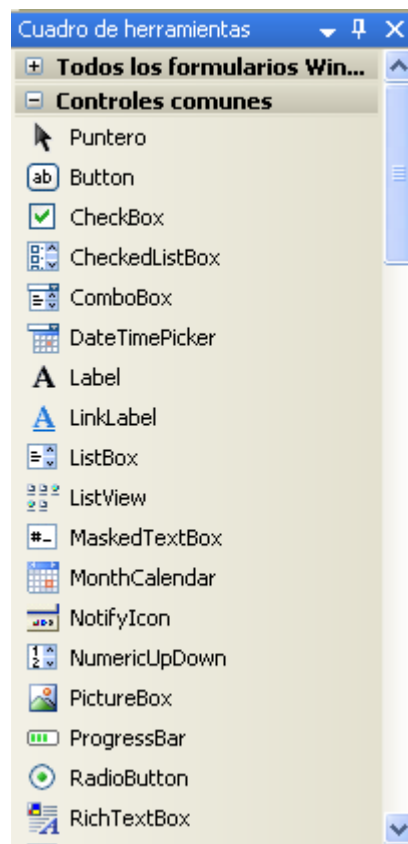




# El IDE de VB .NET. Herramientas y editores

## 1.- El Cuadro de herramientas

Situado habitualmente como una ficha expandible en el lateral izquierdo del IDE, la ventana **Cuadro de herramientas** contiene todos los controles que podemos insertar en un formulario para construir el interfaz de usuario de la aplicación.



## 2.-Organización en fichas

Esta ventana está organizada en base a una serie de fichas, en las que al hacer clic se despliegan sus elementos.

Para seleccionar un control, sólo hemos de desplazarnos por la lista de controles de la ficha que tengamos abierta con las teclas de desplazamiento o los botones de la ventana que realizan también dicha función y que se encuentran en el título de la ficha actual y la siguiente.

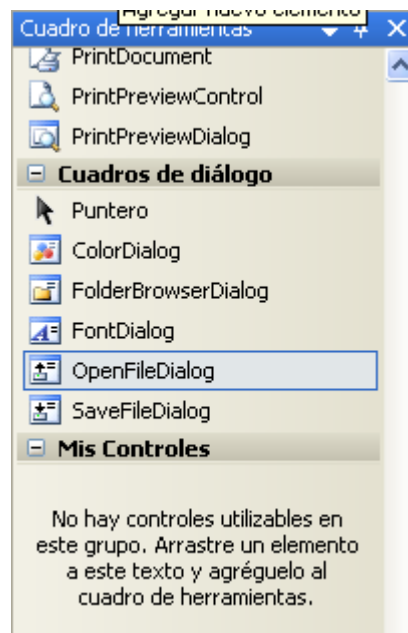


## 3.- Manipulación de fichas

Las fichas del cuadro de herramientas se muestran en un orden predeterminado por el IDE, pero podemos cambiar la posición de cualquier fichas haciendo clic sobre la misma y arrastrando arriba o abajo hasta soltarla en una nueva posición.

Esta flexibilidad también se extiende al hecho de que podemos crear nuestras propias fichas para situar en ella controles ya existentes en el cuadro de herramientas o nuevos controles.

Para crear una nueva fichas, que llamaremos MisControles, haremos clic derecho en el cuadro de herramientas y seleccionaremos la opción de menú **Agregar ficha**, ello creará una nueva ficha vacía en la que escribiremos su nombre y pulsaremos [INTRO] para terminar,.

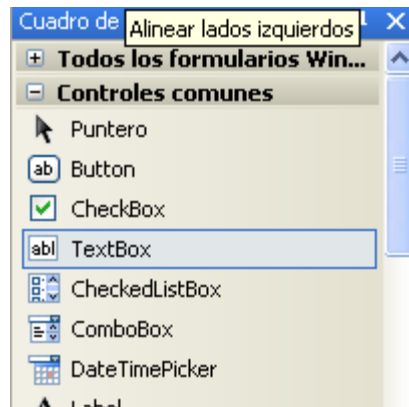


Es posible también, cambiar de nombre y eliminar las fichas existentes, haciendo clic derecho sobre su título y eligiendo del menú contextual, la opción **Eliminar ficha** o **Cambiar nombre de ficha**. Debemos tener en cuenta que sólo podremos eliminar las fichas que hayamos creado nosotros, puesto que las que originalmente se muestran en el IDE, pertenecen al entorno de desarrollo y no pueden eliminarse.

## 4.- Organización de controles

Los controles dentro de una ficha se visualizan por defecto con el icono del control y su nombre, en el llamado modo *Vista de lista*..

Los controles de una ficha vienen ordenados alfabéticamente. El orden alfabético puede tener el inconveniente de situar algunos controles de uso frecuente (por ejemplo TextBox), al final de la lista. Para remediar este problema, podemos cambiar la posición de los controles dentro de la lista de dos maneras: haciendo clic sobre el control y arrastrándolo hasta la posición que deseemos; o bien, haciendo clic derecho sobre la lista de controles y eligiendo las opciones **Subir** o **Bajar**. En este último modo podemos lograr una ordenación mixta, en parte alfabética, pero con nuestros controles de mayor uso en posiciones preferentes.

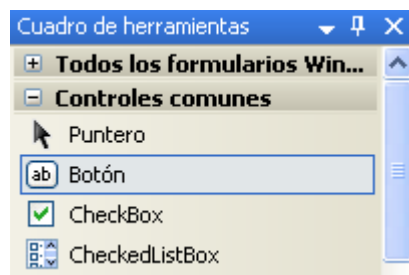


Si se ha modificado demasiado el cuadro de herramientas se puede restaurar con los valores iniciales pulsando el botón derecho del ratón y eligiendo la opción **Restablecer cuadro de herramientas** del menú emergente.

## 5.- Manipulación de controles

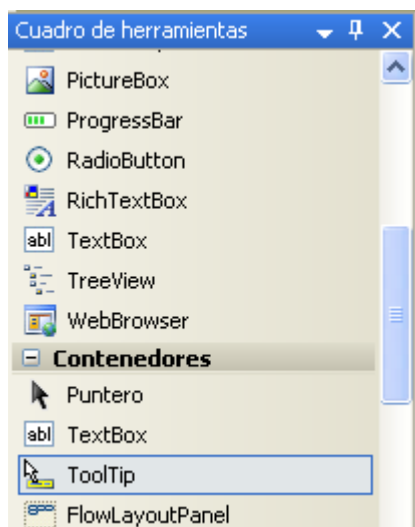
Todos los controles aparecen con su nombre original en el cuadro de herramientas. No obstante, es posible cambiar dicho nombre por otro que sea más fácilmente identificable por nosotros, o con el que nos sintamos más cómodos.

Por ejemplo, supongamos que queremos cambiar el nombre del control **Button**. Para ello, haremos clic derecho sobre dicho control y seleccionaremos la opción **Cambiar nombre de elemento**, lo que nos permitirá editar directamente el nombre en la lista de controles. Introduciremos como nuevo nombre **Botón** y pulsaremos [INTRO].



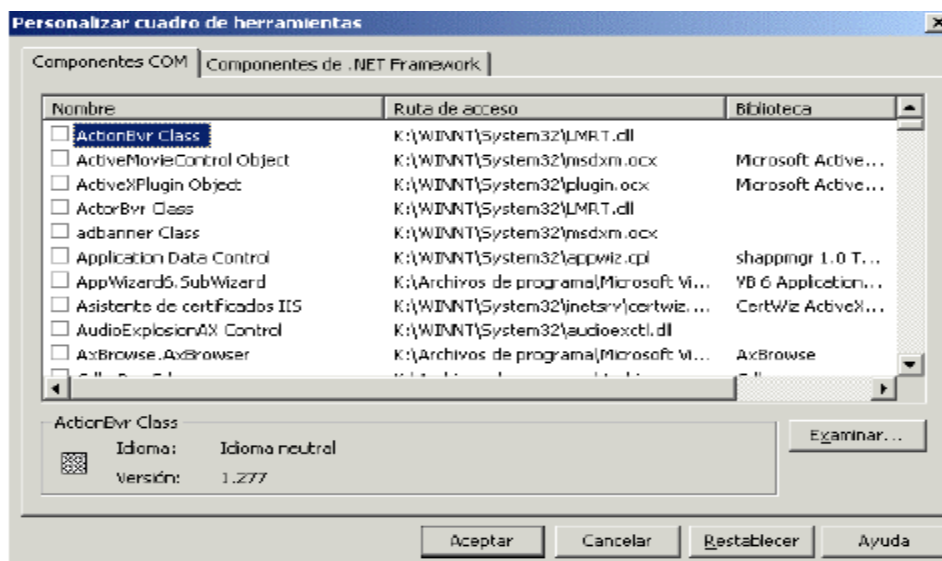
Es importante remarcar, que este tipo de cambio sólo es a nivel del literal mostrado por el control en el cuadro de herramientas, ya que cuando añadamos una copia del control al formulario, el nombre que le asignará el IDE seguirá siendo su nombre original: Button.

Mediante las operaciones estándar de Windows: Cortar, Copiar y Pegar, podemos cambiar de ficha los controles, mientras que si hacemos clic y arrastramos un control hacia otra ficha, moveremos dicho control de su ficha original, depositándolo en una nueva. La siguiente figura muestra la ficha **Contenedores**, con los controles TextBox y ToolTip, originarios de la ficha **Controles comunes**: el primero se ha copiado y el segundo se ha movido a esta ficha.



## 6.- Agregar controles

Además de poder organizar los controles existentes, es posible añadir nuevos controles a cualquiera de las fichas del cuadro de herramientas. Para ello, y una vez situados sobre una ficha, debemos hacer clic derecho y seleccionar la opción de menú **Elegir elementos...**





En esta ventana, y dependiendo de la ficha seleccionada, podemos añadir al cuadro de herramientas tanto controles basados en componentes COM, como componentes de .NET Framework. Como ejemplo, nos situaremos en la ficha General del cuadro de herramientas, abriendo a continuación la ventana para personalizar el cuadro de herramientas. Haremos clic sobre la ficha de componentes de .NET, y marcaremos la casilla del control **DirListBox**, pulsando seguidamente el botón Aceptar, con lo que dicho control se añadirá a la ficha.

La operación inversa, es decir, eliminar un control del cuadro de herramientas, podemos realizarla de dos formas:

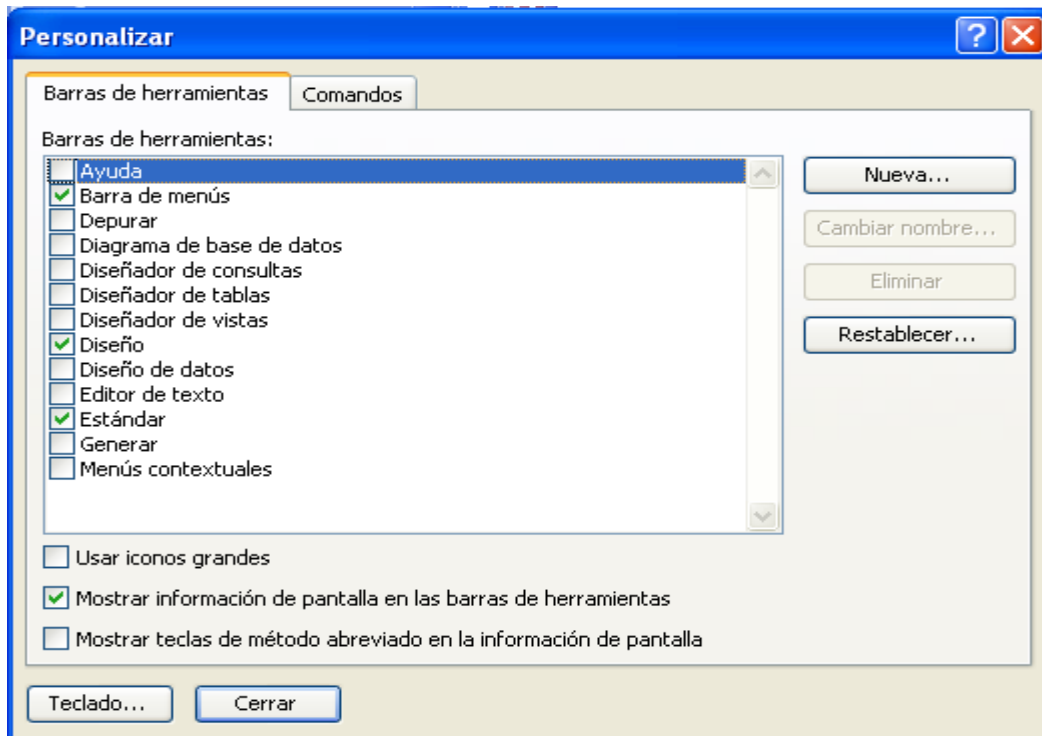
- Seleccionar el control y pulsar [SUPR], confirmando su eliminación del cuadro de herramientas.
- Abrir la ventana para personalizar el cuadro de herramientas, localizar el control y quitar la marca de su casilla. Con ello desaparecerá del cuadro de herramientas.

## 7.- Las barras de herramientas

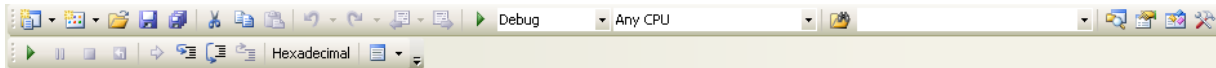
El IDE de VS.NET dispone de un gran número de barras de herramientas, de las cuales sólo se muestran por defecto la estándar. El programador puede posteriormente, visualizar barras adicionales y crear sus propias barras de herramientas personalizadas.

Para mostrar alguna de las barras de herramientas definidas en el IDE, podemos hacerlo de la siguiente forma:

- Opción de menú *Herramientas + Personalizar*, que mostrará la ventana Personalizar, en la que dentro de la ficha *Barras de herramientas*, podremos visualizar y ocultar barras, marcando y desmarcando respectivamente la casilla asociada a dicha barra.

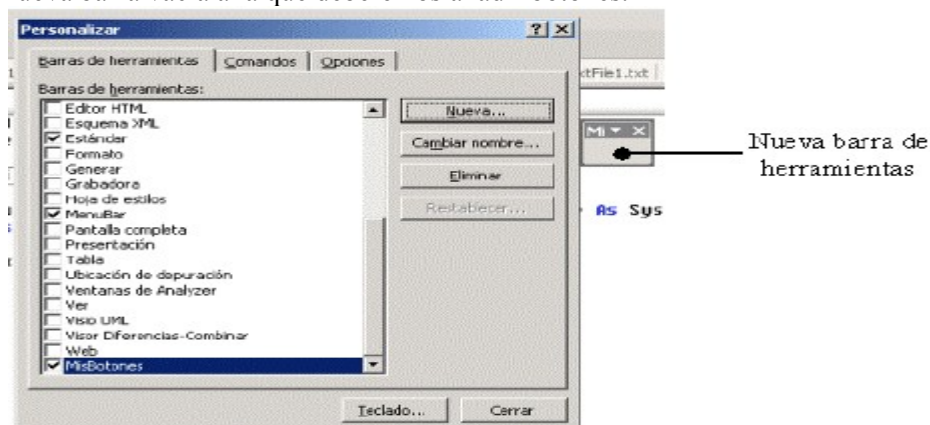


Marcando por ejemplo, la barra *Depurar*, se visualizará esta barra, situándose debajo de la estándar.



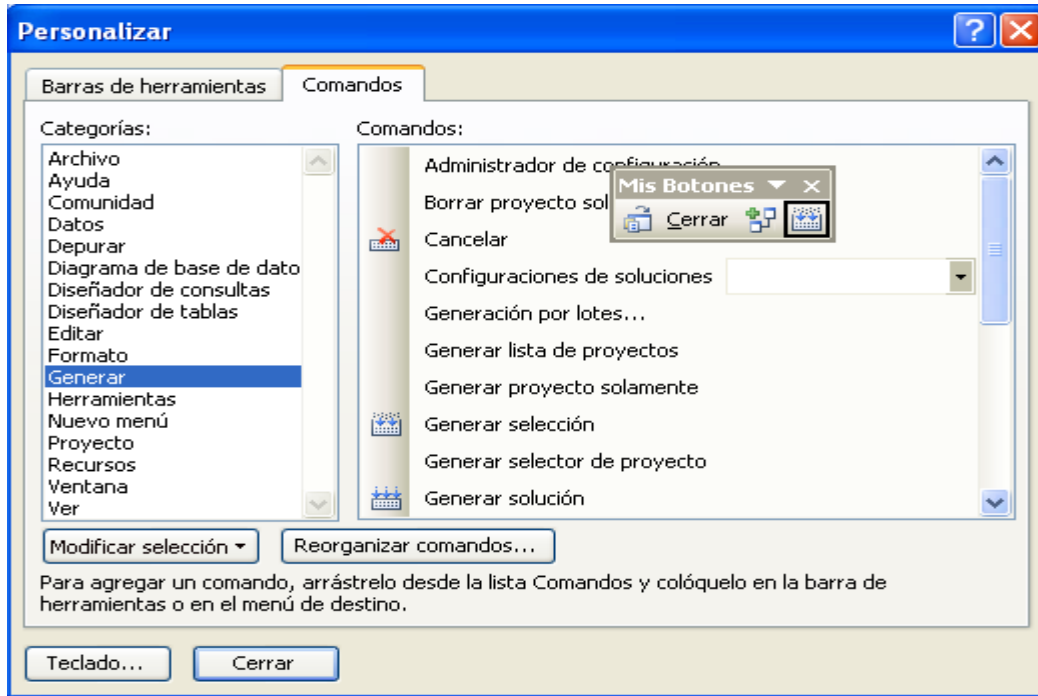
## 8.- Barras de herramientas personalizadas

Durante nuestro trabajo habitual, es muy posible que empleemos con frecuencia acciones situadas en diferentes barras de herramientas. Para no tener una cantidad excesiva y posiblemente innecesaria de barras abiertas al mismo tiempo, podemos crear una barra personalizada (o varias) con nuestro juego de botones favoritos. Los pasos a dar para conseguirlo se describen seguidamente: Abriremos en primer lugar la ventana Personalizar y pulsaremos el botón Nueva, tras lo que deberemos de introducir el nombre de la nueva barra de herramientas, por ejemplo: MisBotones. Esto creará una nueva barra vacía a la que deberemos añadir botones.



El siguiente paso consiste en hacer clic sobre la ficha Comandos de la ventana Personalizar, para seleccionar en el panel izquierdo la categoría de comando a incluir, y en el panel derecho el

comando a insertar. Una vez elegido el comando, lo arrastraremos y soltaremos en nuestra barra de herramientas.



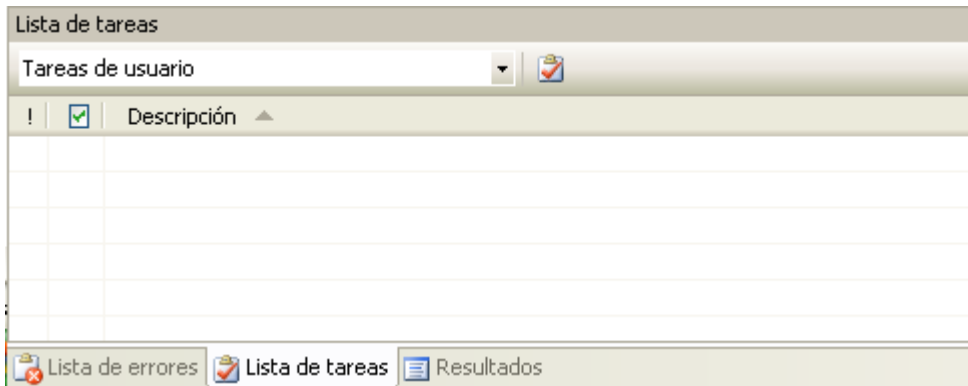
Terminada la confección de nuestra barra de herramientas, pulsaremos el botón Cerrar de la ventana Personalizar, quedando nuestra barra en modo flotante sobre el IDE.

## 9.- Acople de barras de herramientas

Al haber finalizado de crear una barra de herramientas, podemos dejarla flotando sobre cualquier área del entorno de trabajo, o bien, acoplarla debajo de las barras ya visibles o en los laterales del IDE. Esta operación la realizaremos de un modo simple: haciendo clic sobre el título de la barra y desplazándola hasta que quede acoplada en el destino elegido. Podemos cambiar de esta forma, la posición de todas las barras de VS.NET.

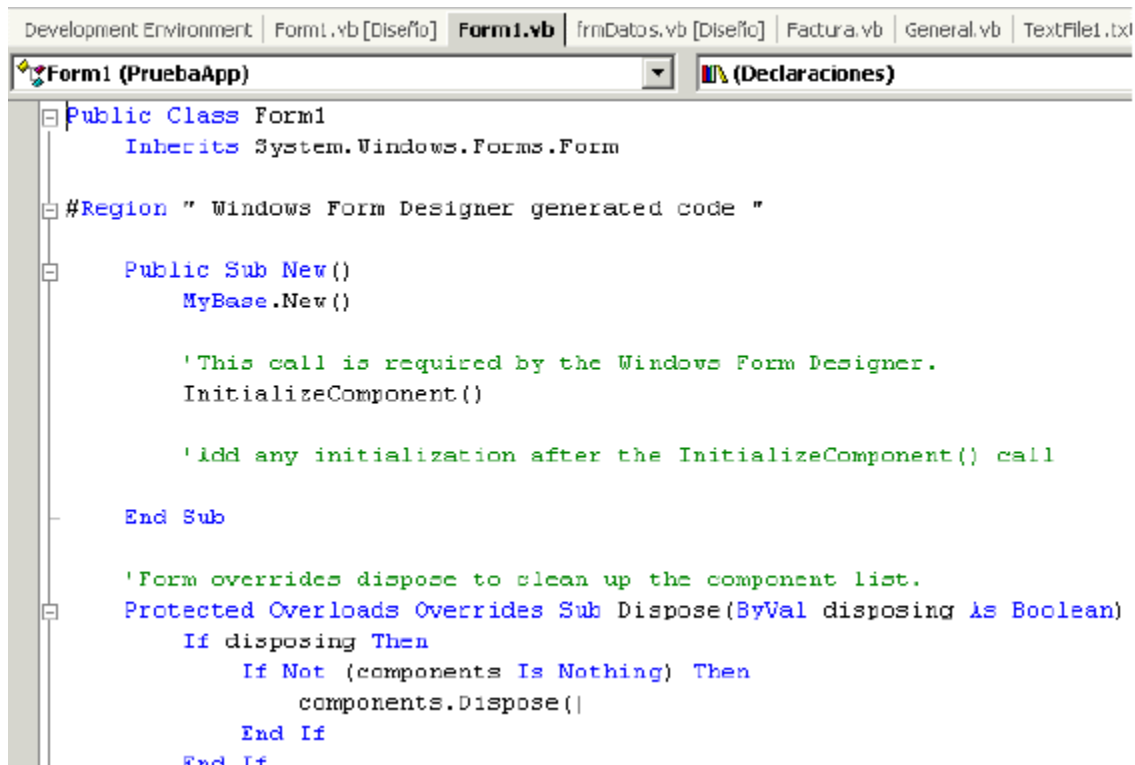
La siguiente figura muestra la barra estándar y la barra personalizada que acabamos de crear en su posición habitual; la barra de edición de código fuente en un lateral del entorno; mientras que la barra para depurar está flotando en el IDE. El indicador con forma de pequeñas líneas situado en la parte izquierda de cada barra, antes del primer botón, sirve para hacer clic sobre él y mover la barra de lugar.





## 13.- El editor de código fuente

En esta ventana escribiremos el código del programa, en forma de declaraciones y procedimientos dentro de clases, módulos, etc.



```

Development Environment | Form1.vb [Diseño] | Form1.vb | frmDatos.vb [Diseño] | Factura.vb | General.vb | TextFile1.txt
Form1 (PruebaApp) | (Declaraciones)
Public Class Form1
    Inherits System.Windows.Forms.Form

    #Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
    End Sub
End Class

```

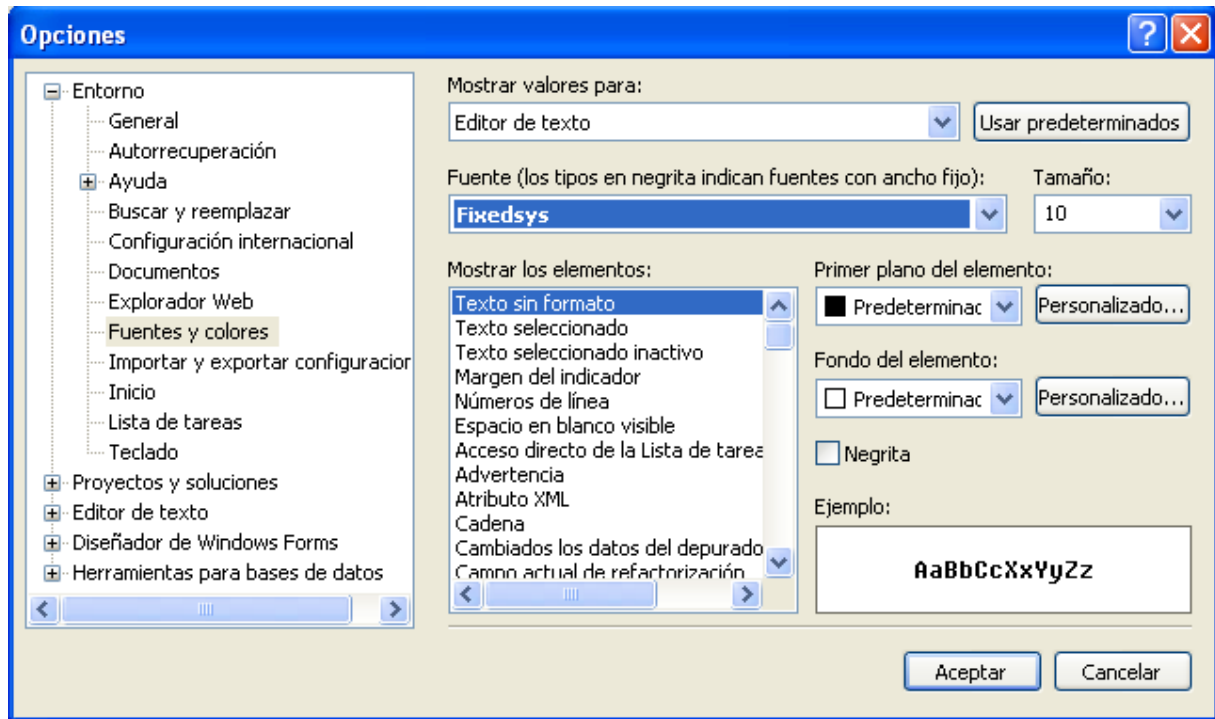
Esta ventana dispone de multitud de opciones y características dada su importancia, por lo que en este apartado trataremos las más importantes.

### Ajuste de fuente y color

En cuanto a tipo de letra y color es posible modificarlos si queremos realizar una configuración más personalizada.

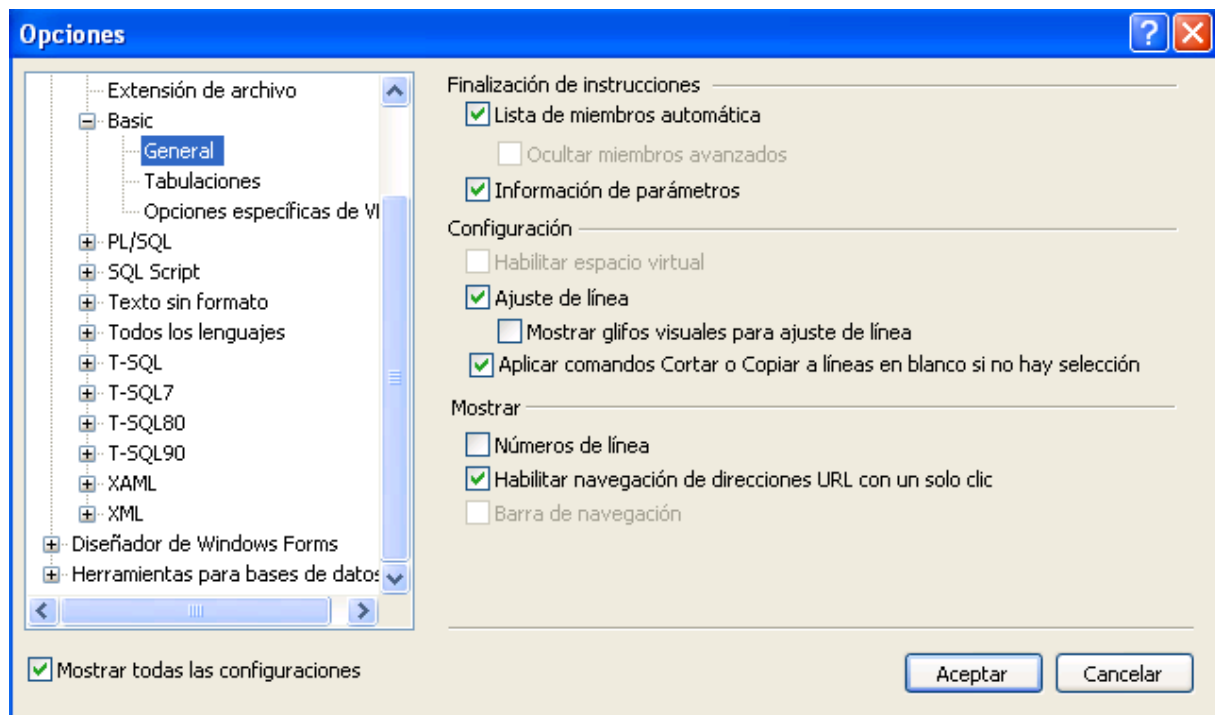
Por ejemplo, si el tipo de letra no es de nuestro agrado, seleccionaremos la opción **Herramientas + Opciones**, que abrirá la ventana Opciones. Dentro de ella, en la carpeta Entorno,

haremos clic en el elemento **Fuentes y colores**, que nos mostrará en la parte derecha la configuración de tipo de letra y colores para el editor de código fuente.



Aquí podemos cambiar por ejemplo, el tipo de fuente a una de ancho fijo como Fixedsys, más cómoda para trabajar, y el color de algunos elementos de código, como los literales de error, comentarios, etc.

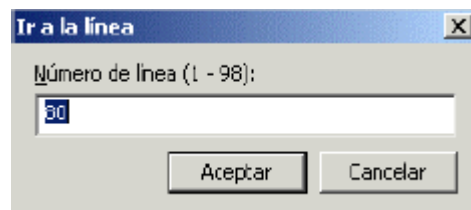
Aparte de estas opciones, la carpeta **Editor de texto** de esta misma ventana, nos permite configurar tanto aspectos generales del editor de código, como particulares para cada lenguaje.



Entre las características del editor para VB.NET que podemos configurar, se encuentra el mostrar la lista de miembros automáticamente para los objetos, visualizar los números de línea, indentación de código, ancho de los tabuladores, finalización automática de estructuras de código, etc.

## Números de línea

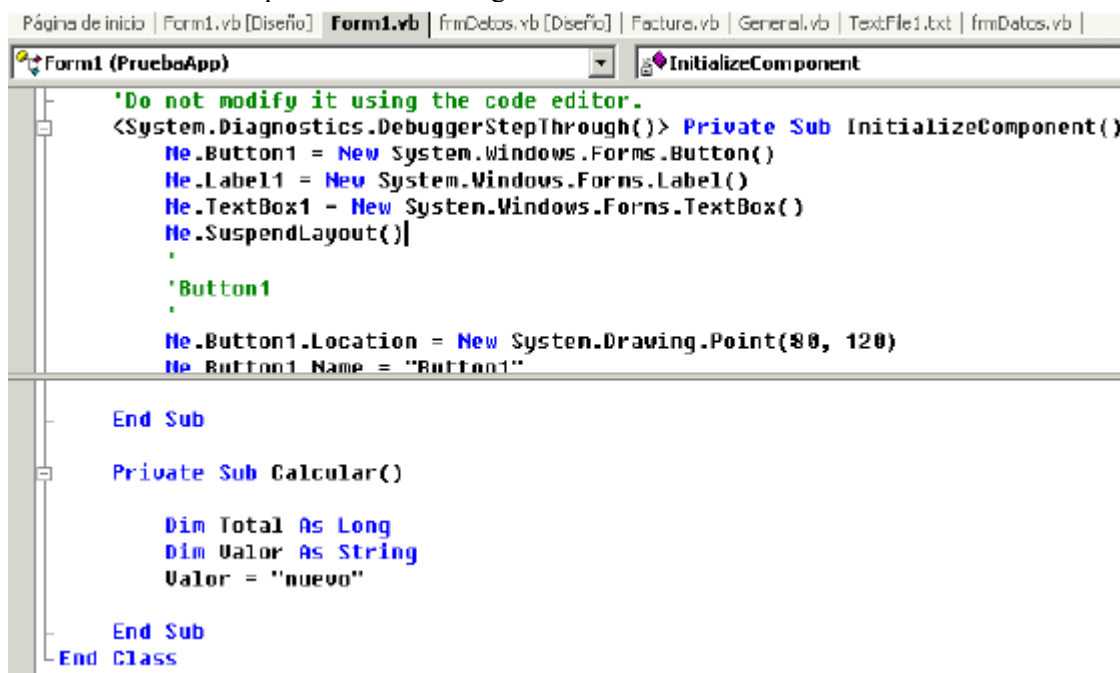
En la parte inferior derecha del IDE nos encontramos la barra de estado, que muestra el número de línea, columna y carácter sobre el que estamos situados actualmente. Si en algún momento necesitamos desplazarnos a un número de línea determinado dentro del documento actual, seleccionaremos la opción de menú **Editar + Ir a**, y se indicará la línea en la que deseamos posicionarnos.



## Dividir el editor de código

Podemos encontrarnos en una situación en la que es muy importante visualizar en todo momento parte de un fragmento de código, pero la edición debemos realizarla en otro punto del documento diferente, por ejemplo: visualizar las variables de un procedimiento, que hemos declarado en su cabecera, pero editar la parte final del procedimiento.

Para solucionar este problema, podemos dividir horizontalmente la ventana del editor de código mediante la opción de menú **Ventana + Dividir**. De esta manera es posible disponer de dos vistas totalmente independientes del código.



Para dejar esta ventana con un solo panel de edición seleccionaremos la opción de menú **Ventana + Quitar división**.

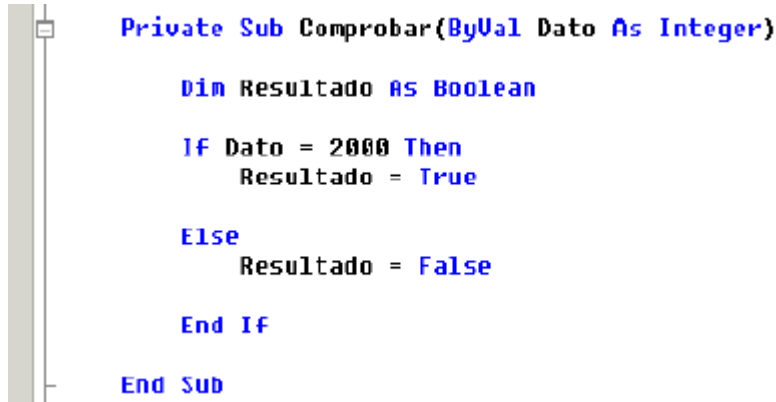
Otro modo de establecer la división del editor de código en dos paneles, consiste en hacer clic y arrastrar el indicador de división que se encuentra en la parte superior derecha de esta ventana, soltando en el punto que deseemos. Con esto quedará también dividida la ventana.



## Esquematización

La Esquematización u Outlining consiste en una característica del editor por la cual podemos expandir o contraer bloques de código, facilitando su lectura.

Cuando la esquematización se encuentra activada (estado por defecto), se muestra una línea o guía en el lateral izquierdo del editor, que discurre paralela al código.

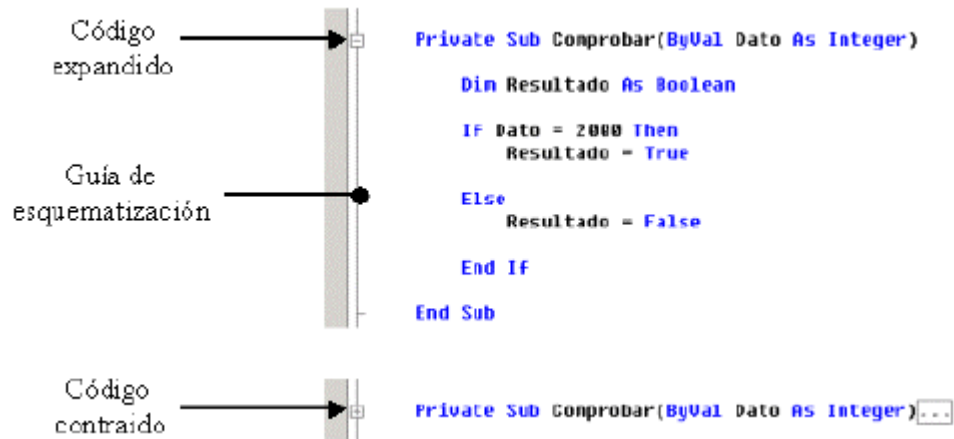


```

Private Sub Comprobar(ByVal Dato As Integer)
    Dim Resultado As Boolean
    If Dato = 2000 Then
        Resultado = True
    Else
        Resultado = False
    End If
End Sub

```

La guía dispone a lo largo del código de una serie de símbolos más (+), y menos (-), que permiten al hacer clic, expandir o contraer respectivamente el código. Cuando hay una parte del código oculto de esta manera, se muestran además unos puntos suspensivos.

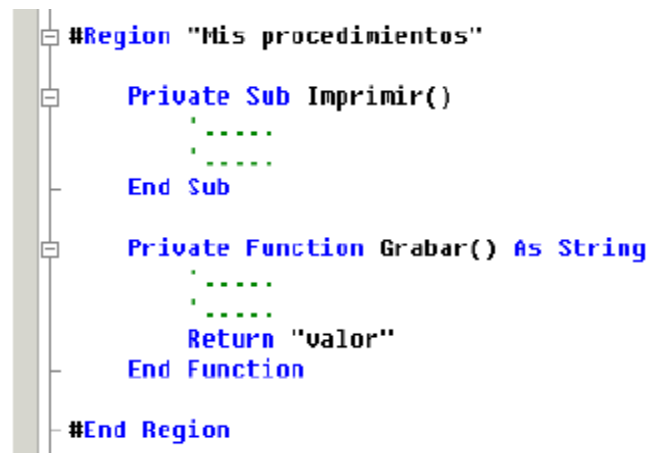


Podemos manipular la esquematización a través de las opciones de menú contenidas en **Editar + Esquematización**, que describimos a continuación de forma breve:

- **Alternar expansión de esquematización.** [CTRL + M, CTRL + M]. Expande o contrae el procedimiento de código en el que estamos situados.
- **Alternar toda la esquematización.** [CTRL + M, CTRL + L]. Expande o contrae todo el código existente en el editor.
- **Detener esquematización.** [CTRL + M, CTRL + P]. Elimina la esquematización del documento actual, suprimiendo la guía.
- **Contraer a definiciones.** [CTRL + M, CTRL + O]. Contrae cada procedimiento existente en el editor.

## Regiones

Mediante las directivas del compilador **#Region...#End Region**, podemos definir bloques de código, que comprendan varios procedimientos, y que a través de esquematización, ocultemos o expandamos colectivamente. Cada región de código debe tener un nombre, que estableceremos en una cadena de caracteres a continuación de la palabra clave Region.



Al crear un formulario con el diseñador de VS.NET, su editor de código correspondiente añade una región con el nombre "Windows Form Designer generated code", que contiene el código que genera automáticamente el diseñador.

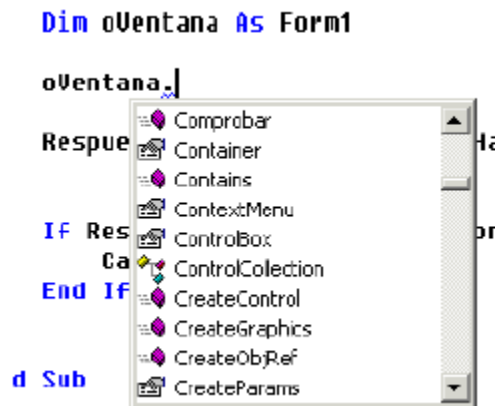
## IntelliSense

Las opciones del menú **Editar + IntelliSense**, suponen una gran ayuda para el programador a la hora de escribir código, facilitando la labor de buscar miembros de clases, informando sobre los parámetros de procedimientos, etc.

A continuación realizaremos un rápido repaso de cada una de estas opciones:

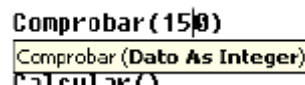
- **Lista de miembros.** [CTRL + J]. Abre una lista que muestra los miembros del identificador seleccionado en el editor: clase, enumeración, etc.

Nos encontramos en la siguiente situación: hemos creado una aplicación Windows con el típico formulario que se encuentra en una clase con el nombre Form1. En un procedimiento declaramos una variable con el nombre oVentana, de tipo Form1. Cuando escribamos esta variable y pongamos un punto, o bien ejecutemos esta opción de menú, aparecerá la lista de miembros de la clase Form1.



- **Información de parámetros.** [CTRL + MAYÚS + I]. Al hacer una llamada a un procedimiento, esta opción muestra una viñeta informativa con un resumen de la lista de parámetros necesarios para un procedimiento del programa.

En el siguiente ejemplo realizamos una llamada a un procedimiento llamado Comprobar( ), que recibe como parámetro un valor numérico Integer.



La definición de este procedimiento se muestra en el siguiente código fuente.

```
Private Sub Comprobar(ByVal Dato As Integer)

    Dim Resultado As Boolean

    If Dato = 2000 Then
        Resultado = True
    Else
        Resultado = False
    End If

End Sub
```

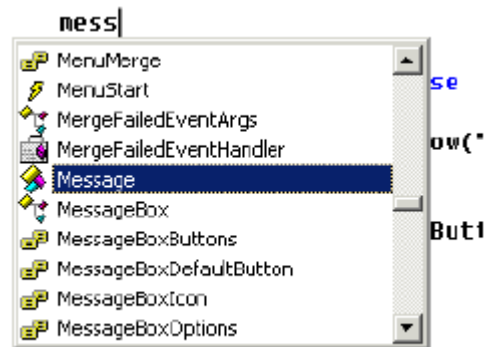
- **Información rápida.** [CTRL + I]. En función del elemento de código sobre el que sea aplicada, esta opción muestra una viñeta informativa.

Si por ejemplo, ejecutamos esta opción sobre la llamada al método **Show()** de un objeto **MessageBox**, se muestra la información de la Figura 134. Esto indica que el método requiere un parámetro de tipo **String**.

**Respuesta - MessageBox.Show("Has pulsado un botón")**

**String**

- **Palabra completa.** [CTRL + ESPACIO]. Ayuda a completar la escritura de una instrucción. Si por ejemplo, comenzamos a escribir en el editor la palabra **mess** y seleccionamos esta opción, se abrirá una lista de valores en el que se mostrará el valor más parecido. En esta lista podremos ya elegir la instrucción que necesitamos.



- **Insertar fragmento de código.** [CTRL +K, CTRL+X]. Ayuda a construir instrucciones.

