

C.F.G.S. DESARROLLO DE APLICACIONES INFORMÁTICAS

MÓDULO: Diseño y realización de servicios de presentación en entornos gráficos

Unidad 1

Introducción Visual Basic .NET

El objetivo de la unidad es conocer el entorno de Visual Basic .NET. Veremos cómo crear nuestro primero proyecto, dibujando algunos controles básicos en el formulario de inicio, así como proporcionarle código a los eventos o acciones de dichos controles. También nos iniciaremos en el uso de la depuración del proyecto y de la ayuda.



ÍNDICE DE CONTENIDOS

¿QUÉ ES VISUAL BASIC .NET?	3
¿DÓNDE PUEDO OBTENER VISUAL BASIC .NET?	3
INICIO CON VISUAL BASIC EXPRESS EDITION.....	3
ENTORNO DE DESARROLLO DE VISUAL BASIC. FORMULARIOS.....	3
DIBUJAR CONTROLES EN EL FORMULARIO.....	5
PROPIEDADES DE LOS CONTROLES	6
AÑADIR CÓDIGO A LOS CONTROLES.....	8
GUARDAR EL PROYECTO	12
UTILIZACIÓN DEL DEPURADOR DE PROYECTOS	12
UTILIZACIÓN DE LA AYUDA.....	13



¿Qué es Visual Basic .NET?

Microsoft Visual Basic 2005 es un lenguaje de programación que está diseñado para crear aplicaciones orientadas a objetos en el entorno del sistema operativo Microsoft Windows. Está implementado sobre el framework .NET que es un amplio proyecto de Microsoft para crear una plataforma que permita un cómodo y rápido desarrollo de aplicaciones.

¿Dónde puedo obtener Visual Basic .NET?

Microsoft nos ofrece una versión gratuita de la herramienta llamada "Visual Basic Express Edition". Podemos descargarla en la dirección web siguiente:

<http://www.microsoft.com/spanish/msdn/vstudio/express/VB/default.msp>

Es importante seguir las instrucciones que te indican en la página web para conseguir la correcta instalación de la herramienta en tu equipo.

Visual Basic Express Edition es funcional durante 30 días. Pero si te registras ya estará operativo de forma indefinida. Este registro es gratuito, sólo necesitaras una cuenta de correo donde te enviarán el número de registro.

En esta página puedes encontrar mucha información sobre el Visual Basic que te será muy útil tanto para el principiante como para el programador ya experimentado. No pierdas el enlace y navega por ella cuando dispongas de tiempo.

Inicio con Visual Basic Express Edition



Una vez instalada la herramienta, la ejecutamos. Aparecerá la pantalla de bienvenida del Visual Basic Express. En ella tenemos también enlaces a distinta información para los usuarios de Visual Basic.

A continuación vamos a la opción del menú "Archivo" y seleccionamos "Nuevo Proyecto". Aparecerá un cuadro de diálogo, seleccionamos "Aplicación para Windows" en "Plantillas instaladas de Visual Studio". El nombre por defecto que asigna al proyecto es "WindowsApplication1", vamos a cambiar ese nombre por "MiPrimerPrograma", y hacemos clic en el botón "Aceptar".

Entorno de desarrollo de Visual Basic. Formularios.

Una vez iniciado el proyecto nos mostrará el entorno de desarrollo de aplicaciones (fig.1). Estamos en la pestaña de diseño: "Form1.vb [Diseño]", y en primer término vemos el "formulario principal" o "formulario de inicio", identificado en su parte superior con el nombre "Form1". Este formulario es el que se mostrará cuando ejecutemos la aplicación. En él dibujaremos los objetos o controles que necesitemos para nuestro proyecto.

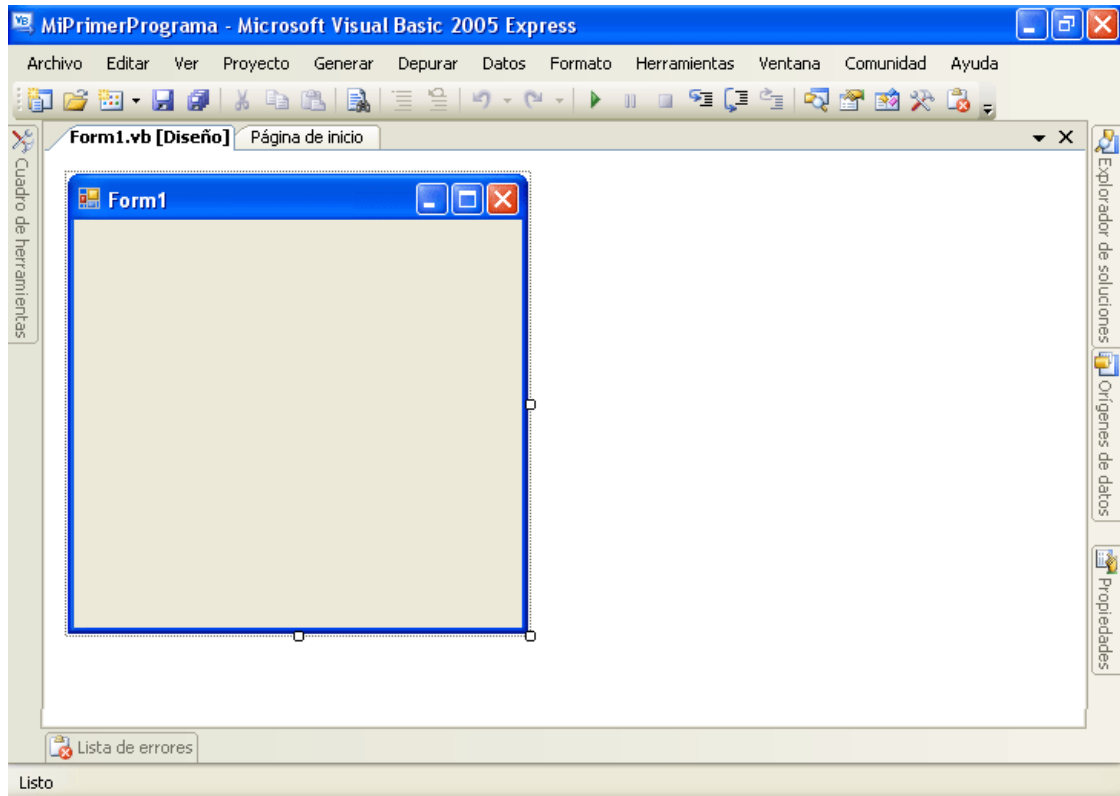


Fig.1. Entorno de desarrollo de Visual Basic

En la parte izquierda tenemos el “cuadro de herramientas” (fig.2). Si situamos el ratón sobre la pestaña (sin necesidad de hacer clic) se desplegará el cuadro con los controles disponibles. Los controles constituyen aquellos elementos que insertamos dentro de un formulario, y que permiten al mismo interactuar con el usuario, tales como botones de pulsación, cajas de texto, casillas de verificación, cajas con listas de valores, etc.

En la ficha “Controles comunes” tenemos los controles más utilizados.

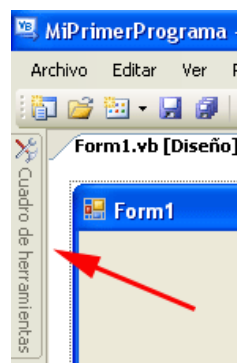


Fig.2. Pestaña cuadro de herramientas

Si no ves la pestaña de la figura anterior (fig.2.). También puedes acceder al cuadro de herramientas desde el icono de la barra de herramientas representado en la figura siguiente (fig.3):

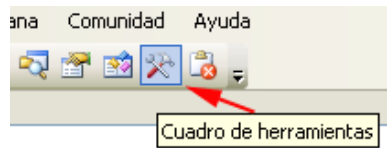


Fig.3. Icono "cuadro de herramientas" de la barra de herramientas.

Dibujar controles en el formulario

Vamos a dibujar un control "TextBox" y un control "Button" (fig.4). Hay varias maneras de realizar este proceso, una de ellas es haciendo doble clic sobre los controles en el cuadro de herramientas. Primero en el control "Button" y después en el control "TextBox", que están señalados en la siguiente figura:

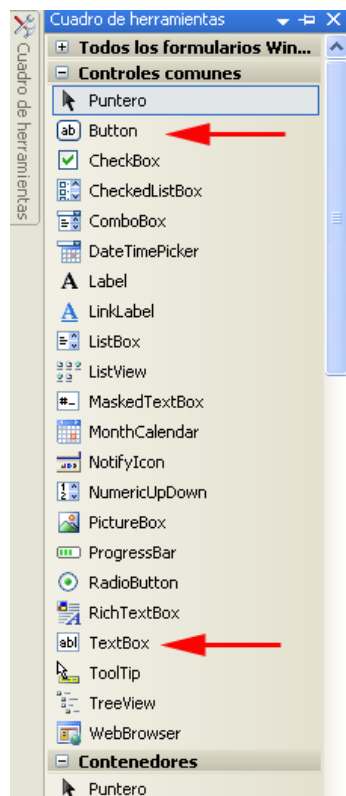


Fig.4. Cuadro de herramientas. Controles Button y Textbox.

Ahora volvemos al formulario "Form1", moviendo el cursor del ratón hacia él se pliega el "cuadro de herramientas" y podemos ver los dos controles en la esquina superior izquierda, uno encima del otro. Uno a uno, los seleccionamos y arrastramos con el ratón hacia el lugar del formulario que consideremos mas apropiado. (fig.5).

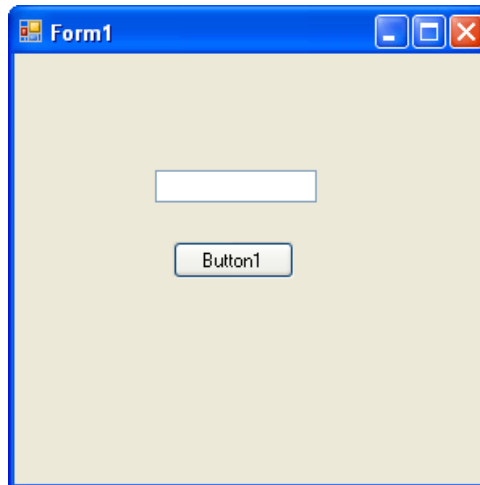


Fig.5. Formulario con los dos controles.

Si nos hemos equivocado y deseamos borrar controles del formulario, simplemente seleccionamos el control a borrar (haciendo clic sobre él) y pulsamos la tecla "Supr" (suprimir).

El control que hemos creado, "Button1", es un objeto de la clase Button, y representa un típico botón de Windows. Y el otro elemento, "TextBox1", es un objeto de la clase TextBox, y representa un cuadro de texto de Windows.

Otras formas de dibujar controles en el formulario son:

- Haciendo clic sobre el icono del control, y a continuación clic sobre la superficie del formulario. El nuevo control se insertará desde el punto en que hemos pulsado, extendiéndose hacia la derecha y abajo con un tamaño predeterminado por el diseñador.
- Haciendo clic sobre el icono del control, y seguidamente clic sobre la posición del formulario en la que queramos comenzar a dibujar el control, arrastraremos y soltaremos dando nosotros el tamaño requerido al control.

Propiedades de los controles

Todos los controles que dibujamos en el formulario tienen un conjunto de propiedades que identifican a cada control y definirán su aspecto y su comportamiento. Vamos a acceder a la ventana de las propiedades de los controles haciendo clic en el icono de la barra de herramientas (fig.6):

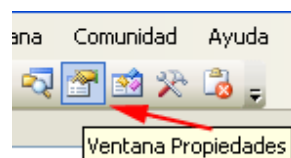


Fig.6. Icono de la ventana de propiedades.

También podemos acceder a la ventana de propiedades seleccionando la opción *Ver + Ventana Propiedades* del menú o pulsando F4.

La ventana que nos aparecerá será la siguiente (fig.7):

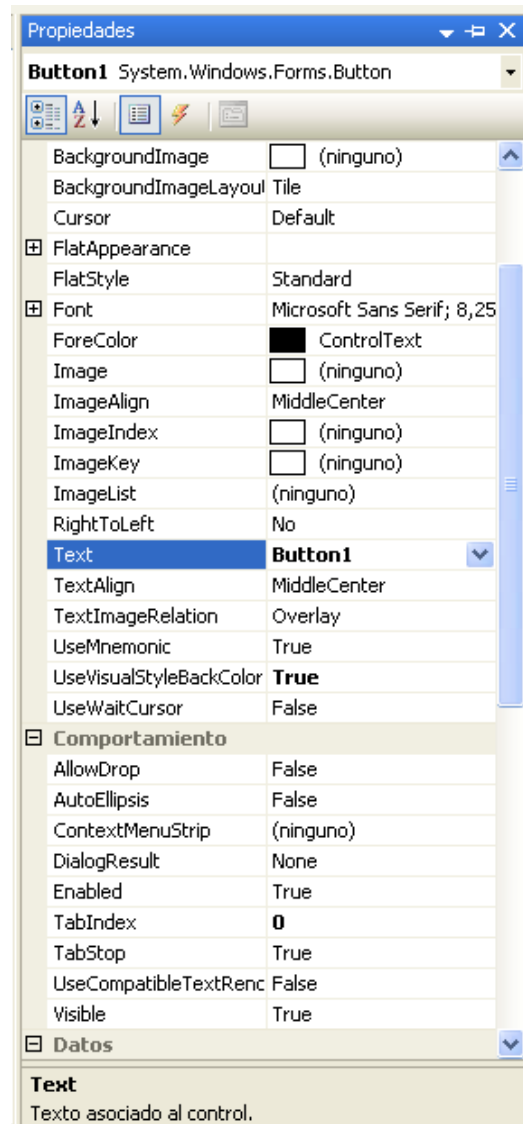


Fig.7. Ventana de propiedades.

Nota: podemos fijar las ventanas, haciendo clic en la “chincheta” de la franja azul superior del cuadro



En la parte superior de la ventana de propiedades vemos una lista desplegable (fig.8), desde ella podemos ver todos los controles que tenemos en el formulario y, al seleccionar uno de ellos accedemos a sus propiedades. Es **muy importante** fijarse en el control que aparece en esa lista, antes de realizar alguna modificación en las propiedades, para cerciorarnos de que estamos editando las propiedades del control que realmente queremos modificar. Esto es motivo muchas veces de equivocaciones.

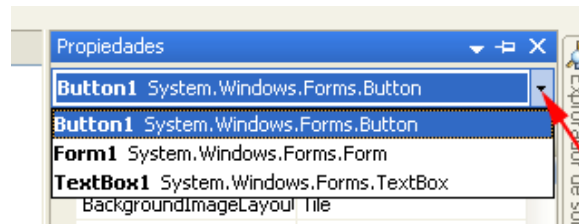


Fig.8. Lista desplegable de la ventana propiedades.

Vamos a modificar la propiedad Text del control Button1. Editamos el valor de la propiedad Text, eliminamos el texto "Button1" y escribimos "Haz clic", cuando pulsemos la tecla Enter o el tabulador se modificará el texto del botón en el formulario (fig.9).

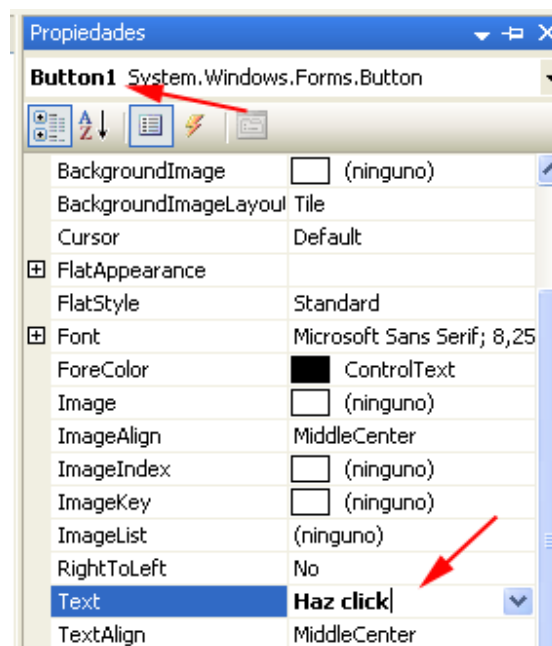


Fig.9. Modificar la propiedad Text del control Button1.

Añadir código a los controles

Vamos introducir código a nuestro ejemplo. Para acceder al código del formulario, hemos de hacerlo de alguna de las siguientes maneras:

- Seleccionar la opción *Ver + Código* en el menú.
- Pulsar F7.
- Hacer clic derecho sobre el formulario y elegir la opción *Ver código* del menú contextual que aparece.

Nos mostrará la siguiente ventana:

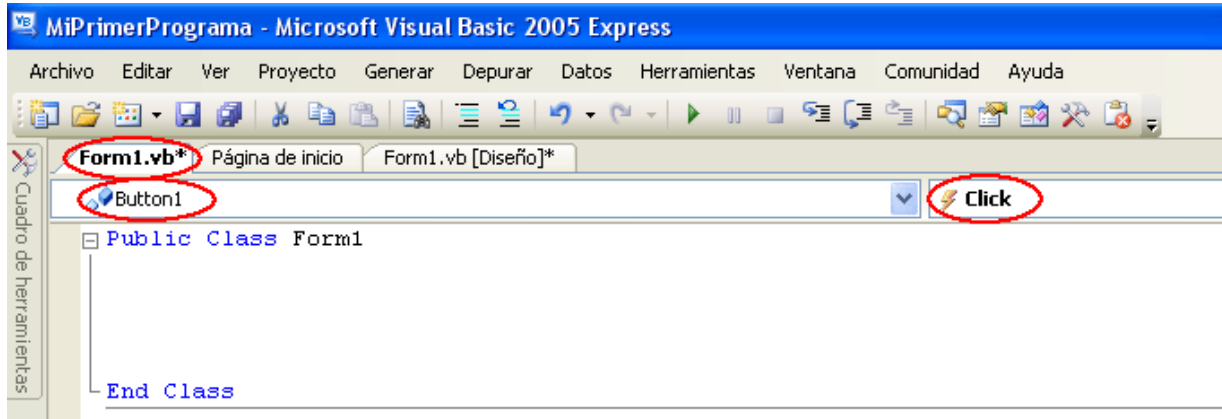


Fig.10. Ventana de código.

Nos aparece una nueva pestaña en el entorno de desarrollo: "Form1.vb", aquí escribiremos el código del formulario "Form1".

Para volver a la ventana que hemos utilizado hasta ahora para diseñar el formulario haremos clic en la pestaña "Form1.vb[Diseño]".

Pero volvamos a la pestaña de código "Form1.vb" y observemos el código que nos muestra:

```
Public Class Form1
```

```
End Class
```

Todo formulario es un objeto y el código de un objeto se escribe en una clase. Por esto, el código lo escribiremos dentro de un bloque **Class... End Class**. "Form1" es el nombre de la clase.

Vamos a introducir código para que al hacer clic en el botón "Button1" (**evento "Click" de "Button1"**) se muestre el texto "¡Hola Mundo!" en el cuadro de texto "TextBox1". La línea de código será:

```
Me.TextBox1.Text = "¡Hola Mundo!"
```

Con esta sentencia conseguimos que al hacer clic en el botón "Button1", la propiedad "Text" del cuadro de texto "TextBox1" reciba el valor "¡Hola Mundo!".

Para escribir el código del evento "Click" correspondiente a la pulsación del control "Button1", la forma más rápida es haciendo doble clic sobre el control en la ventana de diseño del formulario (pestaña "Form1.vb [Diseño]"), lo que abrirá el editor de código y nos situará en el procedimiento vacío "Button1_Click". Escribimos dentro de él la línea de código anterior.

```
Public Class Form1
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

```
        Me.TextBox1.Text = "¡Hola Mundo!"
```

```
    End Sub
```

```
End Class
```

Probemos la aplicación ejecutándola, pulsemos la tecla F5 (o también la opción *Depurar + Iniciar depuración*). Nos aparecerá el formulario en la pantalla y al hacer clic en el botón, aparecerá el texto que escribimos.

Lo que ha ocurrido es que se ha ejecutado el evento "Click" del botón "Button1" tal y como nos lo indica la expresión que vemos al final del encabezamiento del procedimiento.

```
...Handles Button1.Click
```

Esto es lo verdaderamente importante. La palabra clave "Handles", seguida del nombre del control, un punto y el nombre del evento; esto le indica al Framework que debe ejecutar este procedimiento cuando se produzca el evento para ese control. El nombre del procedimiento "Button1_Click" no tiene importancia, se puede modificar y poner cualquier otro; pero normalmente lo dejaremos como está ya que le otorga un nombre significativo.

Ahora vamos a dibujar otro botón en nuestro formulario. Para ello seleccionemos la pestaña "Form1.vb [Diseño]". Vayamos al "cuadro de herramientas" y hacemos doble clic en el control Button. Lo colocamos debajo del botón que ya teníamos. Y modificamos la propiedad "Text" asignándole el literal "Borrar" (fig.11).

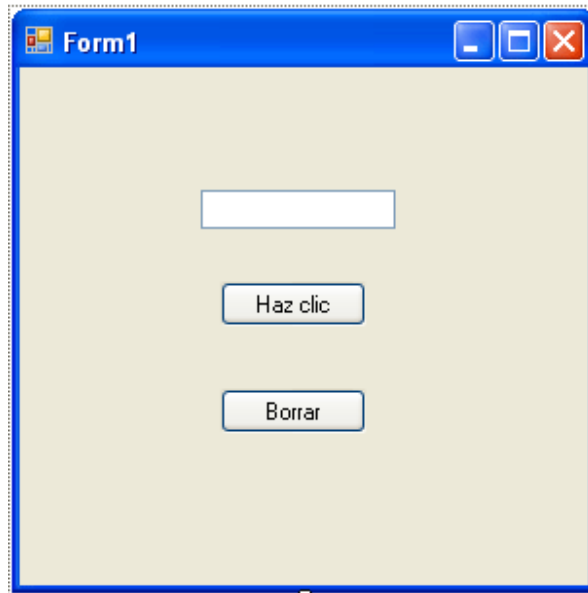


Fig.11. Formulario con el nuevo botón.

Para asignarle el código al nuevo botón, vamos a acceder al procedimiento "Button2.Click" de otra manera. Seleccionamos la pestaña de código: "Form1.vb" y en la lista despegable de la parte superior de la ventana, donde están todos los objetos que tenemos en el formulario, seleccionamos el nuevo botón "Button2" (fig.12)

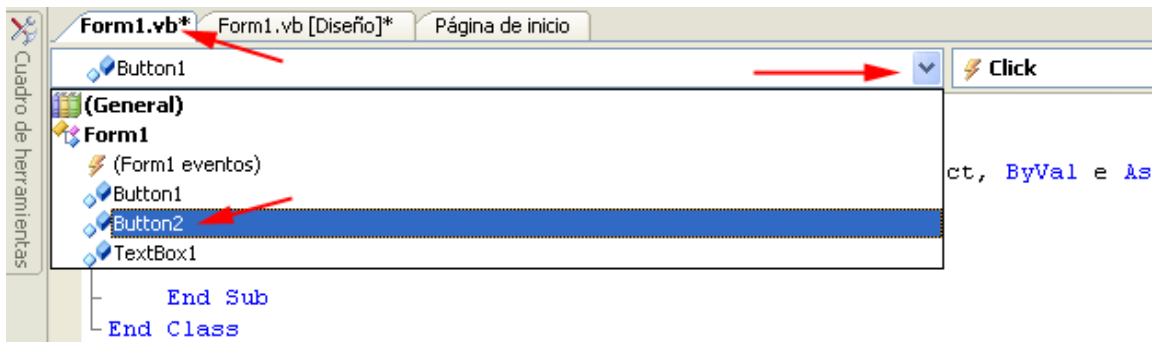


Fig.12. Lista desplegable con los objetos del formulario.

Ahora, de la lista desplegable que está a la derecha de la anterior, seleccionamos el evento que deseamos programar, será el evento "Click" (fig.13).

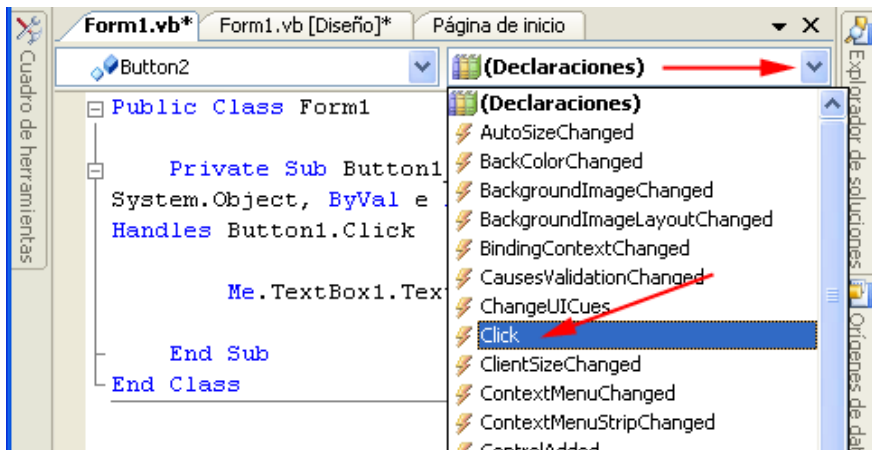


Fig.13. Seleccionar el evento "Click"

Ya tenemos el procedimiento "Button2.Click" para programar el segundo botón de nuestro formulario. Podíamos haber llegado al procedimiento haciendo doble clic en el "Button2" en el diseño del formulario, como hicimos con el "Button1", pero así hemos visto otra forma de acceder a los procedimientos de los objetos.

Añadimos a este procedimiento la siguiente línea de código:

```
Private Sub Button2_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
    Me.TextBox1.Text = Nothing
```

```
End Sub
```

Con lo que conseguiremos "borrar" el contenido de la propiedad "Text" del objeto "TextBox1" al hacer clic en el botón "Button2."



Ejecutamos la aplicación, pulsar F5, y, cuando aparezca el formulario, pulsamos primero en el botón "Haz clic" y después en el botón "Borrar".

Guardar el proyecto

Por defecto, el proyecto se guardará en la carpeta "Mis documentos", dentro de ésta en la ruta: "Visual Studio 2005\Projects\MiPrimerPrograma". Dentro de esta última carpeta tenemos otra carpeta con el mismo nombre y un fichero del tipo "Microsoft Visual Studio Solution". Este es el fichero que hay que hacer doble clic en él para que podamos editar el proyecto con el Visual Basic Express.

Utilización del depurador de proyectos

VB (Visual Basic) tiene un buen depurador, que nos permitirá introducirnos dentro del código de nuestro programa durante la ejecución del mismo, para observar qué es lo que está ocurriendo: ejecutar línea a línea el programa, observar el valor de las variables, etc., aspectos todos ellos fundamentales para el seguimiento de errores y fallos en la lógica de la aplicación.

Vamos a ejecutar nuestro proyecto en modo depuración. Para ello pulsaremos F8, o seleccionaremos en el menú "Depurar + Paso a paso por instrucciones". Cualquiera de estas acciones iniciará la ejecución del programa dentro del contexto del depurador. Nos aparecerá el formulario, hacemos clic en el botón con el texto "Haz clic" y la ejecución se detendrá en la primera línea de código del procedimiento "Button1.Click", destacándola en color amarillo. La línea marcada en amarillo indica que es la siguiente línea a ejecutarse, para ejecutarla y pasar a la siguiente línea pulsaremos de nuevo F8, y así sucesivamente hasta llegar a la última línea del procedimiento (End Sub), ejecutada ésta, se finalizará el procedimiento y nos mostrará de nuevo el formulario. Si hacemos clic en el botón con el texto "Borrar", se realizará el mismo proceso pero con el procedimiento "Button2.Click".

En el proceso de depuración, podemos ver de forma inmediata el valor de una variable o de una propiedad simplemente situando el cursor del ratón sobre ella. En la figura siguiente (fig.13), observamos que la línea siguiente a ejecutarse es "End Sub", por lo que la línea anterior que asignaba el texto "¡Hola Mundo!" a la propiedad "Text" de cuadro de texto "TextBox1" ya se ha ejecutado. Si colocamos el cursor sobre la expresión "`Me.TextBox1.Text`" aparece un recuadro con el contenido de dicha propiedad (fig.14).

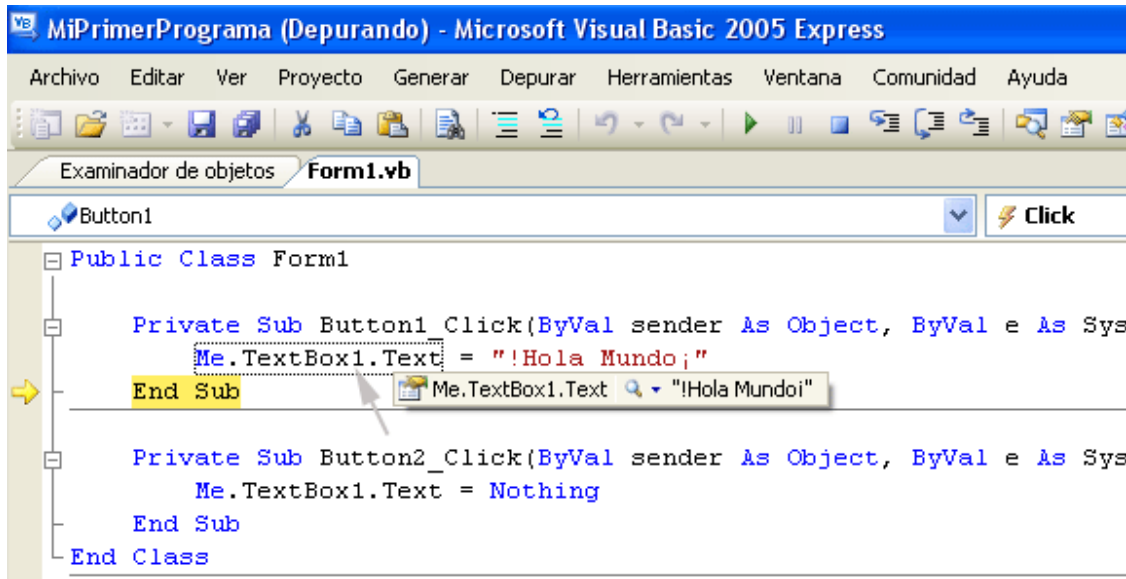


Fig.14. Depurando la aplicación.

Probar a hacer lo mismo pero antes de ejecutar la línea de código que asigna el texto a la propiedad. Vemos que la propiedad Text no tiene ningún contenido.

Si en cualquier momento queremos continuar la ejecución normal del programa sin seguir usando el depurador, pulsaremos F5.

Utilización de la ayuda

La ayuda o MSDN Library (Microsoft Development Network Library) es una importantísima fuente de información de la plataforma NET. La aplicación utilizada para navegar por la documentación de la ayuda es Microsoft Document Explorer, que permite al programador una enorme flexibilidad a la hora de realizar consultas sobre la ayuda disponible de los diferentes productos de desarrollo que componen la plataforma .NET.

Para acceder a ella pulsaremos la tecla F1, tanto si estamos en la pestaña de diseño como en la de código. Aparecerá una nueva entrada en la barra de tareas (fig.15).

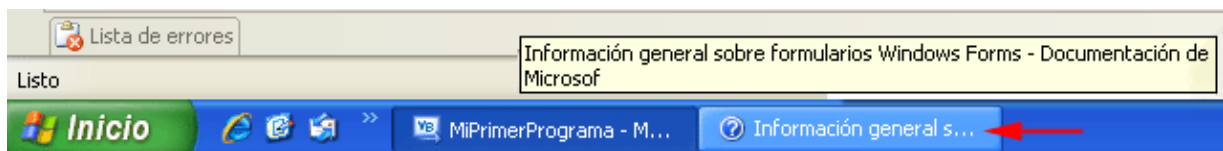


Fig.15. Accediendo a la ayuda.

Como estamos trabajando con el VB, vamos a establecer un "Filtro del lenguaje" para orientar la información que nos proporcione la ayuda a este lenguaje (fig.16). Dejamos sólo seleccionadas las opciones que hacen referencia al Visual Basic.

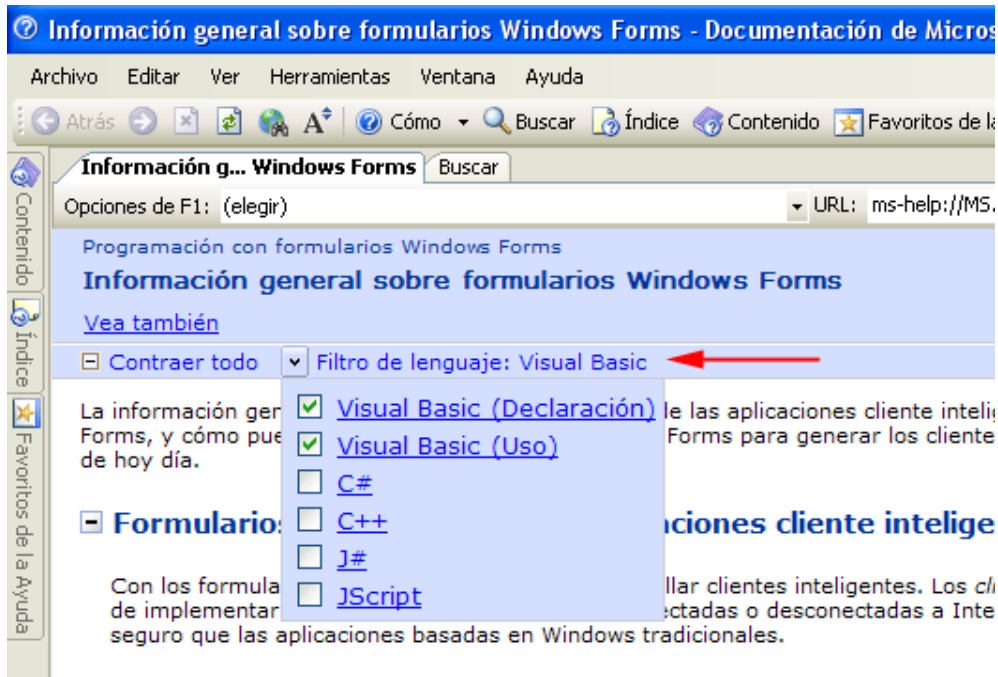


Fig.16. Filtro de lenguaje.

Vamos a realizar una búsqueda en la ayuda para que nos proporcione información de los tipos de datos del lenguaje. Seleccionamos la pestaña “Buscar” y escribimos en el primer cuadro “Resumen tipos de datos”, hacemos clic en el botón “Buscar” de la derecha y esperamos a que realice la búsqueda.

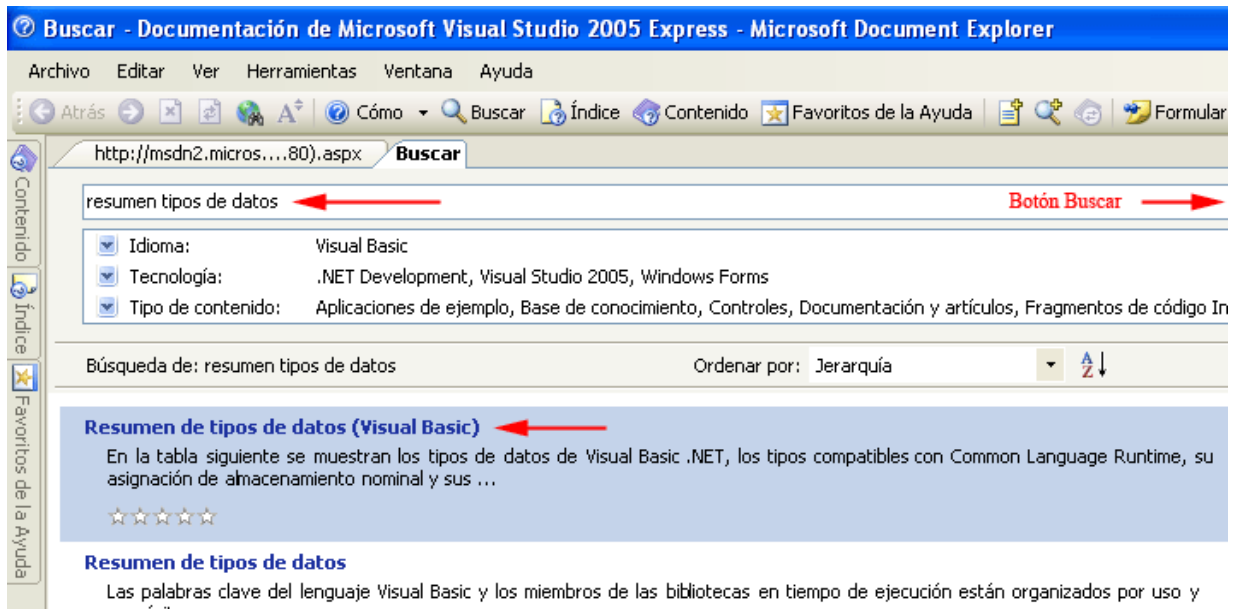


Fig.17. Búsqueda en la ayuda.

Como podemos observar, nos proporciona numerosas entradas que hacen referencia al tema buscado. Tendremos que seleccionar aquellas que se ajusten más a nuestras necesidades de información. Aunque

en este caso la primera entrada nos puede resolver totalmente la consulta, muchas otras veces no es así, hay que realizar una exhaustiva búsqueda para encontrar aquello que buscamos.

Nota: También podemos observar que realiza la búsqueda primero en el MSDN online, en comunidades y foros, y por último en la ayuda local, si quisiéramos que busque primero en la ayuda local, o sólo en esta, podemos realizar la siguiente modificación: en el menú seleccionamos *Herramientas + Opciones* y dentro de esta en *Ayuda + En Pantalla*. Aquí seleccionamos la opción que nos resulte más conveniente en el apartado “Cuando cargue contenido de la ayuda...”

Hacemos clic en la entrada “Resumen de tipos de datos (Visual Basic)” y nos mostrará un cuadro informativo de todos los tipos de datos existentes en este lenguaje. Si la información va a ser consultada con frecuencia, conviene agregarla a “Favoritos de la ayuda”, para ello hacemos clic en el icono de la barra de herramientas (fig.18). Para acceder a los “favoritos de la ayuda” ya guardados, tenemos la pestaña lateral “Favoritos de la ayuda” que está en la izquierda de la ventana (fig.18.).



Fig.18. Agregar a favoritos de la Ayuda.

En la Ayuda encontraremos información de muy distintos niveles. Consultas como “Crear el primer programa en Visual Basic” nos proporciona un conjunto de enlaces donde os explican todo lo contenido en esta unidad. Es sin duda el mejor manual del lenguaje que podéis encontrar para todas las necesidades de información. Os animo a que la consultéis con bastante frecuencia.