos tiempo como el valor numérico del contenido del cuadro de texto(2) Si la propiedad left del círculo es menor que (aquí el punto final del movimiento) La propiedad left del círculo será igual a la posición actual + el resultado de dividir (división entera) el contenido del cuadro de texto espacio por el contenido del cuadro de texto tiempo Si no (es decir si el círculo llega al final del formulario) El control timer se desactiva La propiedad left es la que corresponde al punto inicial. Fin del condicional Salida</p>
<pre>ver: If Err.Number = 13 Then Timer1.Enabled = False MsgBox ("faltan datos o no son correctos") Else Timer1.Enabled = False MsgBox ("faltan datos o no son correctos") End If Exit Sub End Sub</pre>	<p>Error Si se trata del error 13 Se detiene el control timer y aparece un mensaje Si el error es cualquier otro Lo mismo Fin del condicional Salida Fin</p>

Botón COMPROBAR

<pre>Private Sub Command2_Click() On Error GoTo mensaje espacio = Val(Text1.Text) tiempo = Val(Text2.Text) veloc = Val(Text3.Text) If veloc = espacio \ tiempo Then Shape3.Visible = True Shape2.Visible = False Else Shape3.Visible = False Shape2.Visible = True End If Exit Sub</pre>	<p>Al hacer clic en el botón Si hay un error ir a mensaje Definimos espacio como el valor numérico del contenido del cuadro de texto(1) Definimos tiempo como el valor numérico del contenido del cuadro de texto(2) Definimos veloc como el valor numérico del contenido del cuadro de texto(3) Si $veloc=espacio\ tiempo$ (división entera) Se hace visible el cuadrado verde Se hace invisible el cuadrado rojo Si no Se hace visible el cuadrado rojo Se hace invisible el cuadrado verde Fin Salida</p>
<pre>mensaje: If Err.Number = 13 Then Shape3.Visible = False Shape2.Visible = False MsgBox ("faltan datos o no son correctos") Else Shape3.Visible = False Shape2.Visible = False MsgBox ("faltan datos o no son correctos") End If Exit Sub End Sub</pre>	<p>Error Si aparece un error (13) Se hacen invisibles los cuadrados Y aparece un mensaje Si no Lo mismo Fin Salida Final</p>

Cerramos la ventana de código.

Probamos el programa mediante el icono de control de la barra de herramientas



Una vez comprobado que el funcionamiento es correcto, creamos el ejecutable en el menú Archivo: Generar programa.exe

RESUMEN

1. Abrir Visual
2. Crear la interfaz del programa (agregar controles al formulario)
3. Modificar las propiedades de los controles
4. Introducir código
5. Probar el funcionamiento del programa
6. Generar el ejecutable