

El IDE, estilos, temas y páginas maestras

Indice

Nº 10 - El IDE, estilos, temas y páginas maestras.....	1
1. El entorno de trabajo.....	1
1.1 Paneles.....	1
1.2 Ayuda.....	4
2. Editar páginas.....	7
Editar páginas.....	10
2.1 Mas vistas.....	18
3. Crear páginas.....	24
3.1 Crear tablas HTML.....	25
3.2 Añadir enlaces.....	28
3.3 Imágenes.....	31
3.4 Editar el código fuente.....	33
4. Diseñar con estilos.....	34
Tipos de estilos.....	34
4.1 Los ficheros CSS.....	35
4.2 El generador de estilos.....	36
5. Los temas.....	46
5.1 Crear temas.....	46
5.2 Utilizar temas en las páginas.....	52
6. Páginas maestras.....	57
6.1 Una página maestra sencilla con una página de contenidos.....	58
6.2 Página maestra con varias regiones.....	62
6.3 Páginas maestras y rutas relativas.....	64
7. Páginas maestras avanzadas.....	65
7.1 Esquemas basados en tablas.....	65
7.2 Código en la página maestra.....	67
Ejercicios.....	71
Ejercicio 1.....	71
Ejercicio 2.....	73

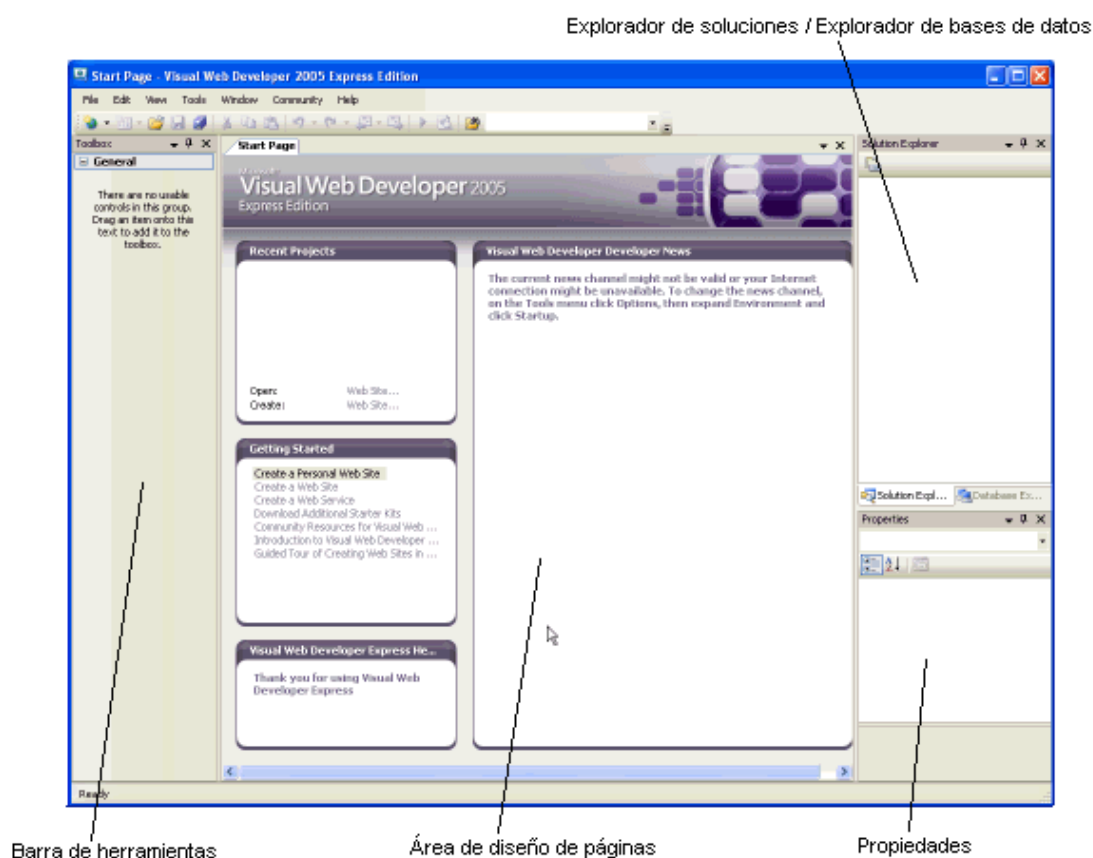
Nº 10 - El IDE, estilos, temas y páginas maestras.

1. El entorno de trabajo

Aunque a estas alturas ya deberíamos haberlo conocido nos vamos a meter un poco ahora con el entorno de desarrollo. Para ver una descripción de las partes que tenemos, mas que nada porque hasta hemos estado preocupados en aprender la teoría y a partir de ahora vamos a trabajar mas estrechamente con el IDE, aprovechando todas sus ventajas para la creación de páginas. Empecemos por recordad los paneles...

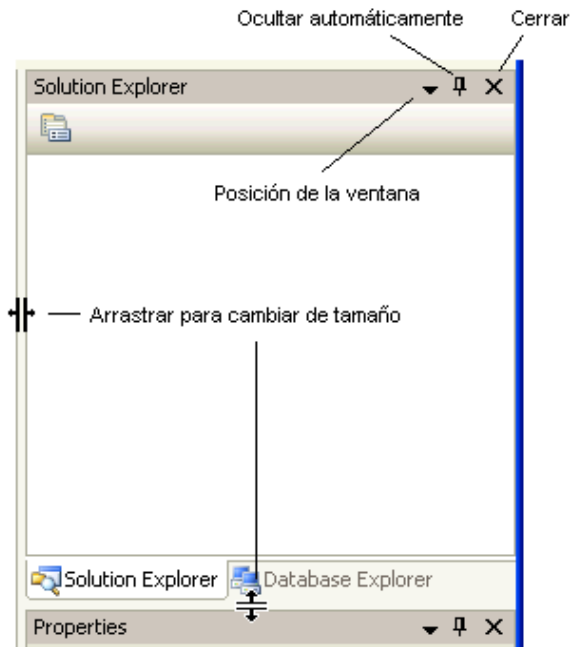
1.1 Paneles

Veamos ahora las partes en las que se compone nuestro entorno.



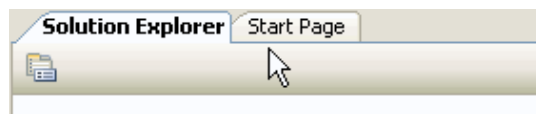
Podemos mover, cambiar de tamaño, mostrar y ocultar los paneles que he señalado en el gráfico anterior. Estas son las acciones que podemos realizar con los paneles:

El IDE, estilos, temas y páginas maestras



El botón de "Posición de ventana" permite varias opciones:

- Flotante: convierte el panel en una ventana que se puede colocar libremente por el escritorio
- Acoplable: Se acopla como una solapa
- Documento solapado. Se convierte en una solapa de la pantalla principal:



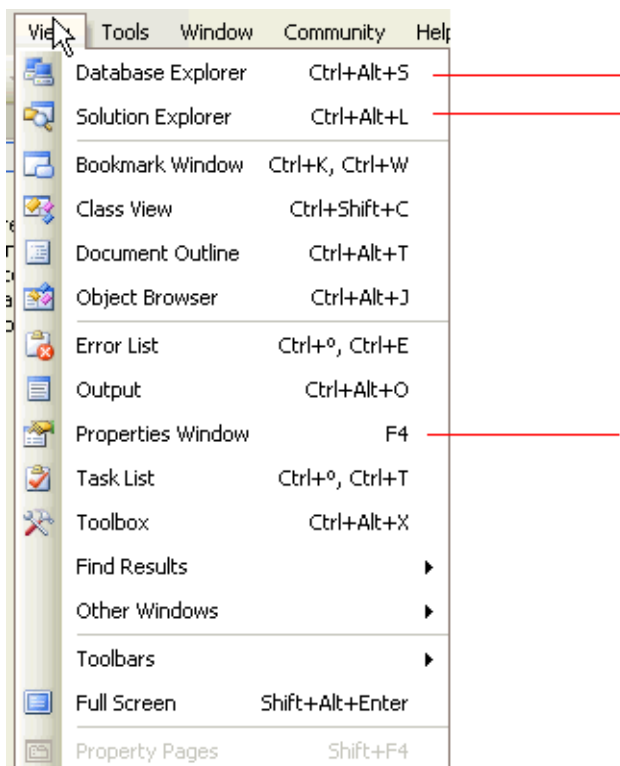
- Ocultar automáticamente. Se oculta en una pestaña a la derecha de la ventana y se mostrará al pasar el ratón sobre ésta.
- Ocultar. Oculta el panel.

Por ejemplo si elegimos que se oculten automáticamente estos tres paneles quedarían de esta forma:

El IDE, estilos, temas y páginas maestras

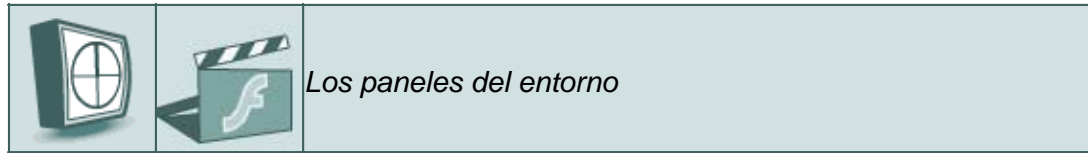
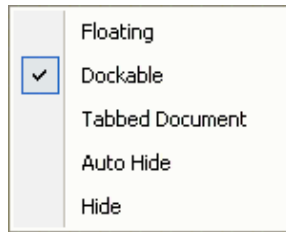


Al pasar el ratón se mostrarían. Con todas estas opciones es normal que a veces perdamos algún panel así que tenemos la opción de menú View:



Te he marcado en rojo los paneles mas habituales que son los que aparecen por defecto en nuestro entorno de desarrollo. Por último decirte que al pulsar con el botón derecho en la barra de títulos de los paneles podemos ver estas opciones de visualización:

El IDE, estilos, temas y páginas maestras



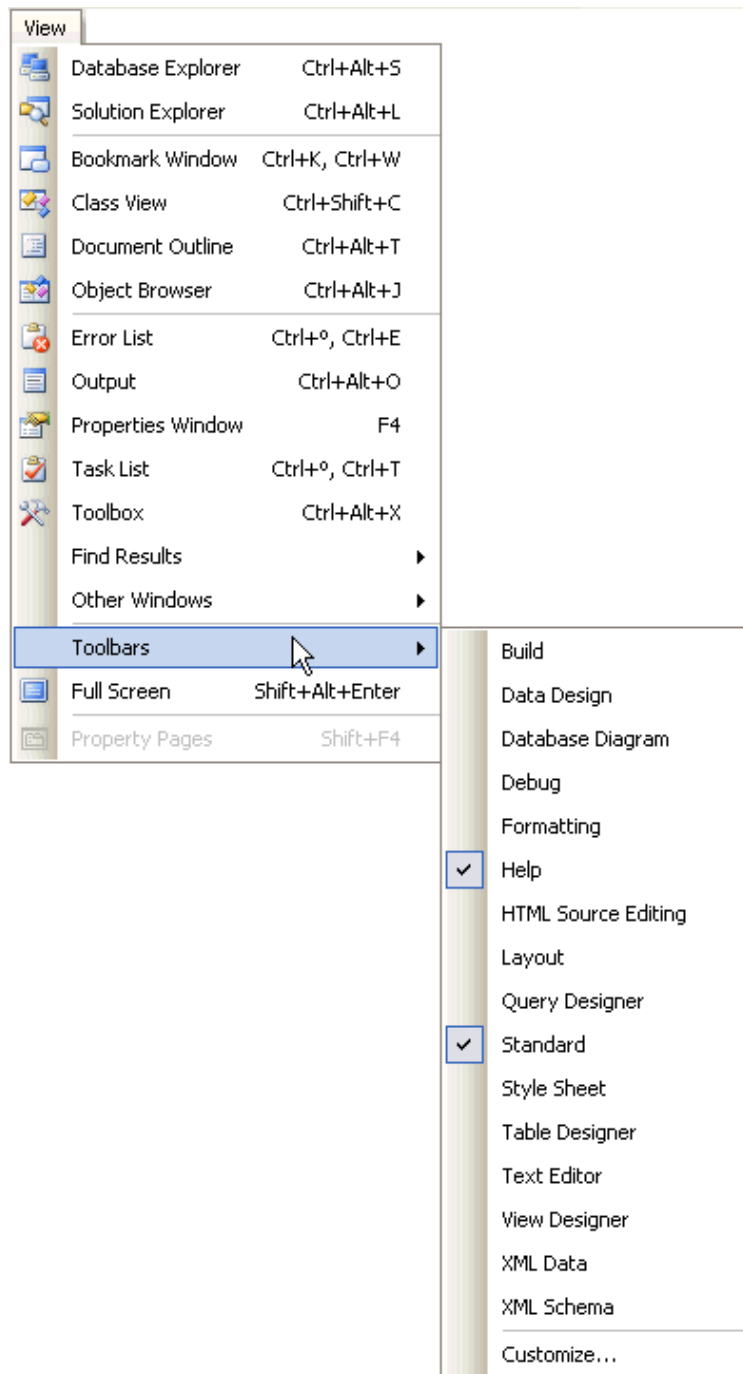
1.2 Ayuda

La ayuda, aunque con el inconveniente del inglés, es un elemento muy importante en nuestro desarrollo. Antes con la primera versión de ASP, los objetos y el lenguaje eran tan limitados que apenas disponíamos de unas páginas de ayuda como páginas Web en el propio IIS.

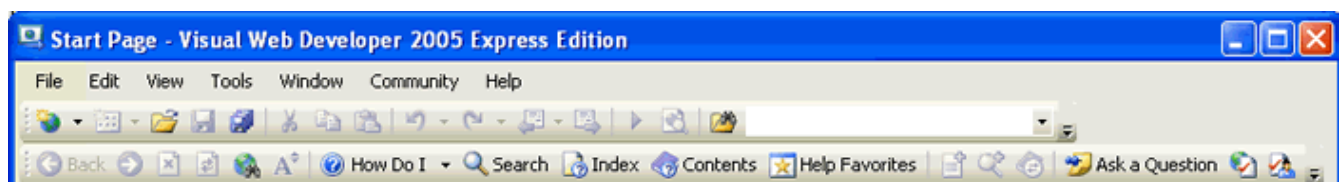
Si has conocido alguna vez un entorno de desarrollo de Microsoft habrás visto que la ayuda es una de las principales ventanas en el IDE o entorno integrado de desarrollo. Si no lo conoces no pasa nada, lo vemos ahora.

Para acceder a la ayuda dejaremos permanentemente activada la barra de ayuda. Para activar esta barra podemos hacerlo de dos formas, una es con el botón derecho en las barras de botones y eligiendo "Help" o desde el menú "View" y luego "Toolbars" que en ambos casos nos muestra las barras de herramientas.

El IDE, estilos, temas y páginas maestras



Para que nos muestre la barra:



Las opciones que tenemos son las siguientes:

El IDE, estilos, temas y páginas maestras

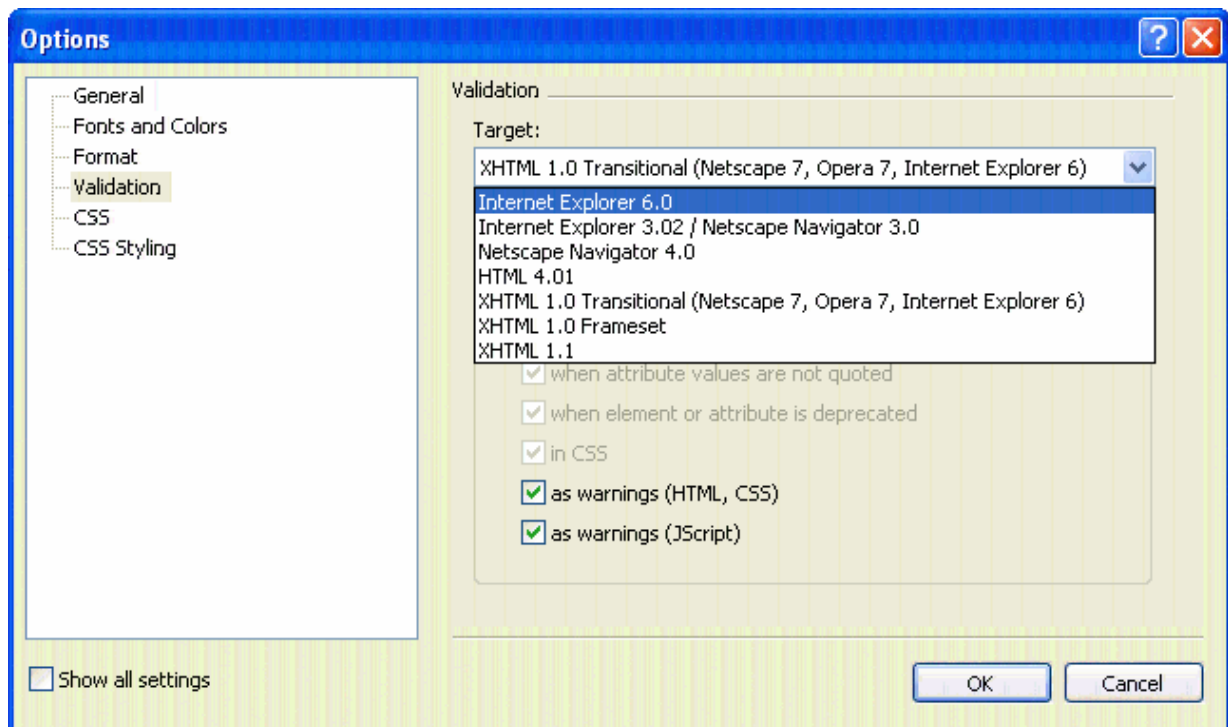
- "How Do it". Contiene enlaces para realizar tareas comunes en el entorno de desarrollo.
- "Search". Pantalla de búsqueda con varias opciones
- "Index". Índice alfabético
- "Contents". Organiza la ayuda por temas
- "Help Favorites". Abre la ventana de enlaces favoritos que podemos ir recopilando cuando trabajemos.

Y alguna opción mas que puedes explorar, pero las principales son estas.

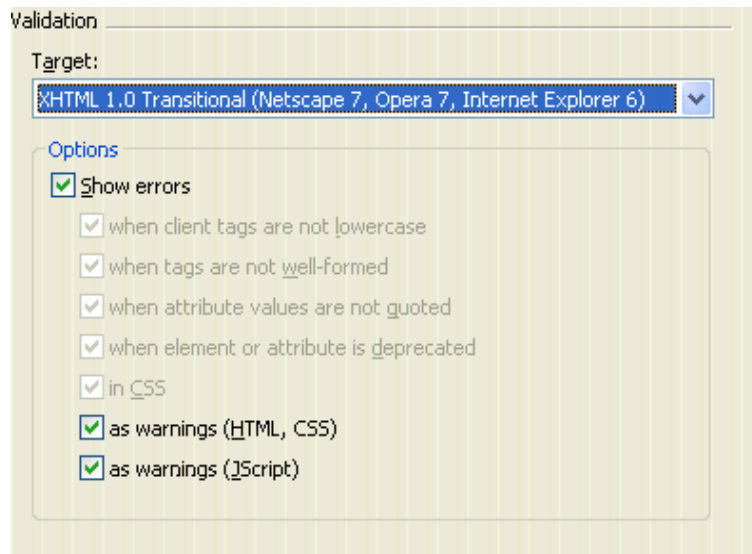
Compatibilidad con exploradores Web

Cuando se desarrollan páginas Web muchas veces no pensamos en el navegador o explorador que tendrá el cliente. Pensamos que la página le funcionará sin problemas y no reparamos en posibles incompatibilidades.

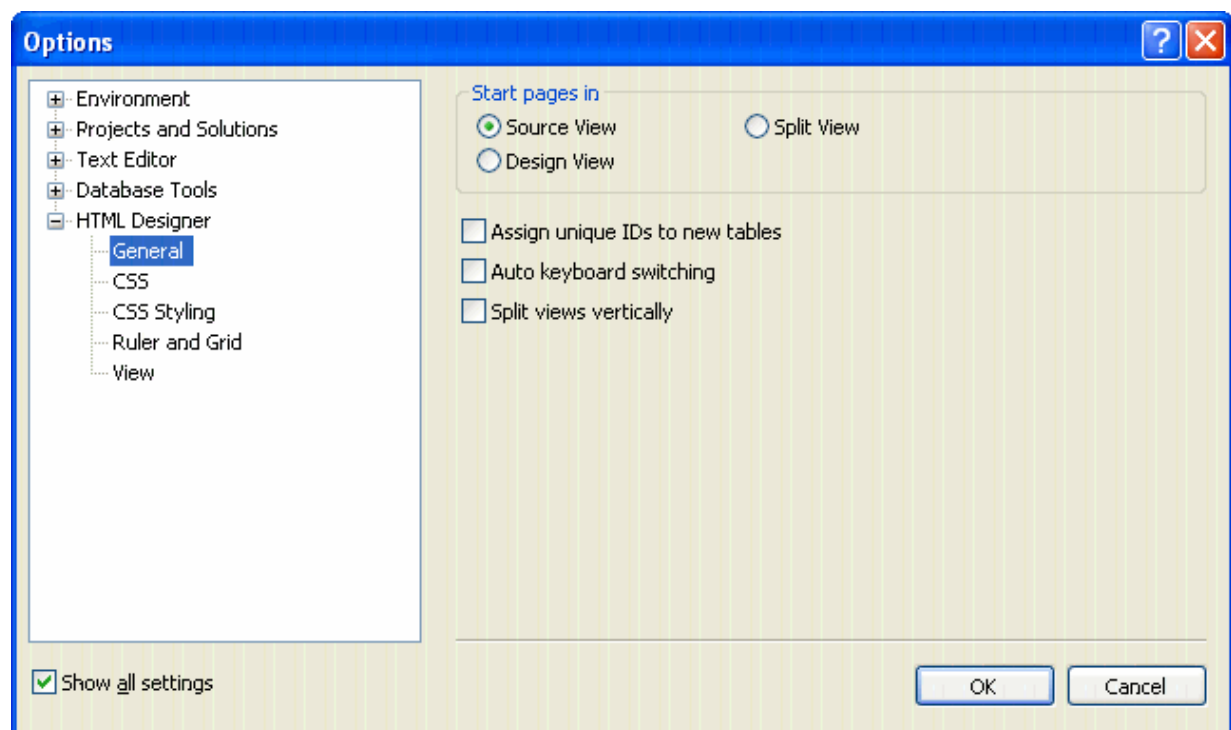
Para elegir el tipo de explorador al que va destinado nuestro sitio Web nos iremos al menú "Tools" y luego a "Options":



Como ves, he seleccionado la opción "Validation" y en la parte derecha "Target" podemos elegir este tipo de navegador. Esta opción de menú se llama "Validation" porque nuestro entorno comprobará automáticamente las páginas que genera para que sea compatible con lo que hemos seleccionado. Elegiremos la opción:



Bien, por último y antes de comenzar a diseñar páginas haz clic en la parte inferior izquierda "Show all settings" para que nos muestre todas las opciones disponibles y por fin en esta opción:

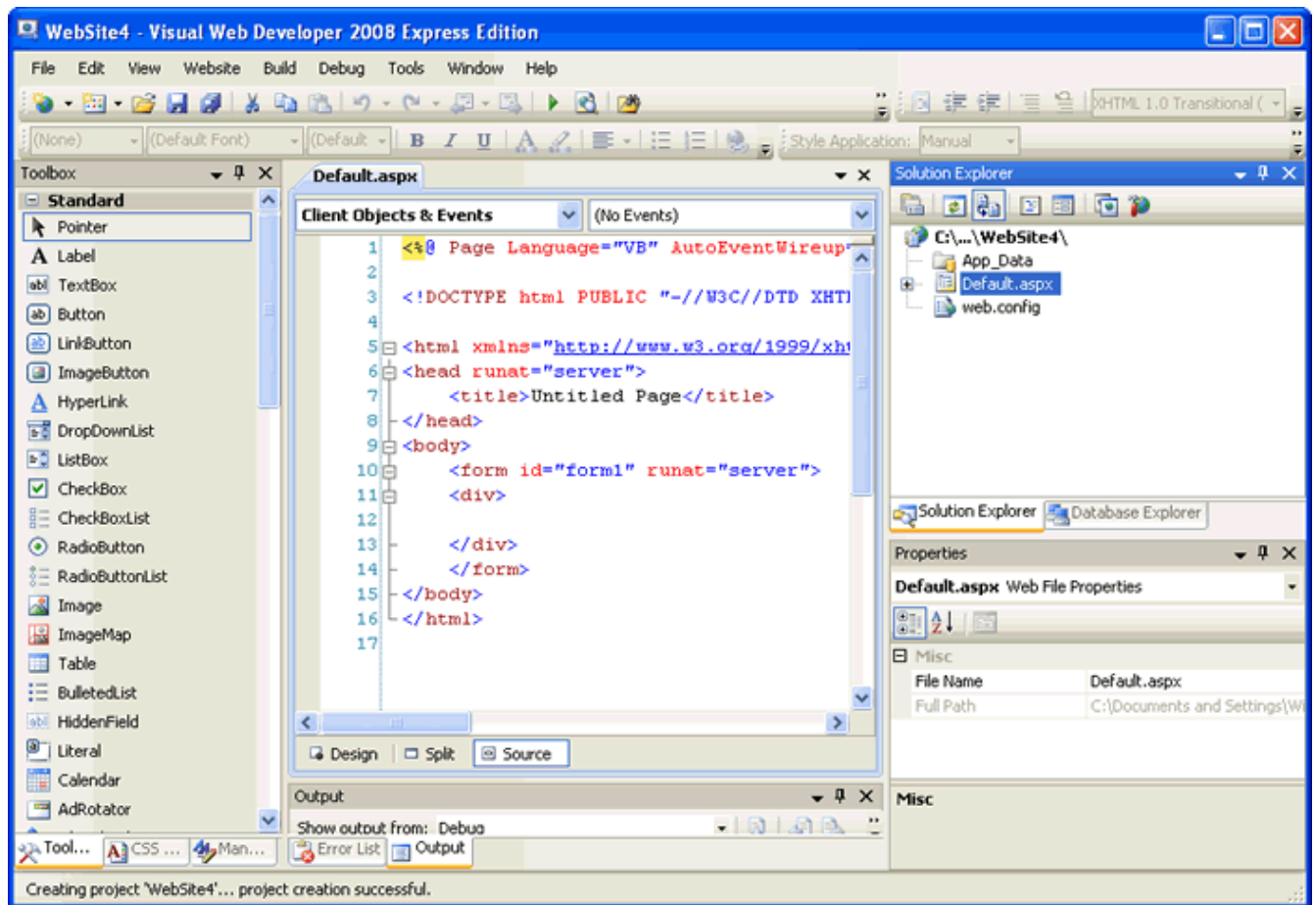


Podemos trabajar de dos formas: con la vista de diseño o con la del código fuente. Vamos a poner como predeterminada la vista diseño ya que es mas cómoda para trabajar, y normalmente, será la que mas utilizaremos.

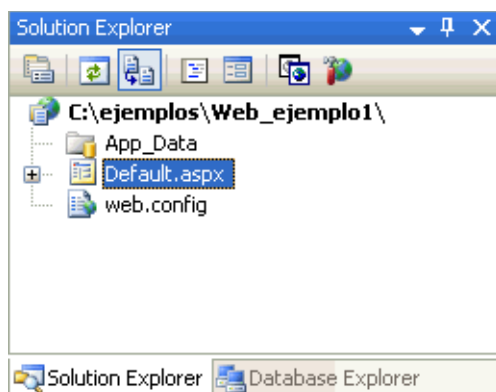
2. Editar páginas

Cuando tenemos un sitio vacío recién creado, nos encontramos con este aspecto ya conocido:

El IDE, estilos, temas y páginas maestras



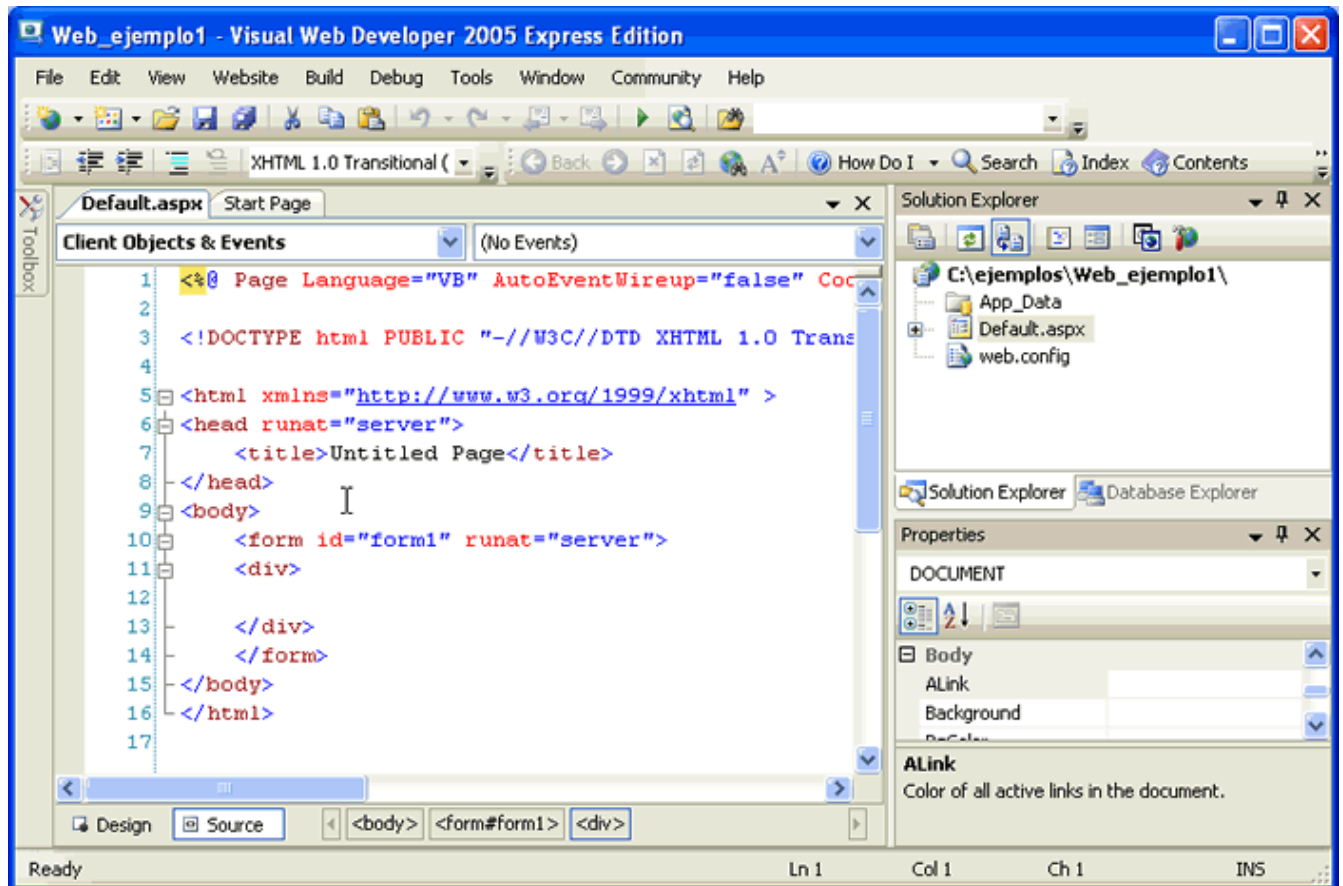
Por un lado vemos a la izquierda la barra de herramientas con los controles de servidor disponibles. En la parte central tenemos ya una página en diseño llamada "Default.aspx" y en la parte derecha dos elementos muy importantes: el explorador de soluciones y debajo las propiedades de los elementos seleccionados en cada momento. Si te fijas en el explorador de soluciones:



Podemos ver la página "default.aspx" que nos ha creado y un fichero de configuración para el sitio Web llamado "web.config".

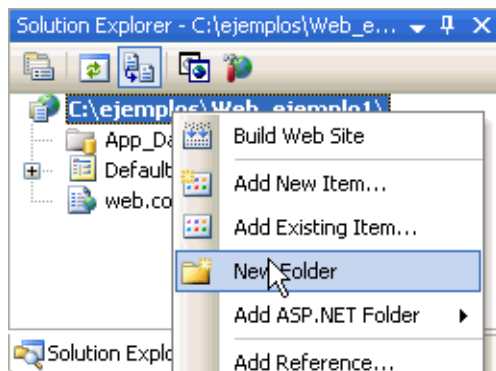
La página default.aspx aparece en modo de diseño, es decir, como en FrontPage donde podemos "pintar" directamente. La otra vista disponible es la de "Source" que podemos activar desde la parte inferior de esa página:

El IDE, estilos, temas y páginas maestras



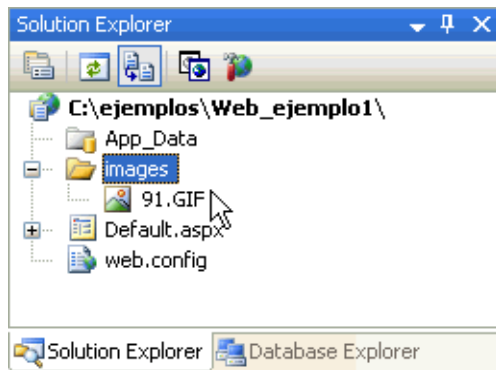
Todo esto te suena porque a lo largo del curso hemos estado editando el código para escribir pequeñas páginas Web pero ahora tenemos este entorno mucho mas amigable para el desarrollo.

Para cerrar páginas simplemente elegiremos la "x" de la parte superior de la derecha "Close" y si está modificada nos pedirá guardar los cambios. Para crear carpetas en nuestro sitio nos iremos al árbol del explorador de soluciones y seleccionando la raíz pulsamos con el botón derecho:



Por ejemplo crea una carpeta llamada "images" para almacenar los gráficos de nuestro sitio Web. Para copiar archivos dentro de esta carpeta de nuestro sitio Web simplemente podemos arrastrar los ficheros desde el explorador de archivos o utilizar el habitual "copiar y pegar":

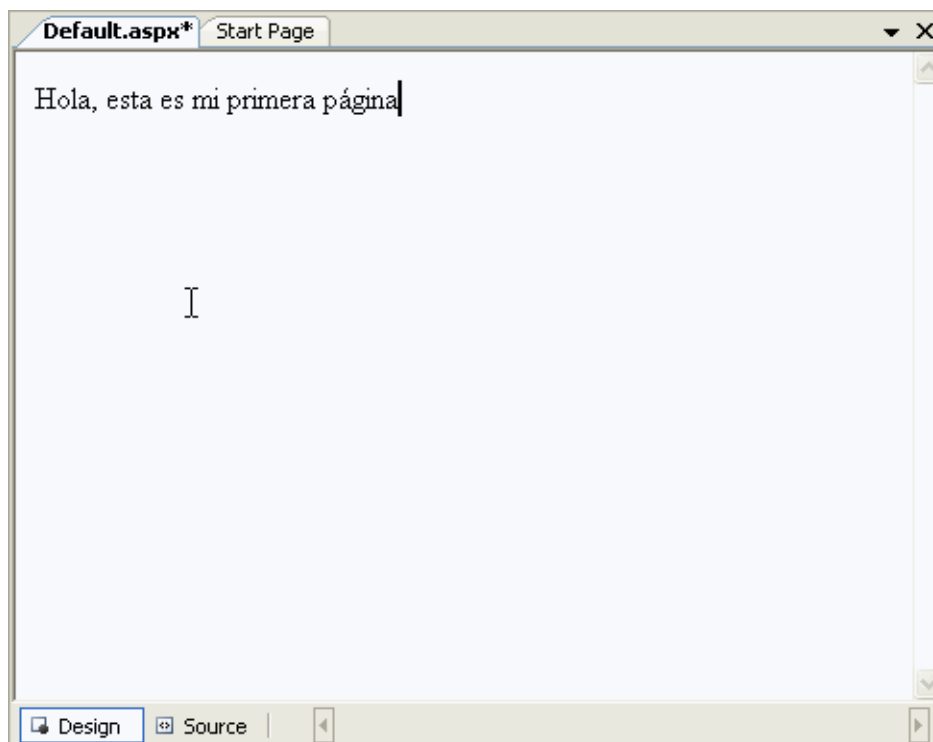
El IDE, estilos, temas y páginas maestras



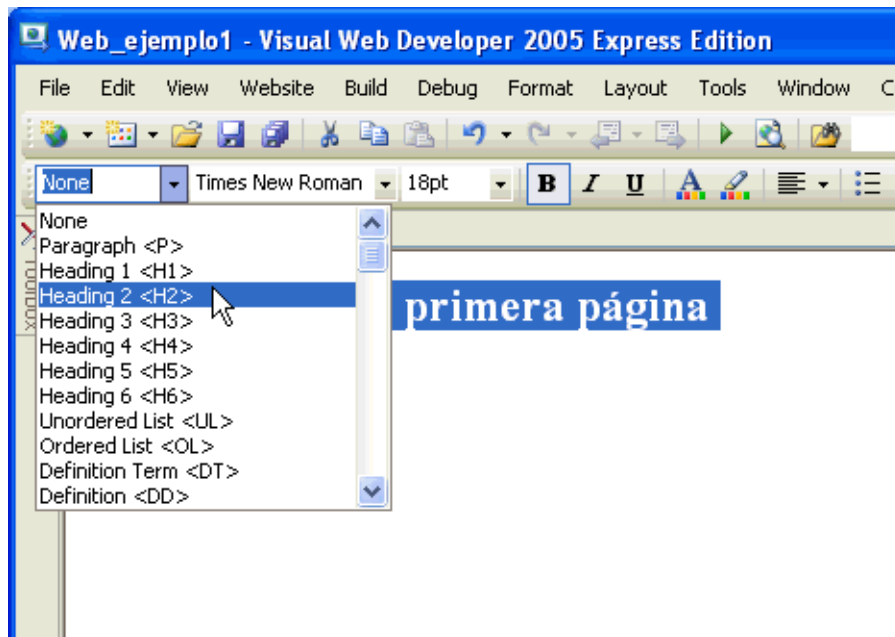
Finalmente con el botón derecho tenemos las opciones de borrar y renombrar tanto ficheros como carpetas.

Editar páginas

Para editar las páginas simplemente la seleccionamos de la lista de la derecha y al hacer doble clic se nos abrirá en modo de diseño en nuestra página. Localiza el cursor en esta página y escribe un texto de prueba:

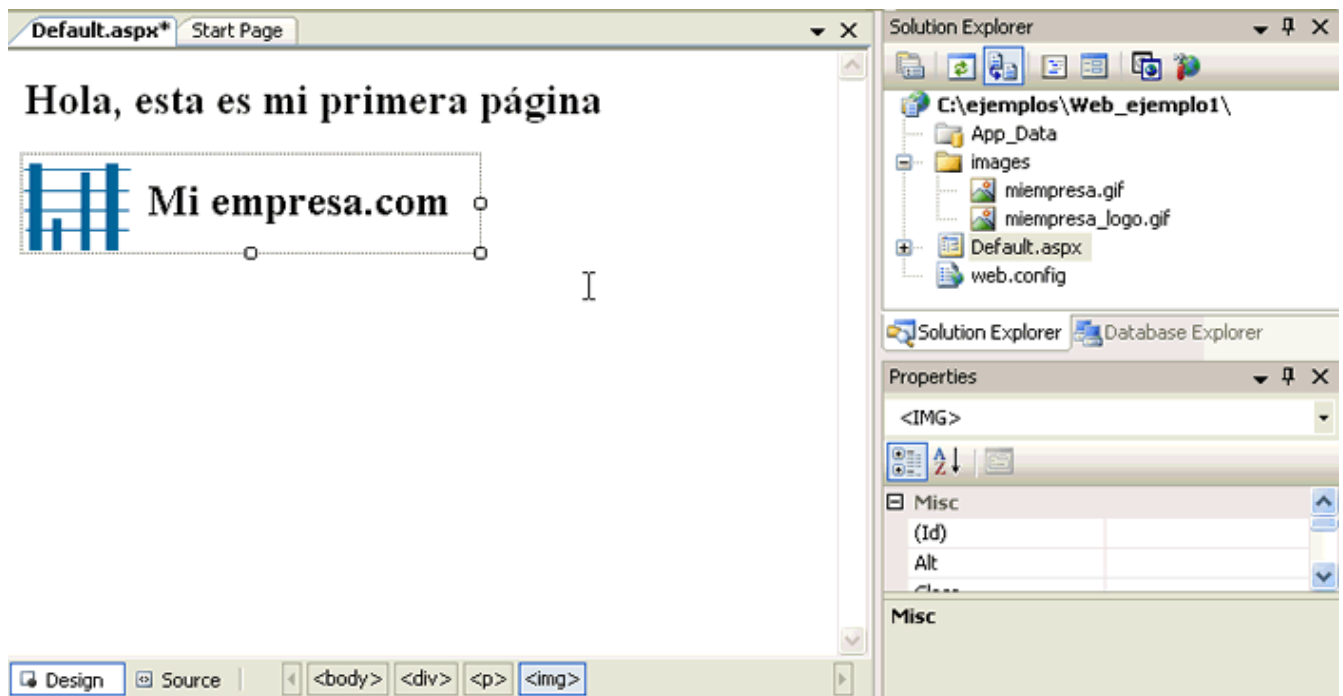


Para ponerle los estilos estándar de HTML utilizaremos la habitual barra de herramientas de los programas de Microsoft:



Añadir imágenes

Para añadir imágenes simplemente arrastraremos desde el explorador de soluciones hasta la zona donde la queremos colocar:



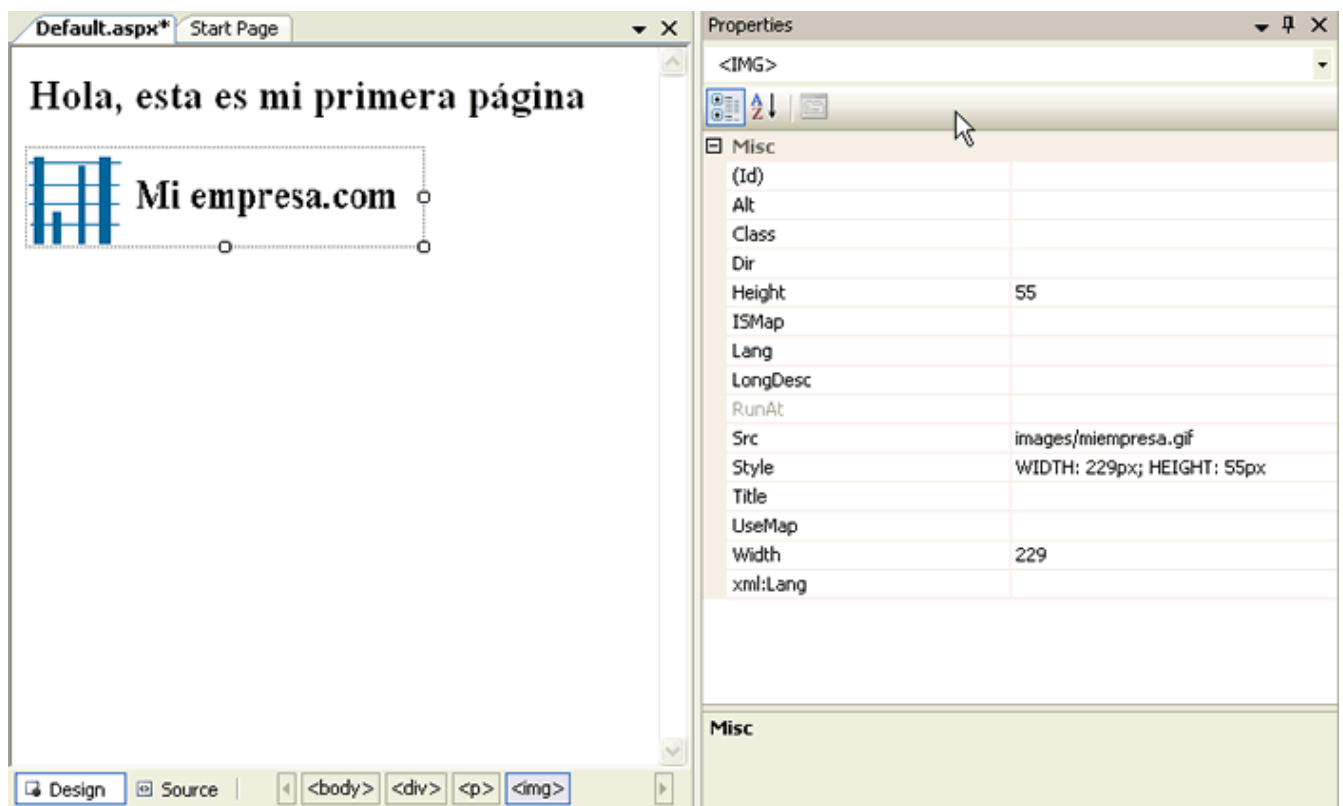
Si te fijas en los laterales tenemos la posibilidad de cambiar el tamaño del gráfico, al hacerlo nos muestra dinámicamente el tamaño en píxeles:

Hola, esta es mi primera página



Cambiar propiedades

Es una de las partes imprescindibles, ya sabes que todos los objetos tienen multitud de propiedades, con el panel de propiedades podremos cambiar todas las propiedades de cada objeto. Selecciona el gráfico, que sabes que es un elemento HTML estándar de tipo "" para ver sus propiedades:



En este caso este elemento tiene muy pocas propiedades ya que los HTML tienen muy pocas, no son como los controles de servidor. Si te es familiar el código HTML identificarás enseguida las etiquetas HTML de las imágenes: "Src" para el origen y "Height" y "Width" para la altura y anchura.

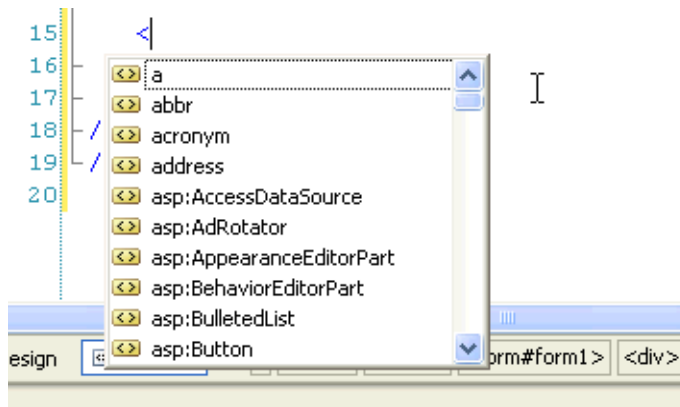
Vista de código

Ahora vamos a ver el código fuente de esta página, pulsamos en "source" para buscar lo que hemos introducido:

```
<h2>Hola, esta es mi primera página</h2>
```

```
<p>&nbsp;</p>
```

Que es el titular "H2" y la imagen "img". Vete acostumbrándote a este entorno y como identificar las etiquetas HTML porque cambiaremos mucho entre estas vistas. Vamos a escribir algo de código, verás que es muy similar a lo que ya vimos en FrontPage, por ejemplo un párrafo, así que escribe el símbolo "<":



ya ves que se parece bastante porque nos muestra todas las etiquetas que se pueden utilizar, escribe pues "<p>":

```
<p>*</p>
</div>
</form>
/body>
/html>
```

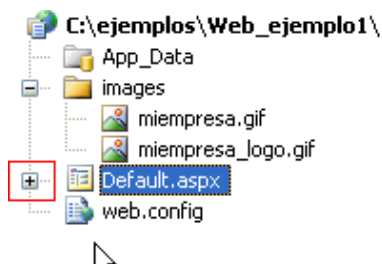
Y nos escribe ya de forma automática el símbolo del fin de párrafo "</p>"

Bien para terminar nuestro ejemplo grabaremos el fichero con el icono del disco habitual y veremos dónde está la página de código...

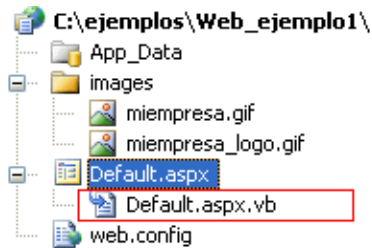
Archivo de código

Para no mezclar el código visual basic con el HTML se puede crear una página de código asociada al fichero html. De esta forma era mas fácil de mantener y reutilizar. Para saber si las páginas aspx tienen estas otras de código simplemente veremos si tiene el símbolo "+" para expandir el árbol en el explorador de soluciones:

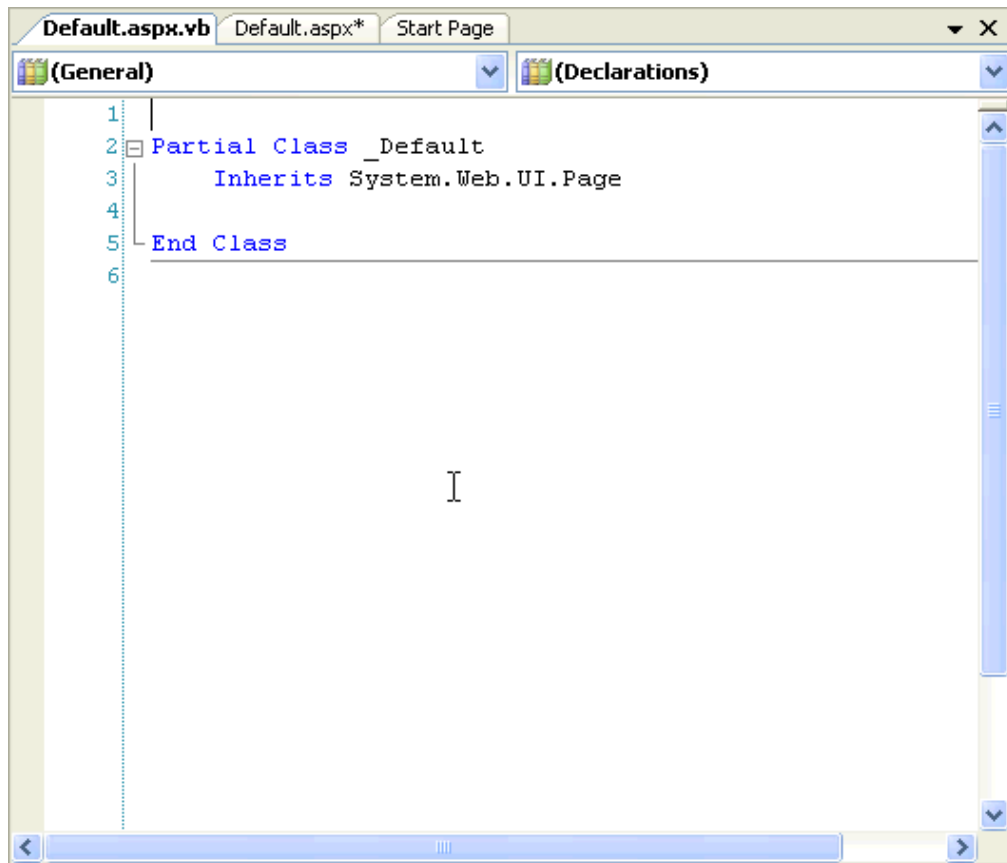
El IDE, estilos, temas y páginas maestras



Pulsa en el símbolo:



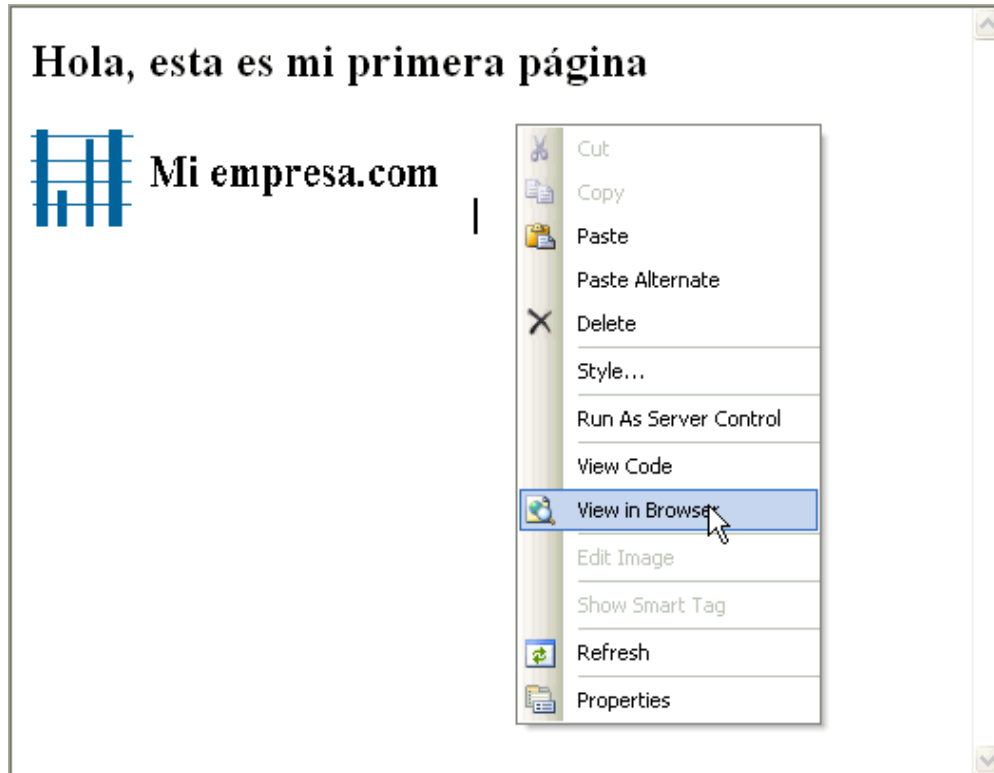
Y vemos que hay una página con el mismo nombre y con extensión ".vb". Haz doble clic para editarla:



Como está vacía sólo tiene el código que indica que es una clase heredada de una página Web.

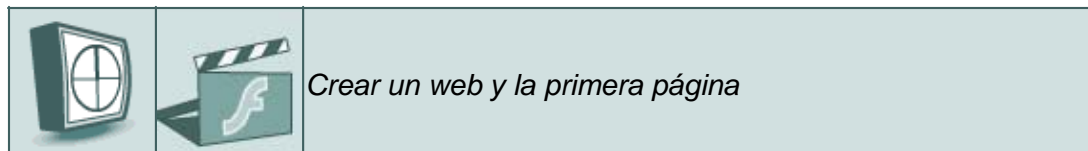
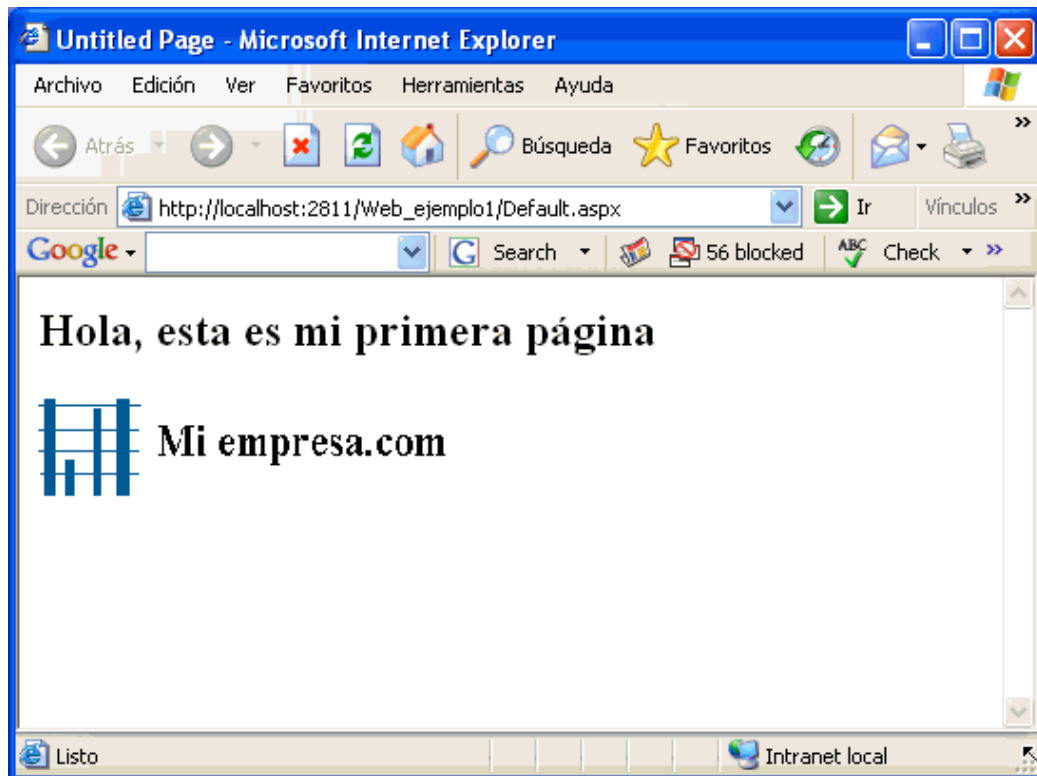
Ver la página en el navegador

Ya nos queda ver el resultado de la página en el navegador. Tenemos varias formas de hacerlo: con las teclas control+F5, utilizando el botón de la barra "View in Browser" o haciendo clic con el botón derecho del ratón en una zona libre de la página:



Que nos muestra la página (también podemos ver la página en modo depuración como hasta ahora):

El IDE, estilos, temas y páginas maestras

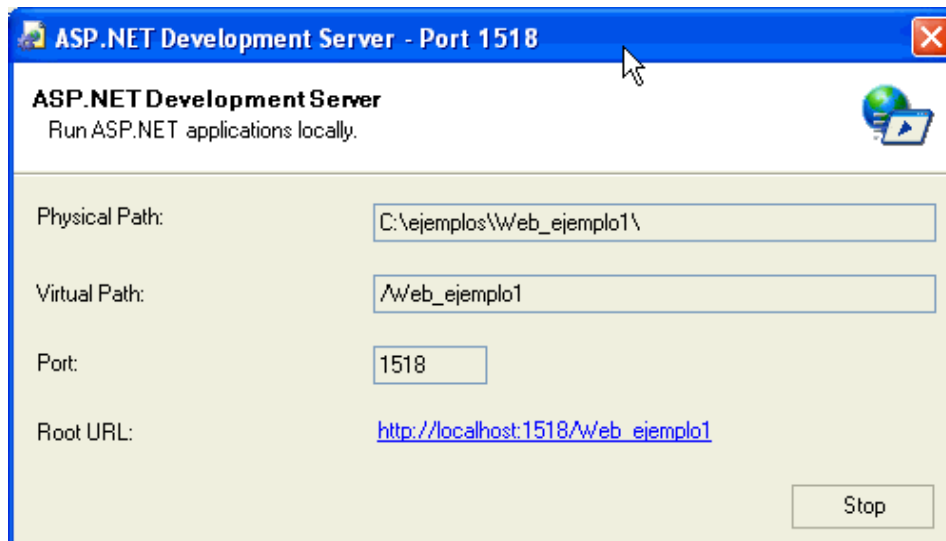


Fíjate en la dirección: <http://localhost:2811/...> además en la parte inferior derecha podemos ver:



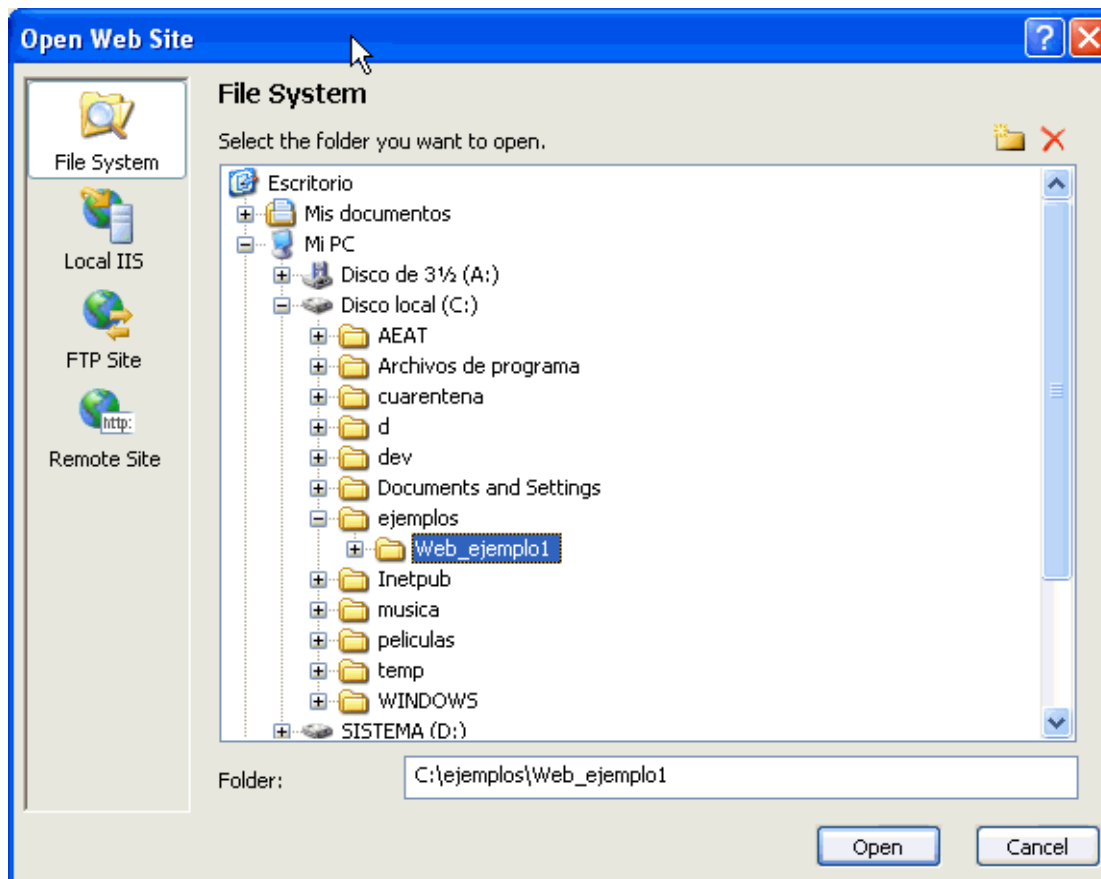
Esto es una de las cosas mas interesantes que tiene nuestro entorno de desarrollo ya que incorpora un servidor Web, si, has leído bien, no nos haría falta en local el IIS. Nos detenemos aquí un poco. Al final del curso veremos como crear y abrir los sitios Web de IIS pero de momento nuestro entorno incorpora un servidor Web limitado pero suficientemente potente como para que pueda ejecutar nuestras páginas aspx. Obviamente es algo temporal y es simplemente para crear Web locales en nuestro equipo. Lo suyo desde luego es abrir y manejar los del servidor Web "de verdad" que es el IIS, esto lo veremos mas adelante...

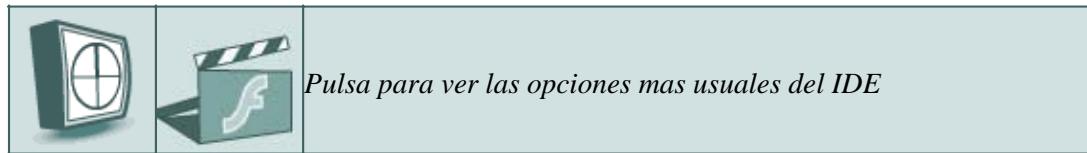
Si pulsamos en ese icono del servidor Web con el botón derecho para ver los detalles:



Podemos ver que entre otras cosas se ejecuta en el puerto "1518" en lugar del habitual 80, no importa como es para pruebas no pasa nada.

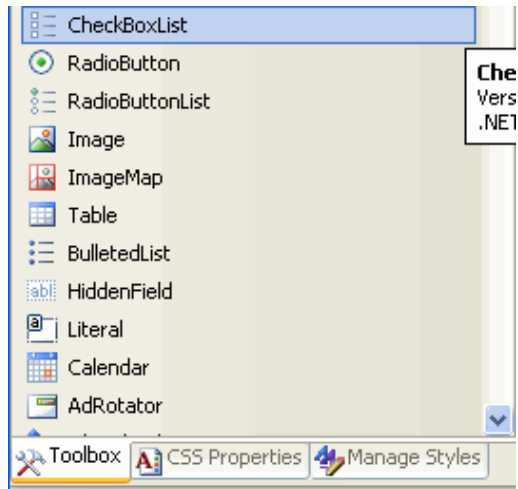
Vamos a terminar nuestro proyecto grabando los ficheros y seleccionando "Close Project" del menú "File". Para abrirlo utilizaremos "Open Web Site" del mismo menú "File" y nos mostrará automáticamente el último que hemos editado:





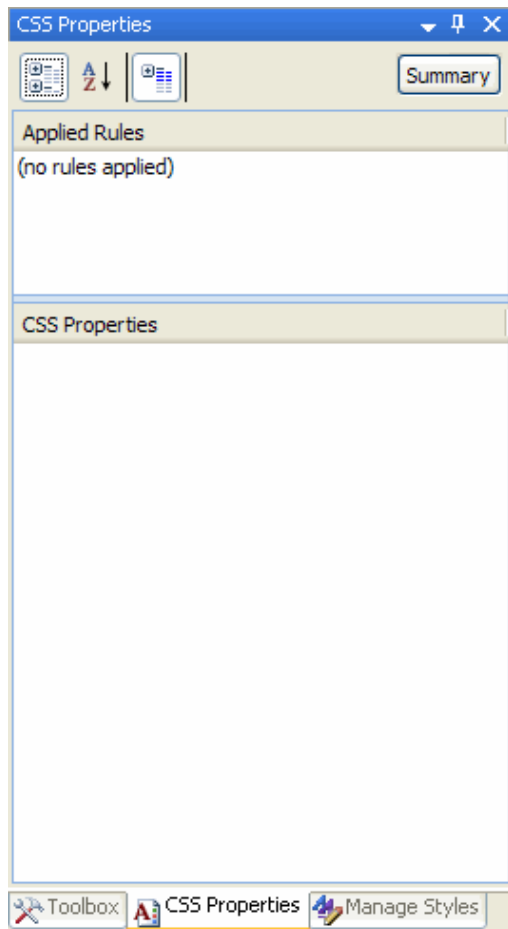
2.1 Mas vistas

En la parte de la izquierda ya hemos visto que tenemos el cuadro con todos los controles, pero además si te fijas debajo tenemos otras dos pestañas:



La primera solapa es el ya conocido cuadro de herramientas, la segunda solapa nos va a mostrar:

El IDE, estilos, temas y páginas maestras

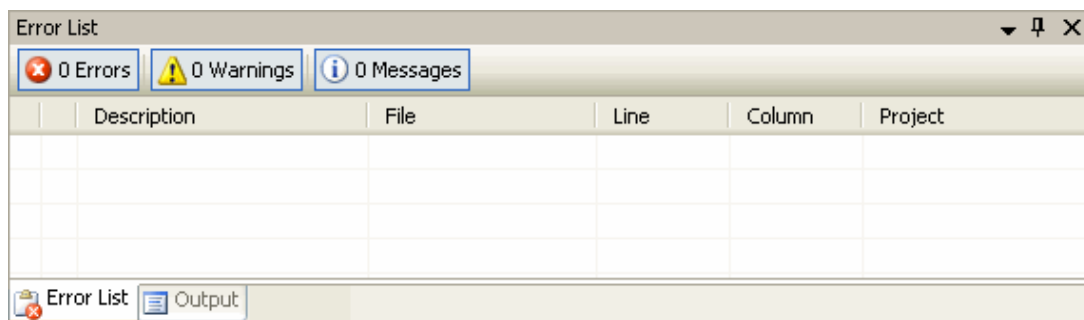


Los estilos utilizados en las páginas y que veremos mas adelante. La tercera pestaña nos permite:

El IDE, estilos, temas y páginas maestras

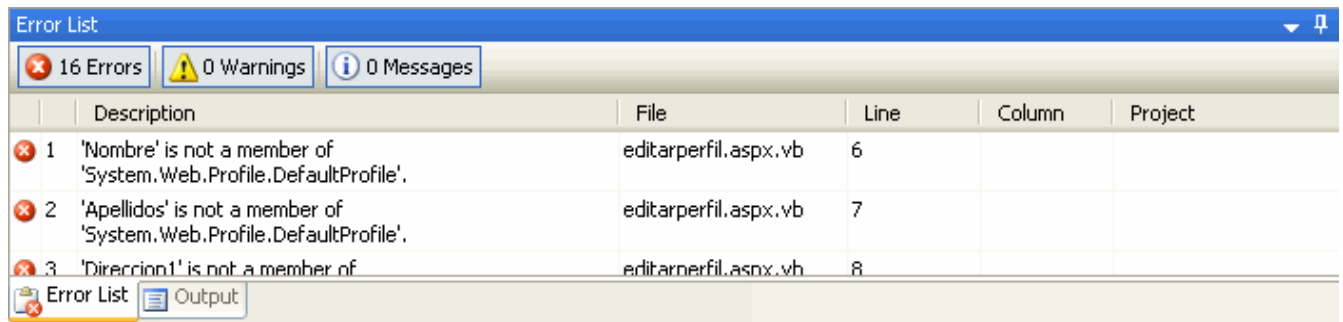


Crear nuevos estilo teniendo debajo una vista previa de ellos. En la parte central e inferior tenemos:



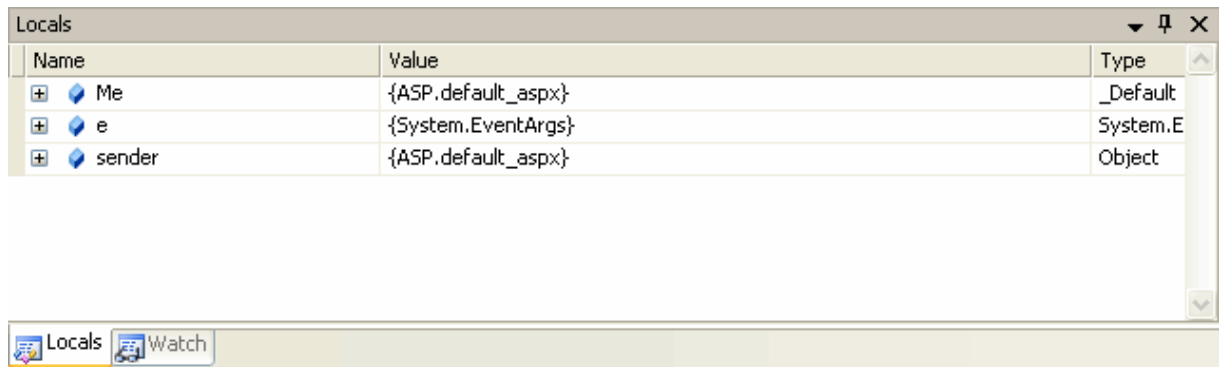
Que es un imprescindible ayuda para nuestra aplicación Web. Nos indicará los errores que tengamos en la generación de las páginas y otros detalles. Por ejemplo:

El IDE, estilos, temas y páginas maestras

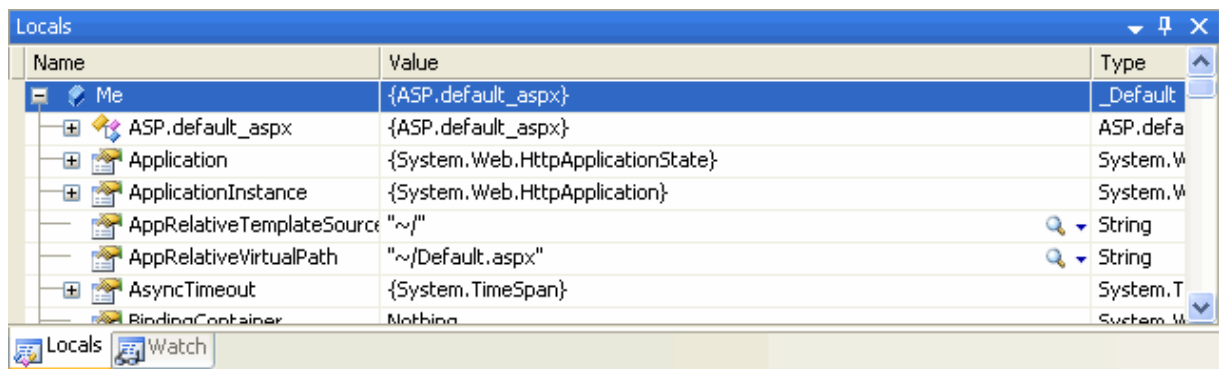


Nos muestra una lista de todos los errores de nuestra aplicación Web, ojo, de toda la aplicación, no de la página. Date cuenta que se genera todo el sitio Web que está configurado bajo unos ficheros de configuración globales, como "web.config". Por tanto, se compila toda la aplicación y se muestra una lista de los errores de todas las páginas. Si hacemos clic en cada uno de ellos nos podremos mover a las páginas de los errores y tendremos el texto destacado para corregirlos.

Si estamos depurando la página y estamos detenidos en un punto de interrupción las ventanas de abajo cambian para mostrarnos información sobre la ejecución. Por un lado a la izquierda tenemos:

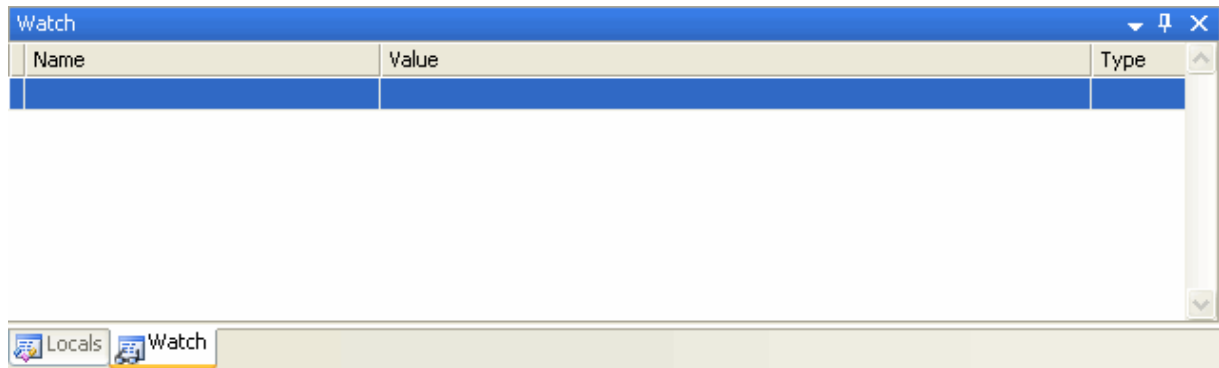


Que es información relativa al propio formulario "Me", a los eventos del controlador de eventos que está deteniendo la ejecución con el punto de interrupción y otros objetos. Si le das al signo "+" de "Me" verás todos los datos referentes a la página y que nos servirá de gran ayuda para saber el estado de todas las variables entre otras cosas:

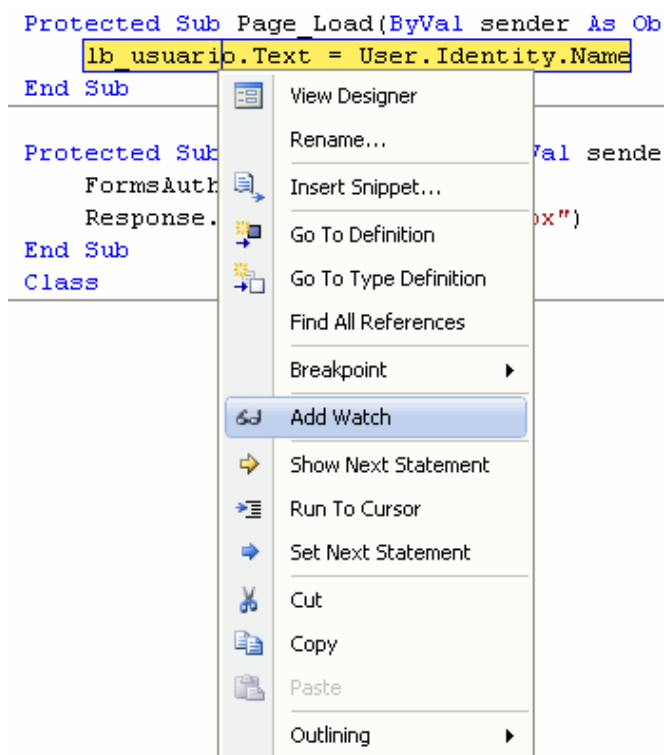


La otra pestaña es "Watch":

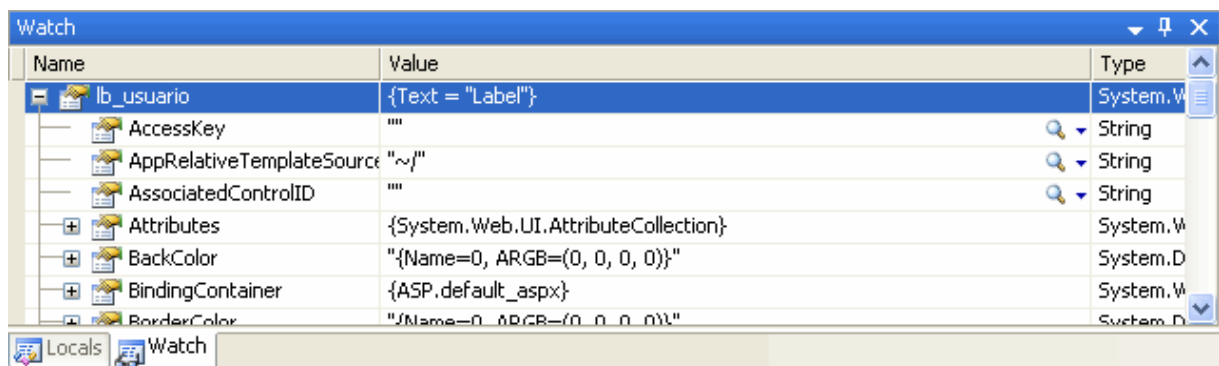
El IDE, estilos, temas y páginas maestras



Que permite poner variables y expresiones para ver como va cambiando su contenido. Para añadir una variable a monitorizar nos pondremos encima de ella y diremos "Add Watch":



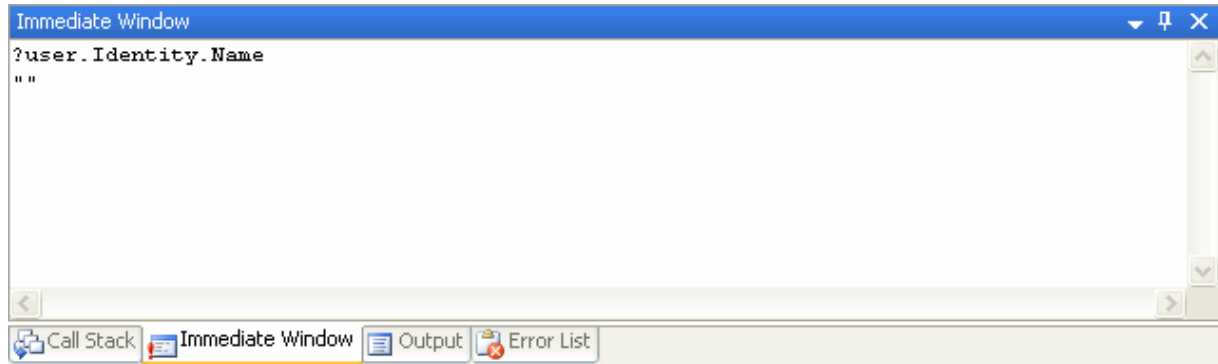
Y aparecerá en el panel:



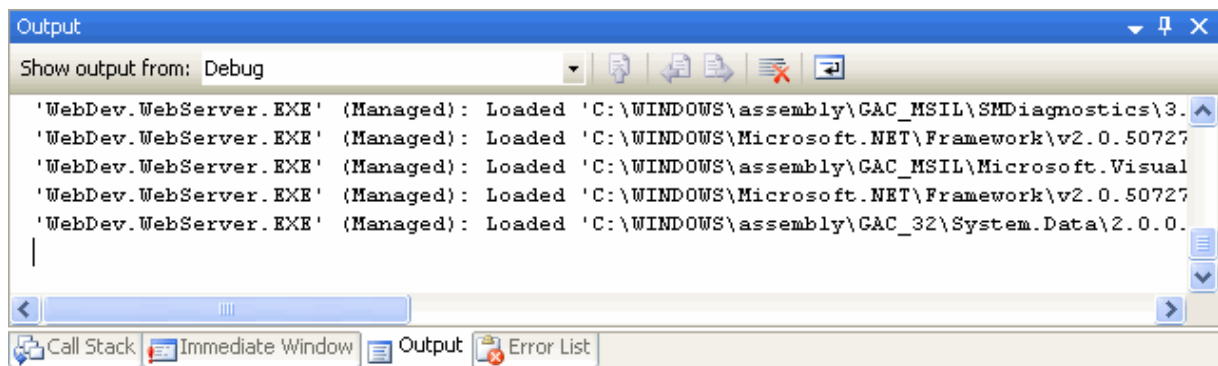
El IDE, estilos, temas y páginas maestras

En este caso era un control que nos muestra todas sus propiedades pero podemos añadir variables, matrices, ... y al ir realizando la depuración podremos ver los valores que va obteniendo.

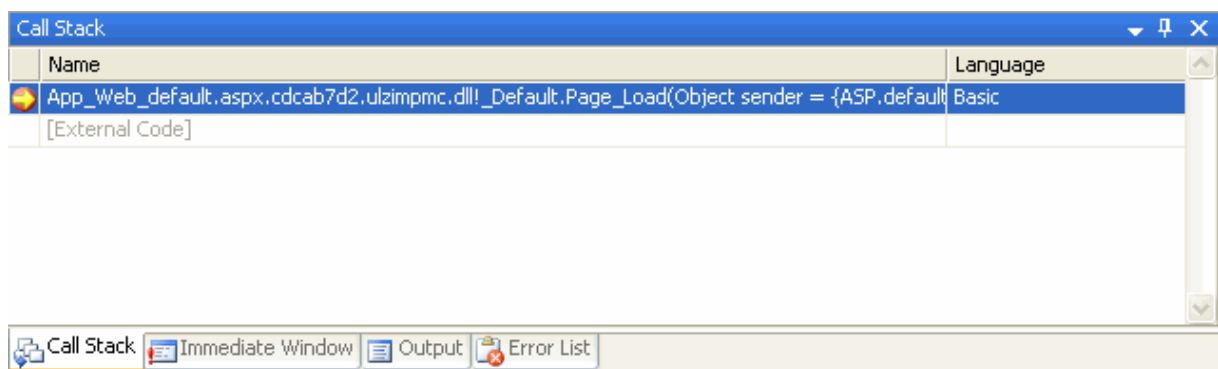
En el panel central y en el mismo modo de ejecución en el que estamos aparecen estos paneles además del de los mensajes que ya hemos visto:



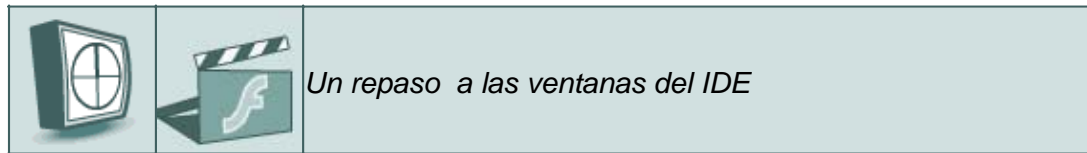
El panel "Immediate" nos permite consultar cualquier valor actual escribiendo un "?" + variable. Por ejemplo ?i nos mostrará el valor de la variable i, si existe claro. En el ejemplo he puesto que me diga el valor de la Identity.Name del objeto User. Es muy útil esta pantalla porque nos permite preguntar por cualquier valor y así depurar mas fácilmente la aplicación. El siguiente panel es el de "Output":



Que son los mensajes que nos ha puesto cuando ha generado la aplicación Web. El último panel "Call Stack":



Nos muestra la pila de llamadas por las que va pasando nuestra aplicación. Es información mas técnica pero cuando tenemos muchas llamadas entre procedimientos y páginas puede ser interesante para saber el proceso de llamadas que ha estado realizando.



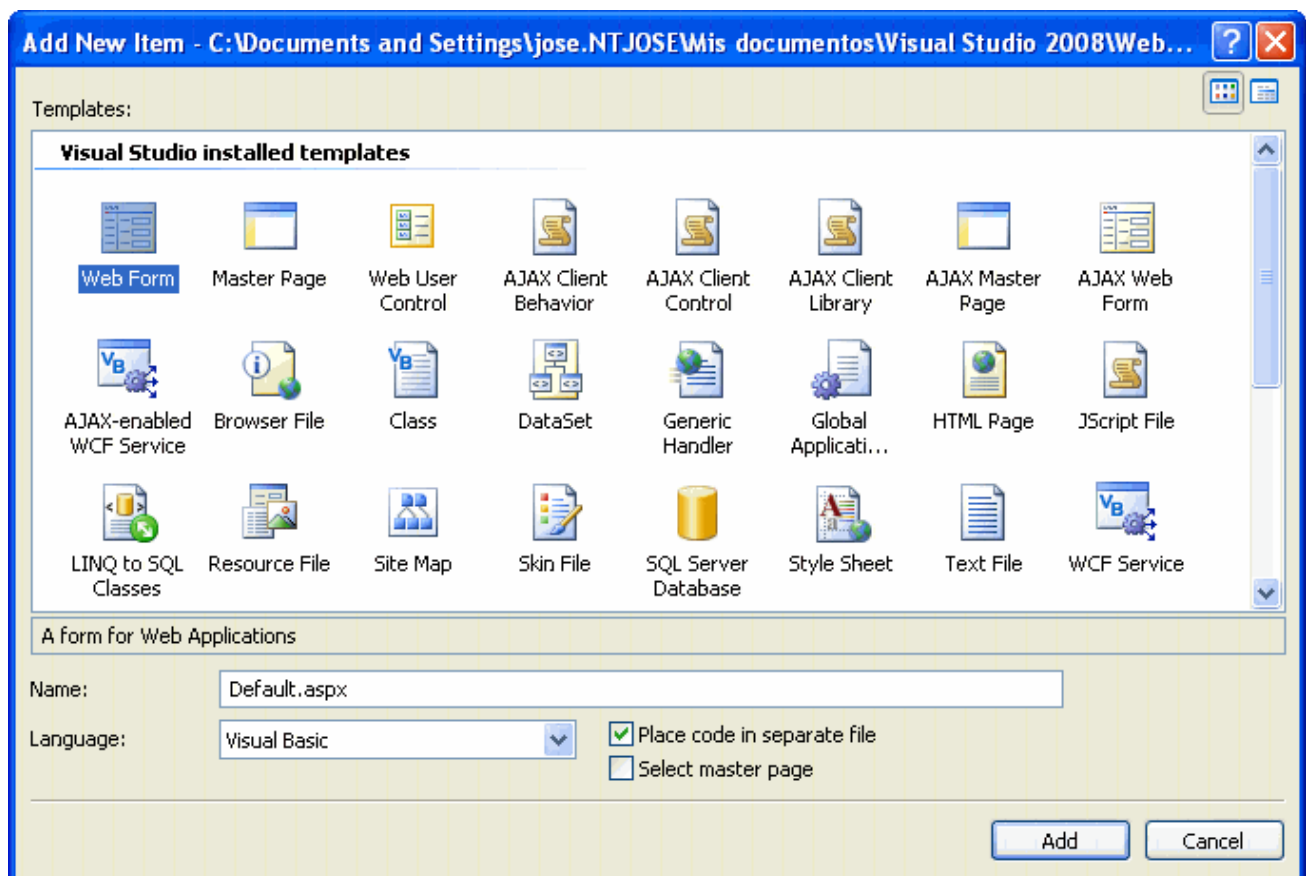
3. Crear páginas

Si, ya se que a estas altura no os voy a enseñar a hacer páginas Web pero bueno, como estamos en un entorno nuevo vamos a ver un resumen de los pasos correctos que debes dar para que todo siga ordenado y coherente con nuestro Web. Comencemos, con el Visual Web Developer podemos crear dos tipos de páginas:

- **Web Forms** que son páginas con extensión .aspx y que contienen código y controles de servidor
- **Páginas HTML**. Que contienen código HTML estándar y su extensión lógicamente es .htm (ó .html)

Así que si queremos utilizar la capacidad de ASP.NET debemos crear obviamente páginas con la plantilla de Web Form. Muchas otras páginas de información de nuestro Web serán simples páginas HTML.

Para crear páginas ya sabes el método, nos ponemos con el ratón en la parte del explorador de soluciones y seleccionaremos "add item":

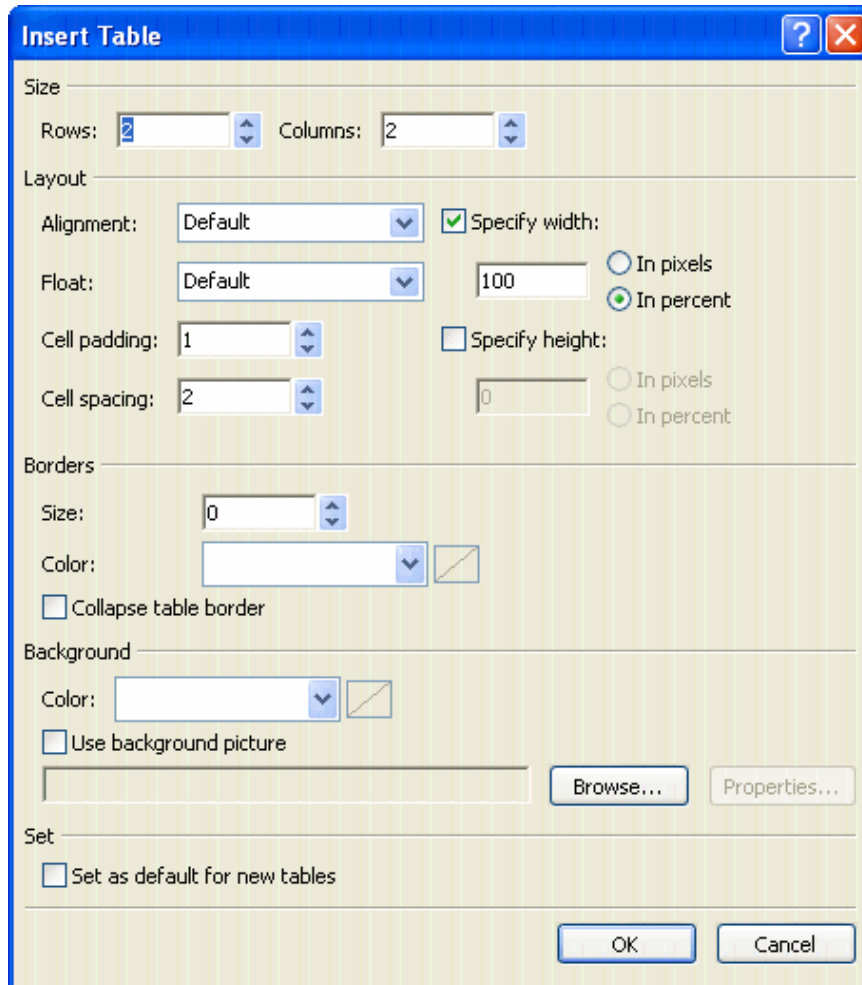


En la parte de abajo le indicamos el lenguaje y en la parte derecha si queremos mantener el código en una página aparte y si esta página va a pertenecer a una maestra (que veremos enseguida)

3.1 Crear tablas HTML

Si ya has trabajado con tablas sabrás que es un elemento imprescindible para colocar los distintos elementos de la página, además de por supuesto mostrar resultados de una consulta a una base de datos por ejemplo.

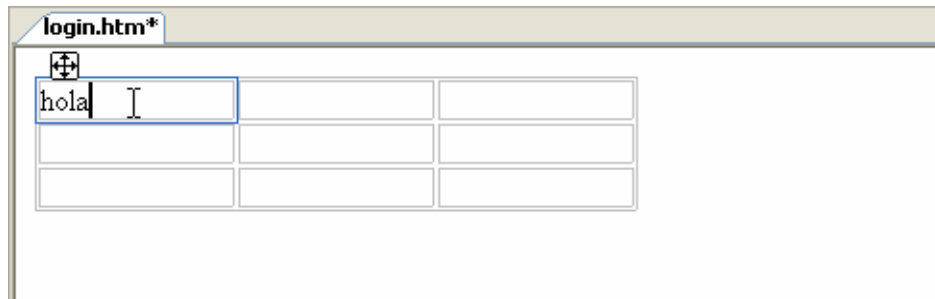
Para crear una tabla nos vamos a "Layout" y luego la primera opción "Insert Table":



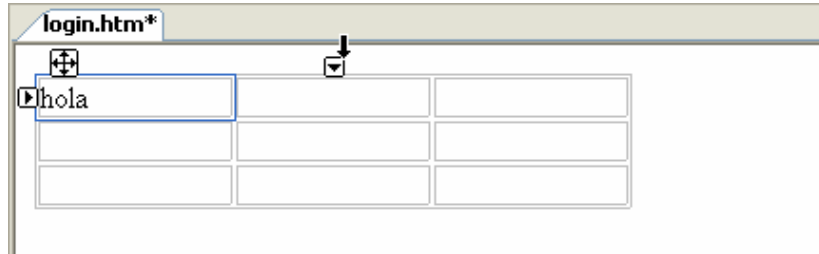
Como veremos mas adelante podemos crear una hoja de estilos CSS para definir las tablas así cuando creemos una tendrá el estilo que venga definido en esa página con lo que todas las tablas de nuestro sitio Web tendrán el mismo estilo.

Te dejo a curiosear las opciones porque no hay nada nuevo... lo único que cuando esté creado por supuesto podremos editarla directamente:

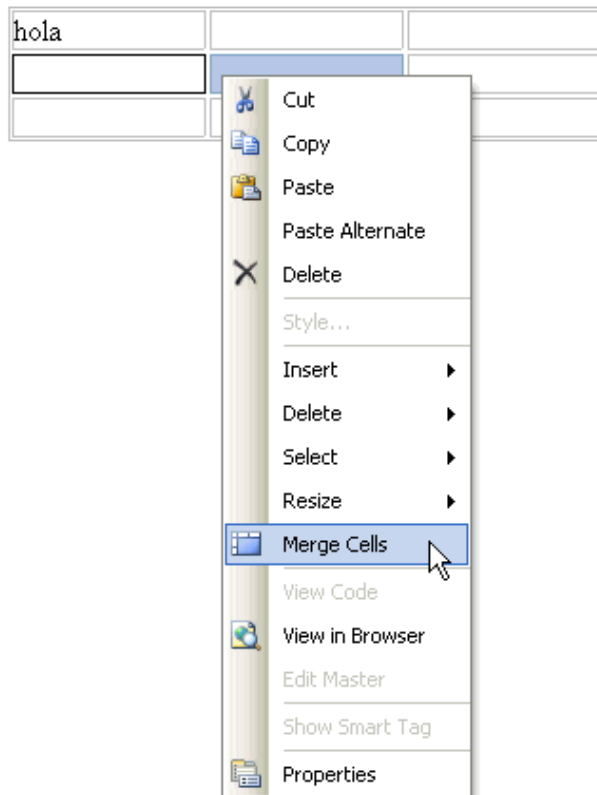
El IDE, estilos, temas y páginas maestras



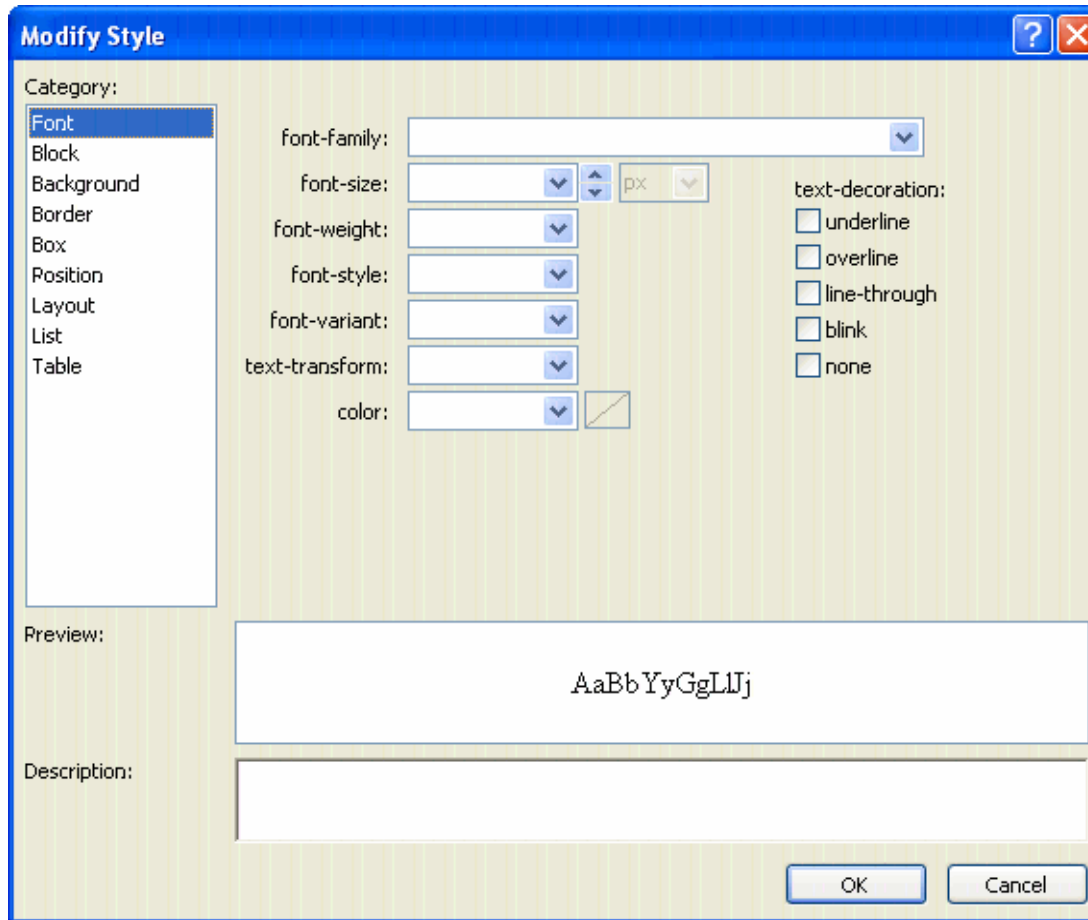
Para seleccionar columnas o files utiliza los marcadores que aparecen en ellas:



Para mezclar celdas márcalas y luego el botón derecho:

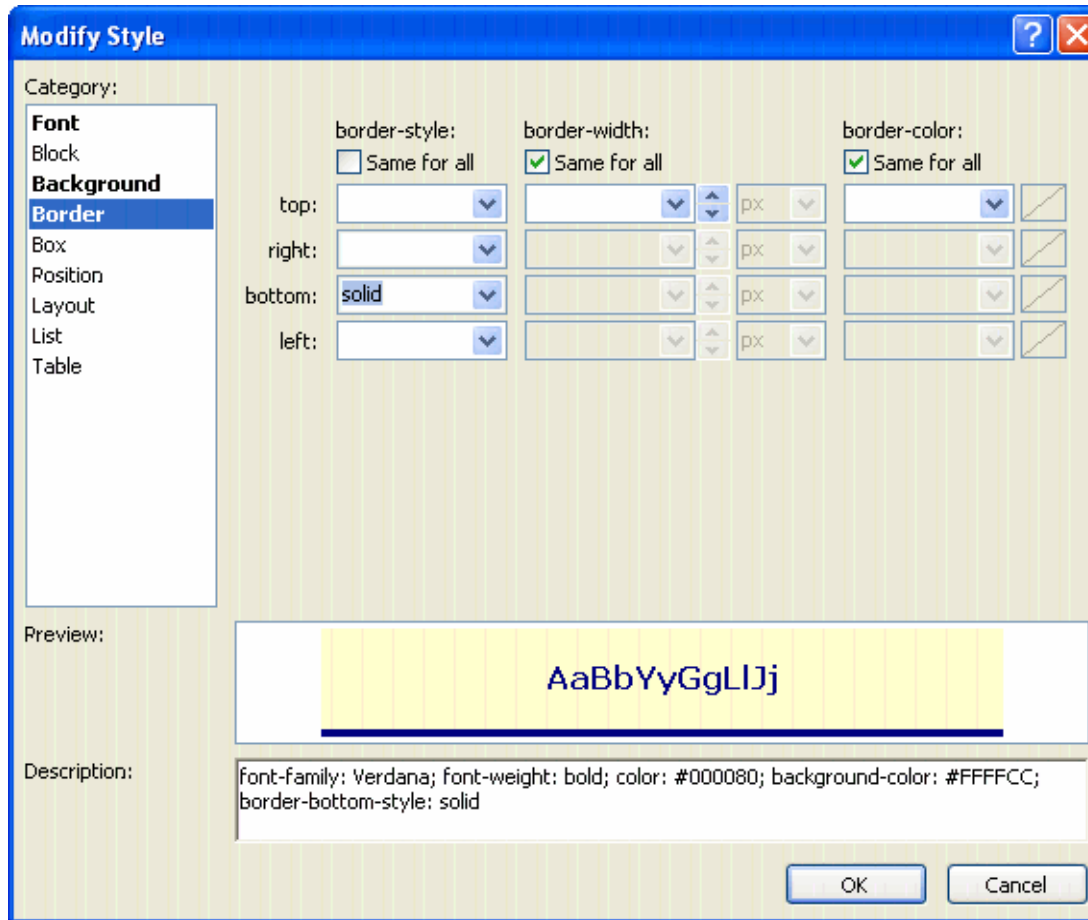


La edición de insertar, eliminar celdas y estas operaciones son muy similares a los demás entornos de Microsoft. Podemos por supuesto acudir al generador de estilos bien con el botón derecho o desde la página propiedades y luego "Style":



Fíjate que cada vez que seleccionamos un objeto lo tenemos señalado a la derecha en esa página de propiedades donde podremos cambiarle sus valores y, como ahora, su estilo. Vamos a aplicar un pequeño estilo a las celdas de arriba, así que marcamos la fila entera, pulsamos con el botón derecho en "Propiedades" y luego en las propiedades de la ventana inferior derecha pinchamos en "Style"

Donde hemos seleccionado el tipo de letra "Verdana", color azul, tamaño de 14 puntos, ... Como queremos destacar estas celdas de título en "BackGroud" ponle un color de fondo a estas celdas. Luego en "Text" pon que centre el texto. Finalmente en los bordes pondremos solo el inferior:



Aceptamos, y completas la tabla con este texto y un formulario de login, recuerda, arrastrando un control de servidor "login":

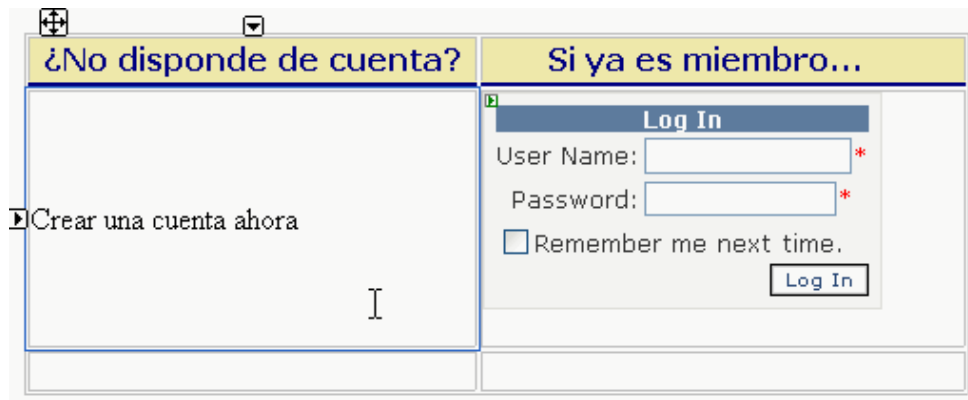
¿No dispone de cuenta?	Si ya es miembro...
	<div> <div>Log In</div> <div> User Name: <input type="text"/> * </div> <div> Password: <input type="password"/> * </div> <div> <input type="checkbox"/> Remember me next time. </div> <div>Log In</div> </div>

Bien, esto es sencillo, como ves utilizaremos mucho las tablas para poder colocar controles y dar formato a la página.

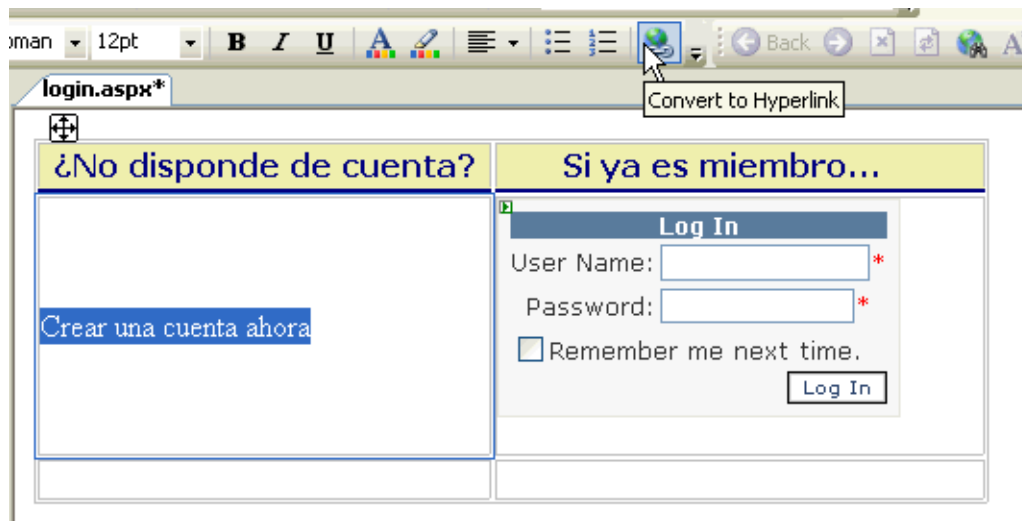
3.2 Añadir enlaces

Otra de las funciones básicas es crear enlaces así que escribimos esto en la tabla:

El IDE, estilos, temas y páginas maestras

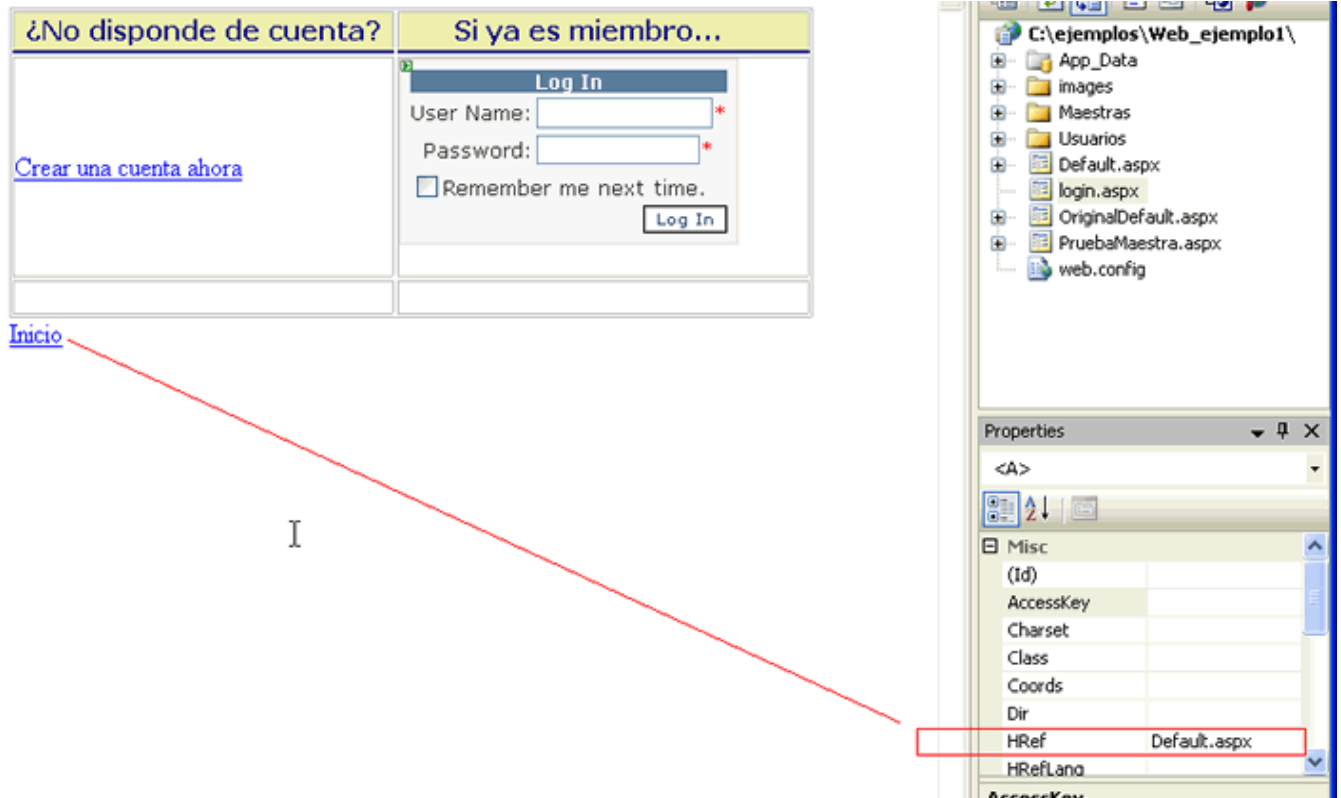


Marcamos el texto que queremos convertir en enlace y le damos al botón de la barra de botones:



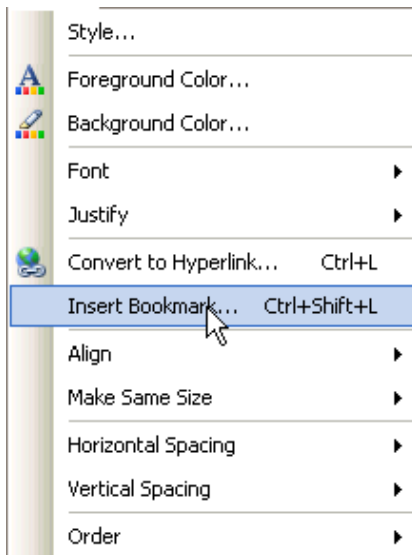
Escribe ahora cualquier texto y pulsa en "Ok". Si queremos poner un enlace a una página de nuestro sitio Web sólo debemos arrastrar la página del explorador de soluciones, por ejemplo vamos a arrastrar la "default.aspx" y le cambiamos el texto para que ponga "Inicio":

El IDE, estilos, temas y páginas maestras

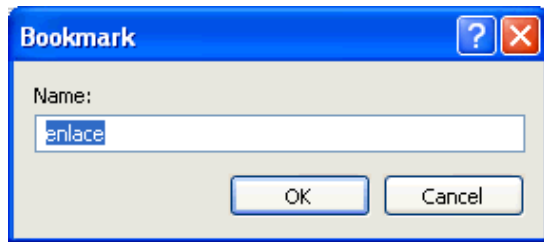


Como ves le he cambiado el texto pero el enlace permanece a la página de inicio.

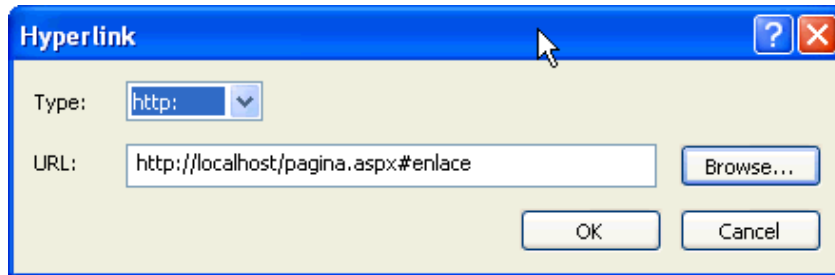
Los marcadores son enlaces a determinadas partes de la página. Es decir, además de decirle que salte a una página podemos decirle a una parte de esa página mediante un marcador. Para insertar un marcador simplemente marcamos el texto y luego nos iremos a la opción "Insert Bookmark" del menú "Format":



Que nos muestra:



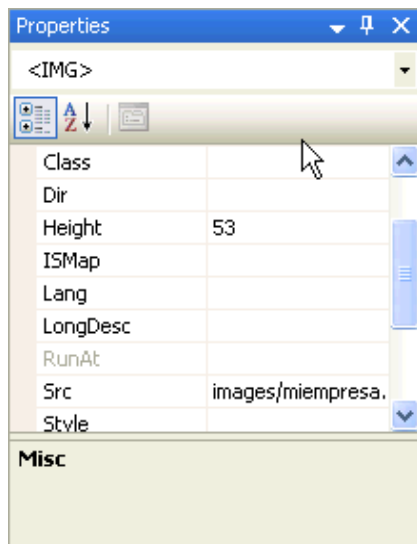
Le ponemos un nombre y luego la diferencia está cuando enlazamos esta página desde otro sitio porque debemos indicarle el nombre de este marcador:



3.3 Imágenes

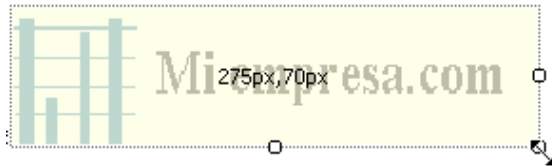
La forma mas sencilla de insertar imágenes es como hemos visto antes... las incluimos en nuestro proyecto en el explorador de soluciones y luego simplemente las arrastramos a nuestra página Web.

Después para editar sus propiedades ya sabes que podemos ir a la parte derecha:

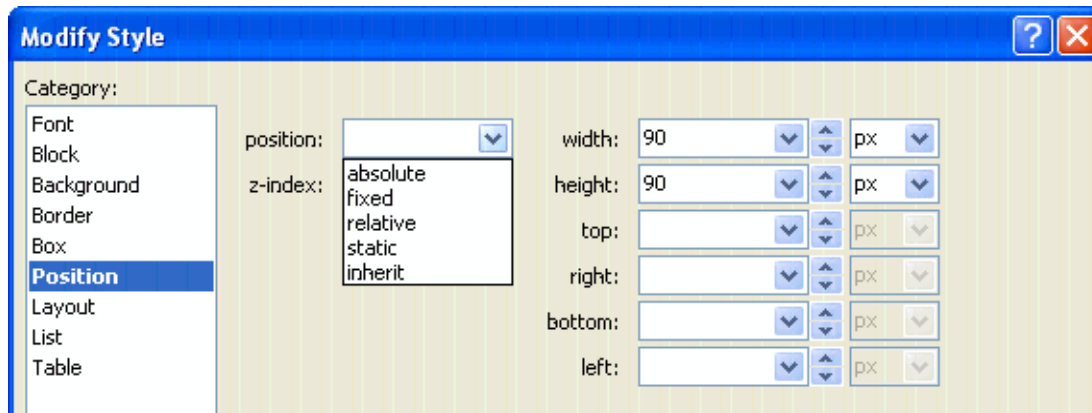


Para cambiar de tamaño elige el mejor sistema: o pones las propiedades del altura y anchura o con las esquinas de la figura:

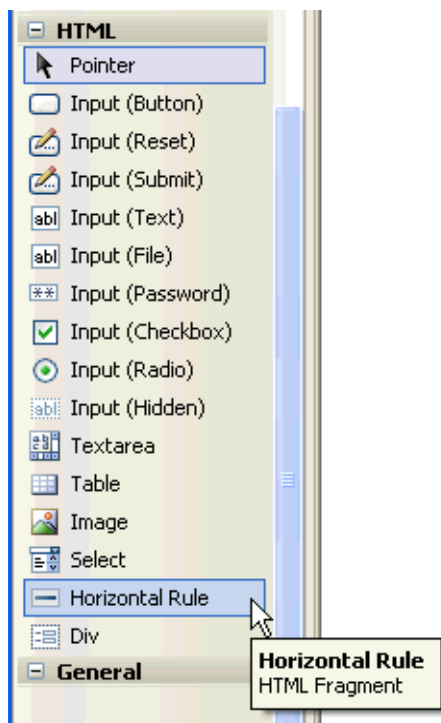
El IDE, estilos, temas y páginas maestras



Estas imágenes también tienen su parte de estilos, es decir, podemos añadir algunos elementos que nos permiten modificar su diseño y comportamiento. Con el botón derecho en la imagen:

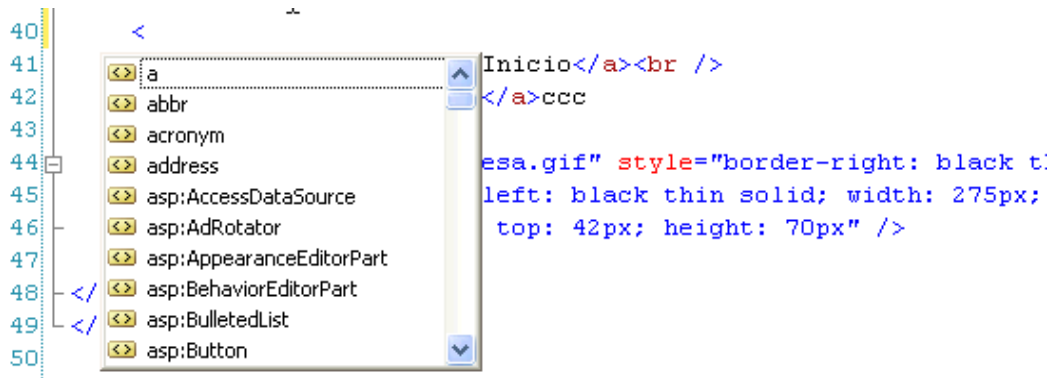


Para finalizar este repaso a los elementos básicos de HTML vemos las líneas, elementos simples para separar partes de la página. Para esto arrastra sin más el control HTML de la izquierda:

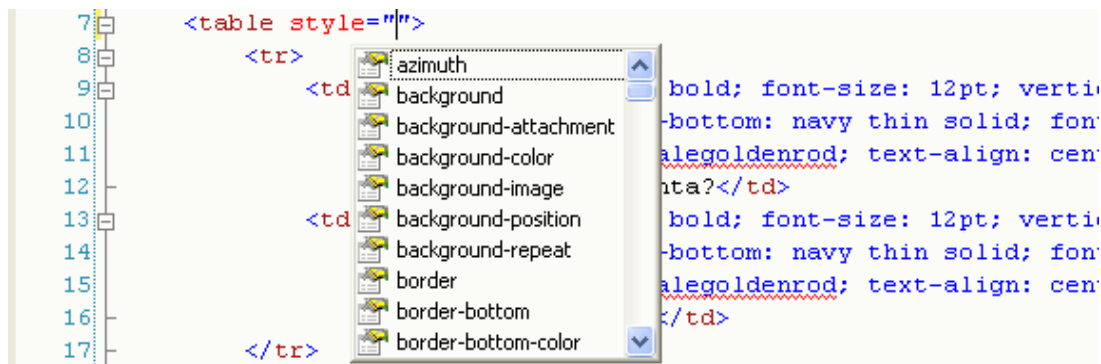


3.4 Editar el código fuente

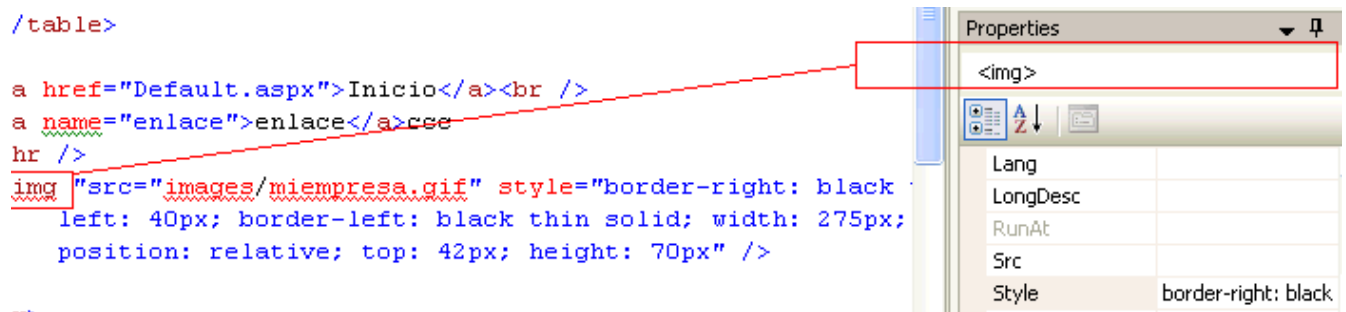
Paso esto un poco rápido porque lo conocemos de sobra de nuestro editor, lo que nos toca ahora es que aquí tenemos las mismas facilidades a la hora de escribir código a mano para los estilos:



La novedad la tenemos en los estilos porque al escribir un estilo nos muestra todos los disponibles en nuestra hora de estilos CSS:



De todas formas si tenemos el elemento seleccionado, por ejemplo una imagen "img" podemos ir a la ventana de propiedades, asegurándonos de que efectivamente aparece, y modificar los estilos desde la propiedad "Style":



Bueno, estamos en un curso de ASP.NET y parece que estas cosas no deberíamos verlas pero ya que nuestro entorno permite crear sitios Web con todos los elementos avanzados disponibles, incluido la gestión de estilos, debemos detenemos a hojear estos temas.

Es algo que personalmente no había utilizado mucho antes porque entre otras cosas no existía, había que hacerlo todo a mano y la verdad es que no había tiempo para dedicarse a cosas en principio no productivas. En

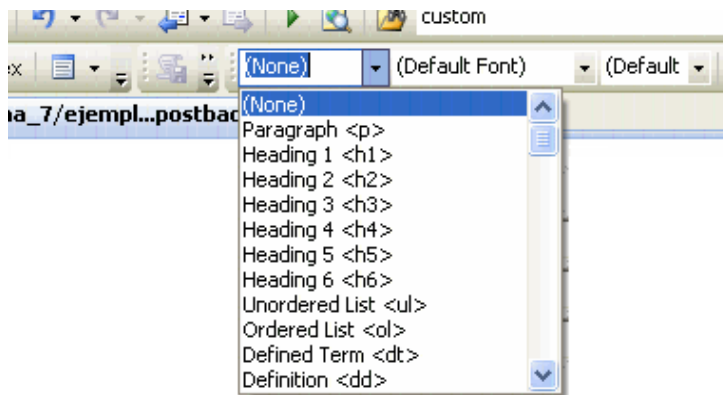
El IDE, estilos, temas y páginas maestras

esta sección vamos a comentar mas el tema de los estilos, como crearlos y administrarlos y como los aplicamos a las páginas.

Los estilos nos van a permitir crear sitios Web coherentes en estilo y sobre todo si alguna vez queremos cambiar una parte de su aspecto basta con cambiar esta hoja de estilos que dará el nuevo formato a todo nuestro Web. Y al final la conclusión que uno saca es que antes hacía "webs de juguete" comparado con el grado de perfeccionamiento que verás en estos temas...

4. Diseñar con estilos

Los diseñadores de sitios web construyen sus páginas con unos determinados formatos y estilo de letra. Estos formatos muchas veces son incoherentes y no se respetan en todas las páginas del sitio web. Los estilos que incorpora por defecto el lenguaje HTML son muy pocos y prácticamente no nos ofrecen versatilidad:



La solución para poder realizar un sitio web coherente y uniforme es la creación de los CSS. Los ficheros CSS es simplemente una definición de estilos para nuestra web. Supongamos que definimos estilos para las tablas, títulos, controles, todo queda coherente ya que utilizarán los que hemos diseñado aquí. Ahora supón que venden tu empresa y donde antes era color azul el corporativo ahora es el verde: a cambiar todas las páginas!. Pues no, solo con cambiar estos estilo se actualizan todas las páginas al trabajar con un único y central archivo de estilos CSS. Los estilos nos van a permitir añadir contornos, establecer detalles de los tipos de letra, definir márgenes, colores, ... Ahora veremos una descripción resumida de los estilos, cómo debes utilizarlos y cómo añadir y mantenerlos.

Tipos de estilos

En las páginas podremos utilizar tres tipos de estilos:

- Estilos en línea. Se colocan directamente en las etiquetas HTML
- Estilos en la propia página. Se definen una serie de estilos dentro de la propia página web en la sección "<head>"
- Hoja de estilos externa. Esta es la opción que trabajaremos porque es la mas eficiente al tener todos los estilos centralizados.

4.1 Los ficheros CSS

CSS es un lenguaje que junto con el HTML permite definir los estilos de los distintos elementos de las páginas Web. Se utiliza junto con la instrucción HTML:

```
style=atributo
```

Cuando editamos un control y le ponemos a mano un tipo de letra lo que estamos haciendo es añadir etiquetas con estilos a los elementos de las páginas:

```
<td style="font-weight: bold; font-size:
```

Pero claro no podemos ir acordándonos de los estilos que vamos aplicando a estos elementos de las páginas así que lo mejor es crear una página especial de estilos y decirle a nuestra página que debe cogerlos de ahí. Así todas las páginas utilizarán ese índice de estilos. Pero... ¿porqué en la definición de CSS dice "cascade" es decir que se aplican en cascada? Mira el siguiente ejemplo:

```
<body>

    <h1>Título de mi página</h1>

    <p>Esto es una línea de texto normal debajo del titular</p>

</body>
```

Ahora fíjate si le aplicamos un estilo al elemento de nivel superior "body":

```
<body style="font-family: 'tipodefuelle'">

    <h1>Título de mi página</h1>

    <p>Esto es una línea de texto normal debajo del titular</p>

</body>
```

¿Qué efecto tiene esa etiqueta? Bueno, como deberíamos esperar el estilo de ese tipo de letra se aplica a los dos elementos: título y párrafo. Por tanto se heredan los estilos, de ahí que diga "en cascada". Lógicamente si aplicamos un estilo diferente a ese párrafo no hereda el anterior y se aplicará el explícito para el párrafo. Por tanto los elementos que no tienen su estilo definido heredan el de la etiqueta padre. Para crear este estilo en la hoja de estilos CSS debemos definirlo así:

```
body {font-family: 'tipodefuelle'}
```

Normalmente se definen los distintos estilos como pares de valores: fuente:nombre ó color:número. Por ejemplo:

```
body {font-family: 'tipodefuelle' font-size: 12pt}
```

Otra forma de escribirlo es esta:

```
body {

    font-family: 'tipodefuelle'

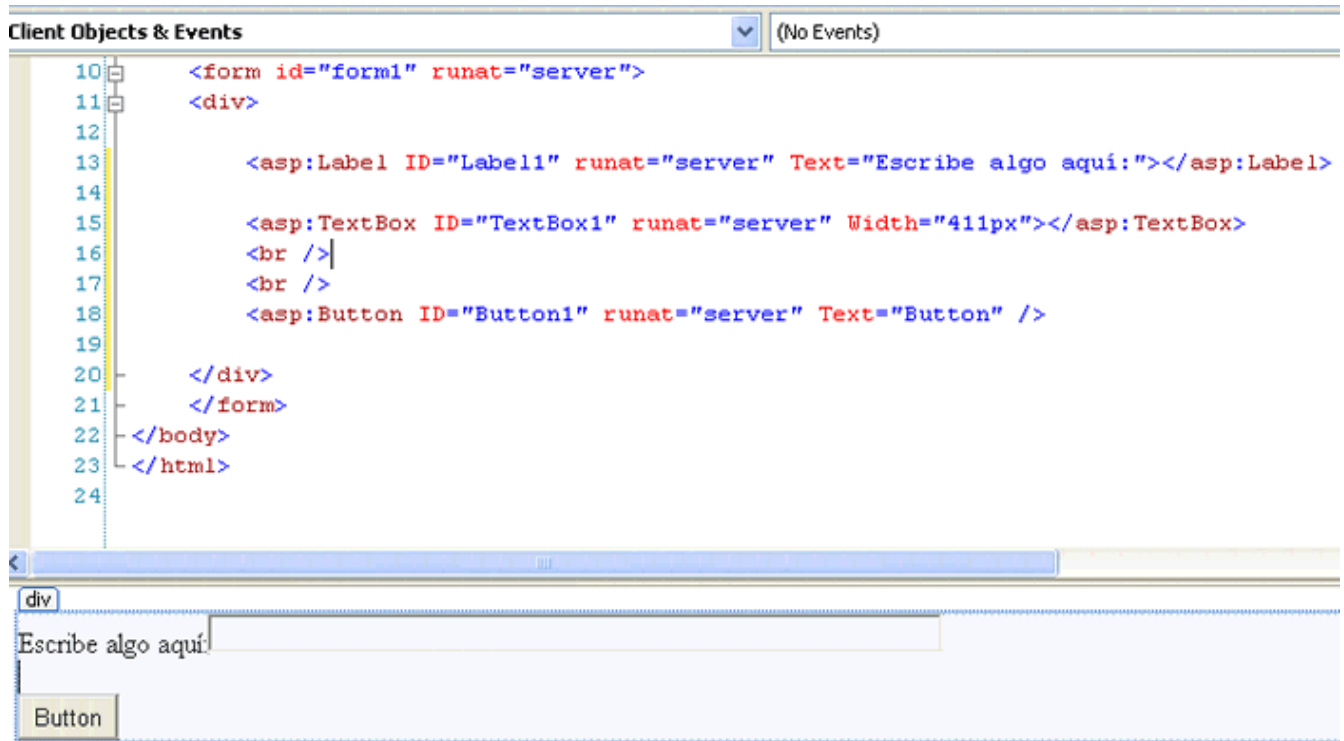
    font-size: 12pt

}
```

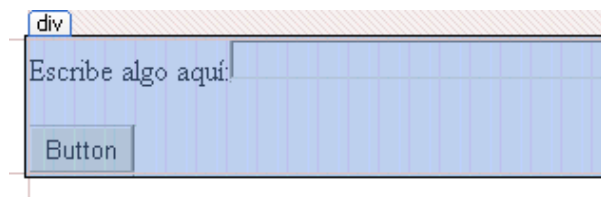
```
color=navy  
}
```

4.2 El generador de estilos

ASP.NET dispone de un práctico y potente editor de estilos que nos generará todas las etiquetas necesarias en ese fichero CSS. Vamos a crear una página nueva y le ponemos unos controles como estos:

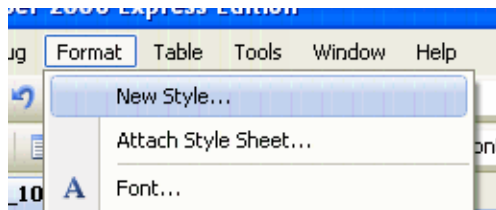


Asegúrate como ves en la parte superior que los controles están dentro de la sección "<DIV>" ya que va a ser la que en este ejemplo nos va a delimitar los estilos. Selecciona ese DIV con el ratón en la vista "Design":

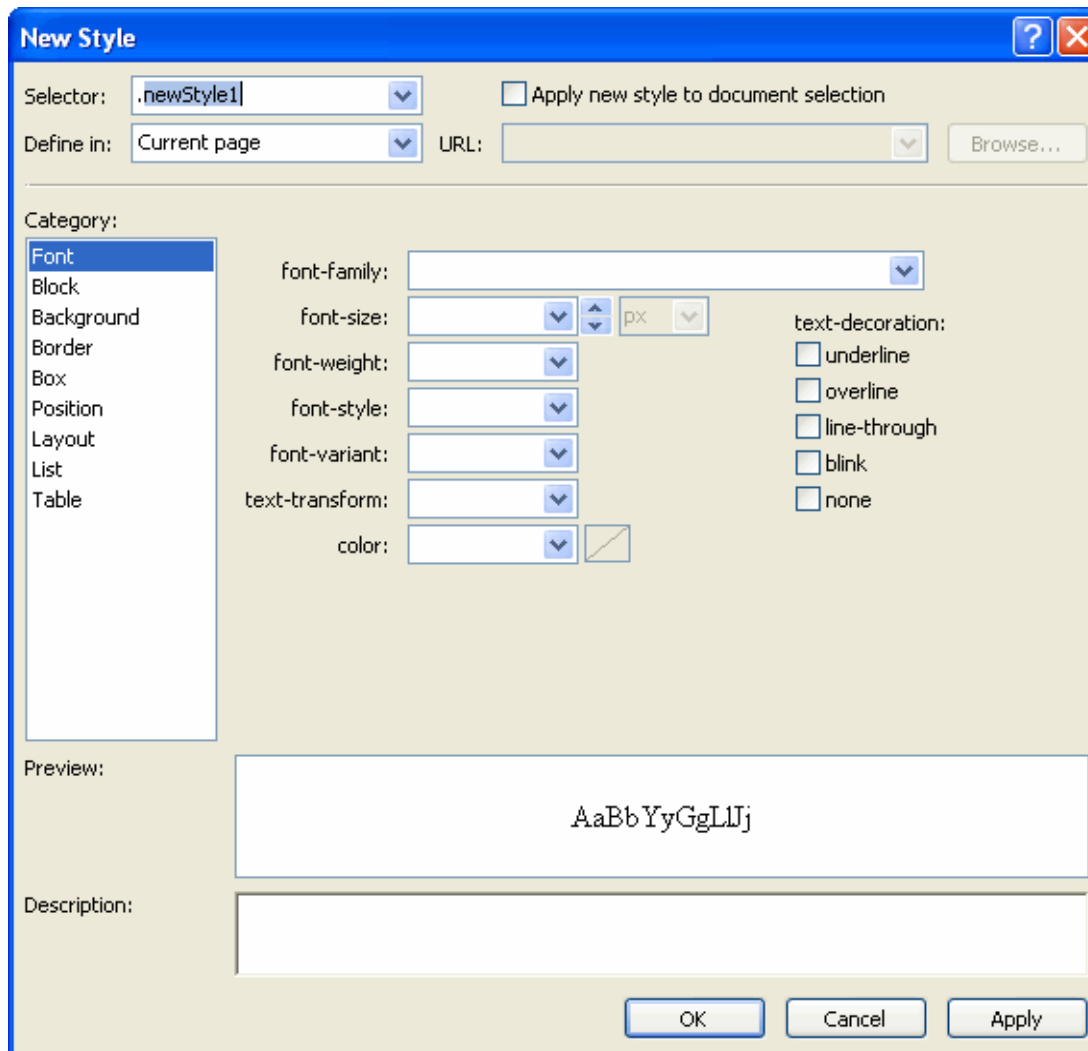


Ahora selecciona la opción de "New Style" del menú "Format":

El IDE, estilos, temas y páginas maestras



Para acceder al generador de estilos:



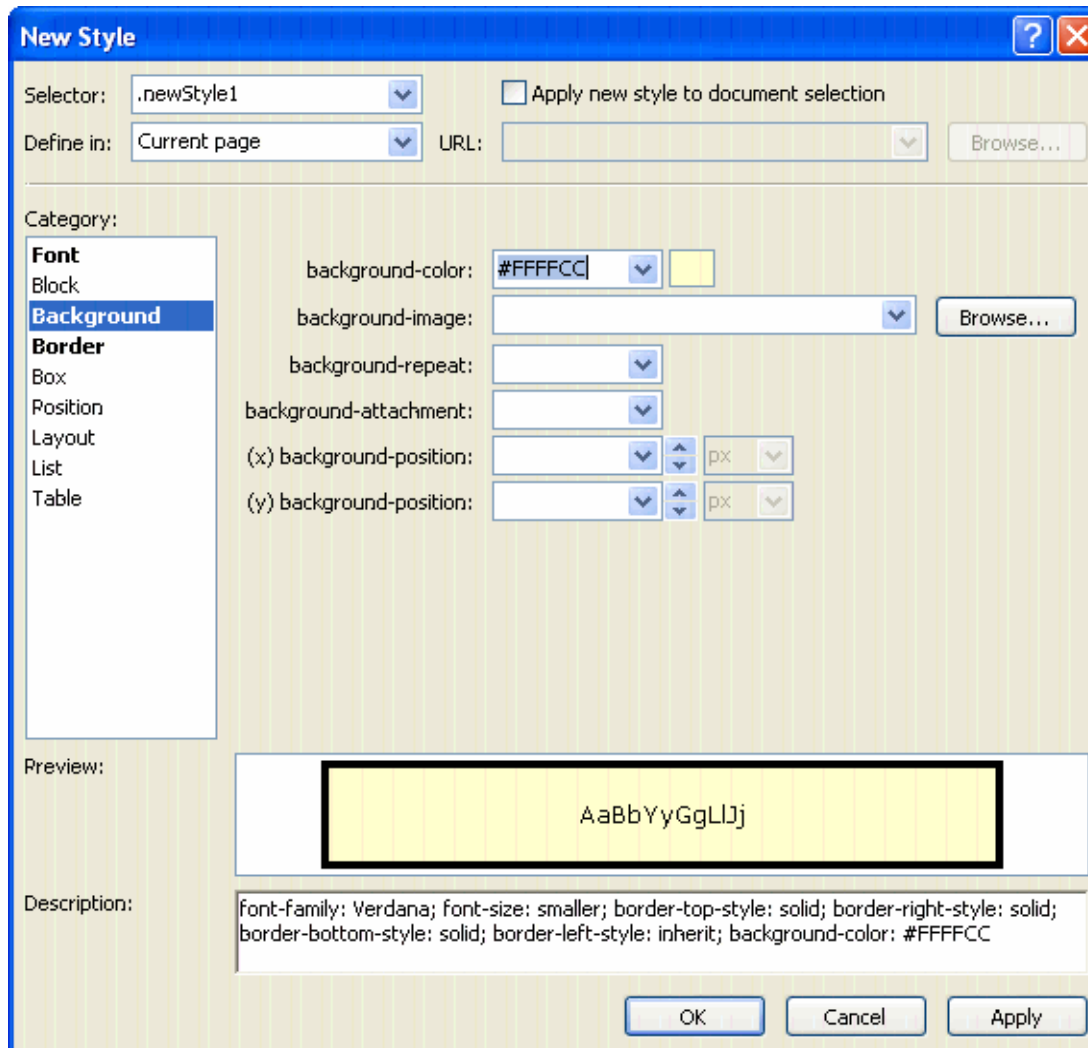
En la primera opción vemos que el "Selector" pone "newStyle1", cámbiala por la opción "(inline style)". En nuestro ejemplo queremos que nos ponga los estilos directamente en el código HTML. Ahora ya tienes la posibilidad de jugar con las opciones para poner la que mas te guste, pudiendo modificar:

- **Tipo de letra.** Podemos seleccionar la familia de la fuente, el tamaño, color y otros detalles.
- **Block.** Permite configurar pequeños detalles de los párrafos, como la altura de las líneas, la alineación, espaciado,...
- **BackGround.** Para seleccionar el color e imagen de fondo
- **Border.** Definiremos los contornos con estilos de líneas, colores, ...
- **Box.** Define los márgenes y espacios en contenedores, como por ejemplo las celdas de una tabla.

El IDE, estilos, temas y páginas maestras

- **Position.** Permite establecer la anchura y altura de los elementos especificando posiciones absolutas. No es recomendable utilizar las posiciones absolutas de los elementos mas que cuando queramos hacer una plantilla o página con un formato fijo, sin redimensionamientos.
- **Layout.** Permite asignar distintos parámetros como la visibilidad, si es flotante, ...
- **List.** El formato para las listas
- **Table.** Formatos para las celdas, espaciado, ...

Seleccionemos un formato, por ejemplo la fuente "Verdana" con un tamaño pequeño "smaller" en nuestro ejemplo, un contorno y un color de fondo:



Como ves en el cuadro de texto de abajo, nos indica las etiquetas HTML que va a colocar en nuestra sección "DIV". Pulsamos "OK" y vemos el resultado:

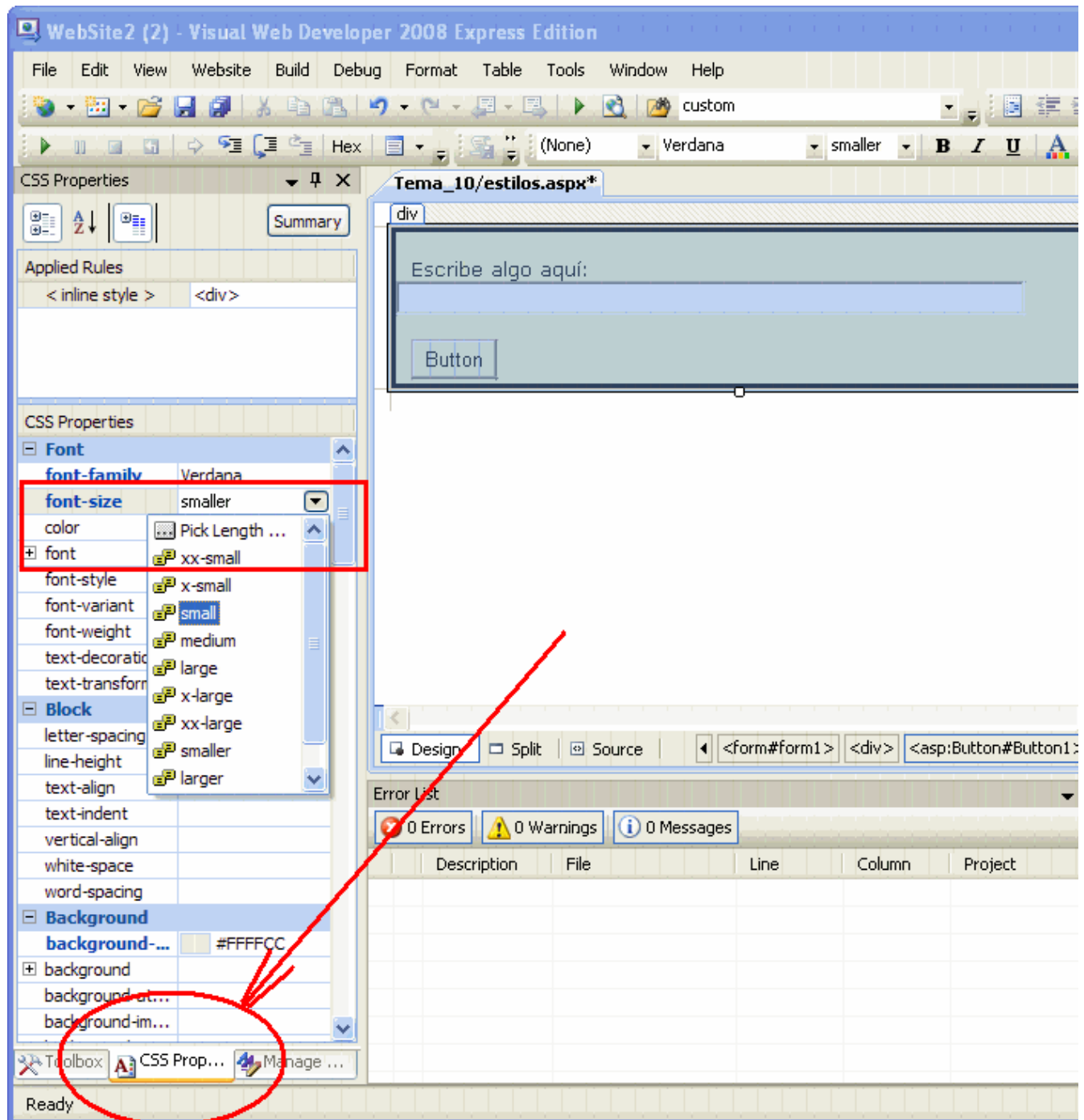


El IDE, estilos, temas y páginas maestras

Y si nos fijamos en el HTML:

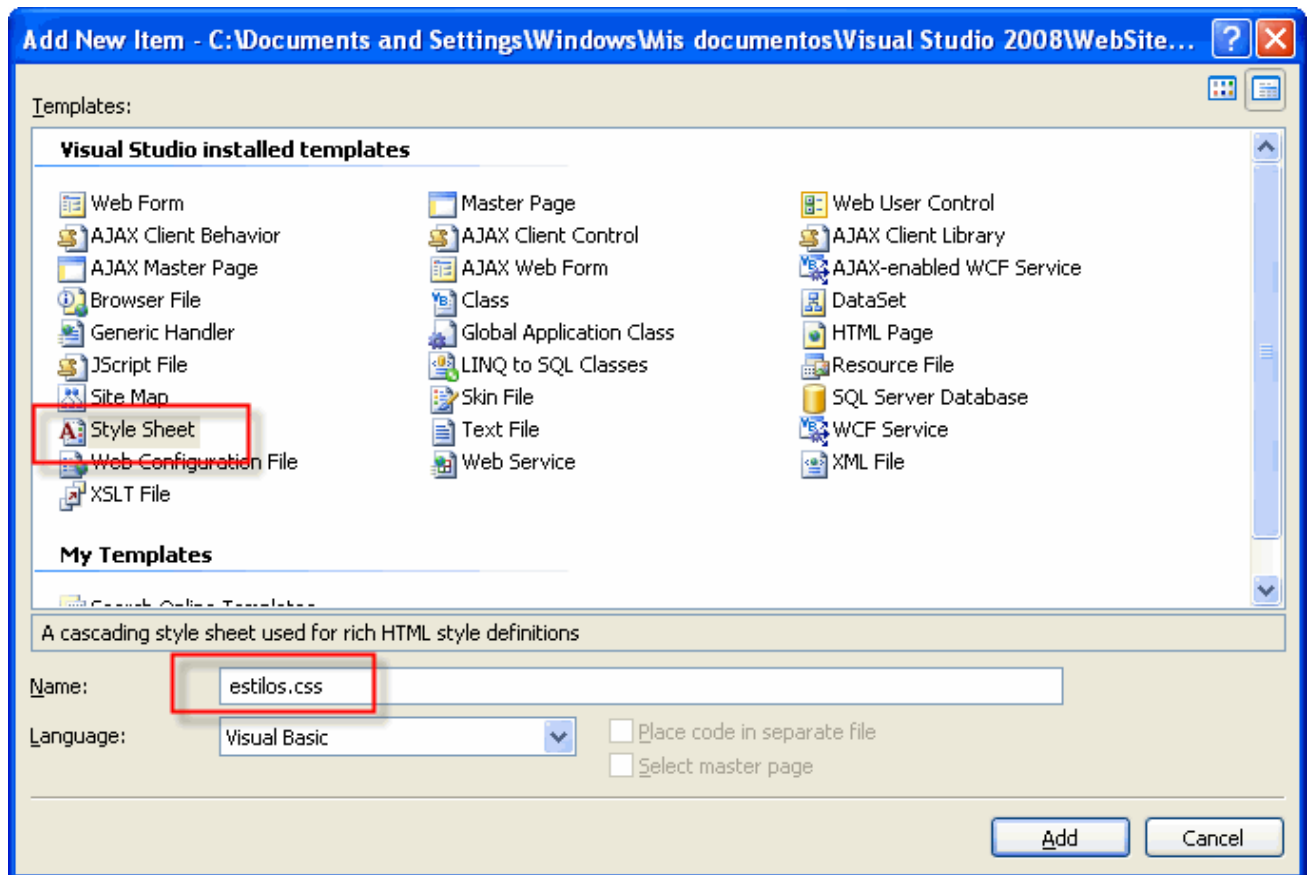
```
<div style="font-family: Verdana; font-size: smaller; border-style: solid;
background-color: #FFFFCC">
```

Le ha puesto las etiquetas de los estilos que le hemos indicado. Una vez que tenemos que trabajar con estilos, lo mejor es que mostremos la ventana de los estilos para poder aplicarlos mas fácilmente:



Crear una página de estilos

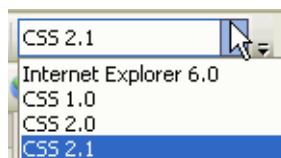
Para crear la página muy fácil, pulsamos como siempre donde queremos añadir la página de estilo y con el botón derecho añadimos la página:



Al crearla se mostrará esto:

```
1 | body {  
2 | }  
3 |
```

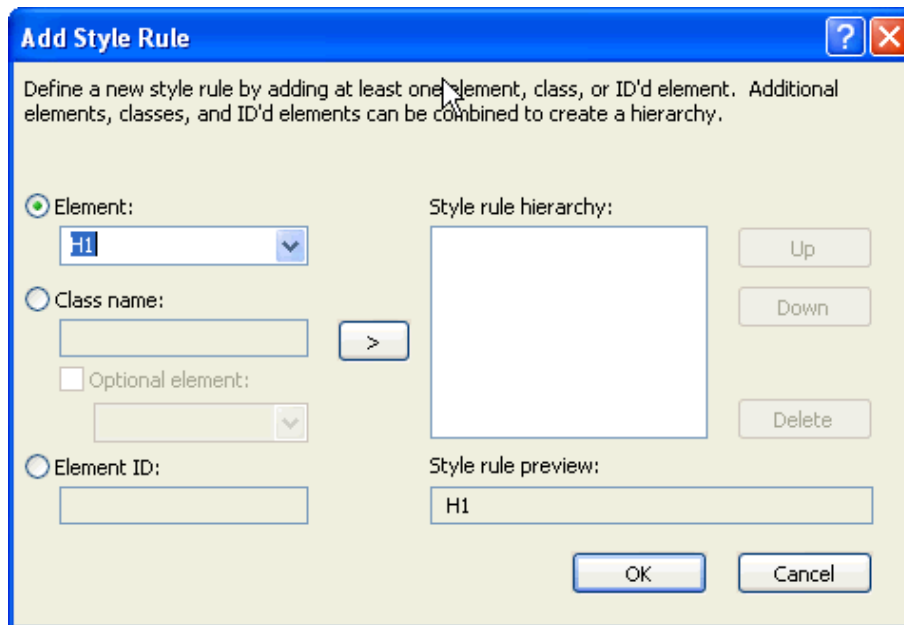
Si te fijas en la parte superior se mostraba un desplegable:



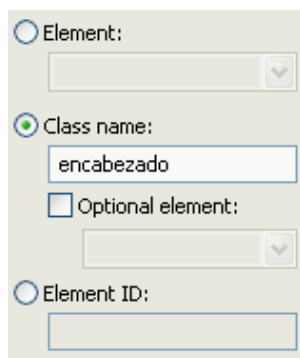
Para indicar la compatibilidad de la hoja de estilos, podemos dejar marcada la predeterminada que nos funcionará bien en la mayoría de los navegadores.

Crear reglas de estilos

Vamos a añadir estilos con una pequeña ayuda: la barra de estilos, que es el mismo sitio donde estaba el desplegable anterior y pulsamos en "Add Style Rule":



Como ves en "Element" tenemos todas las etiquetas HTML posibles. Pero no tenemos que crear estilos en todas las etiquetas... sólo a nuestro gusto. Pero... ¿cómo puedo crear estilos propios? Un ejemplo muy normal es para las celdas, podemos definir un estilo para las celdas normales y otro estilo para las del encabezado. Entonces debemos crear estilos especiales para estos casos... con "class name", por ejemplo:

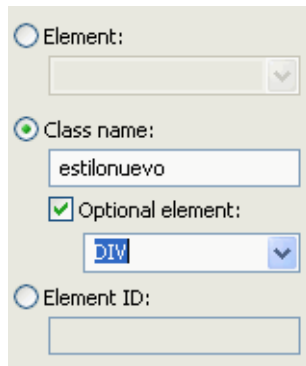


Que produce en la página:

El IDE, estilos, temas y páginas maestras

```
1 body {  
2 }  
3 .encabezado  
4 {  
5 }  
6
```

Podemos asignar también una clase de estilo a una etiqueta HTML y así utilizar esta variante del estilo de la etiqueta HTML en otras partes de la página. Por ejemplo podemos crear una división <DIV>:



Element:

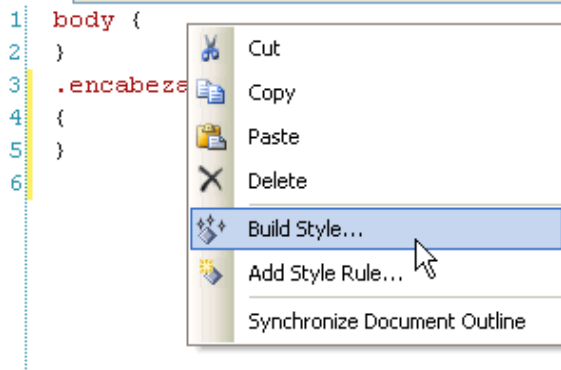
Class name: estilonuevo

☒ Optional element: DIV

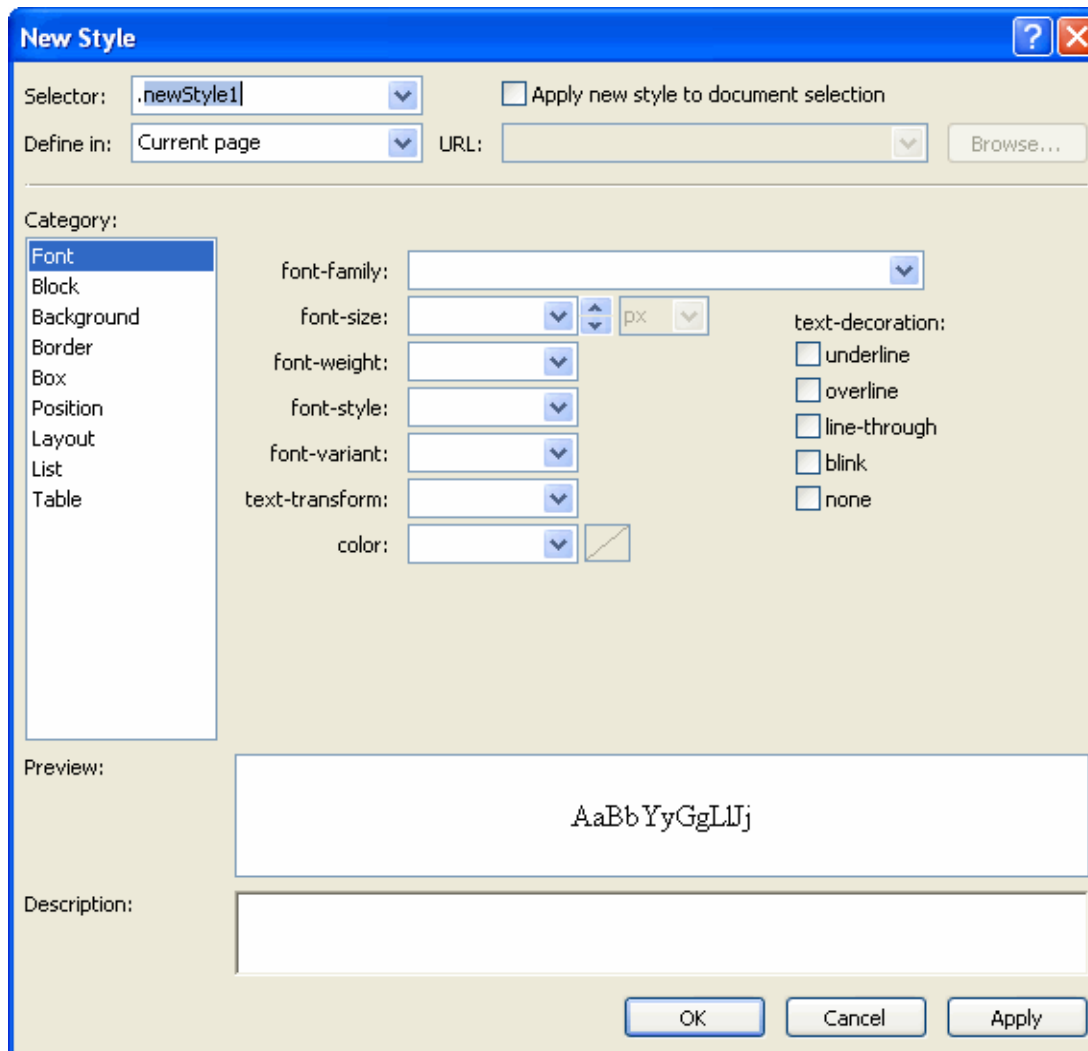
Element ID:

Definir reglas con el generador de estilos

Escribir los estilos es laborioso y además tiene una sintaxis muy estricta así que tenemos una gran ayuda con el generador de estilo para poder autorellenar estas etiquetas. Sitúa el cursor dentro de las llaves que identifican el principio y fin del estilo y pulsa con el botón derecho:



Que nos muestra el ya conocido generador:



Nota Como ves en la pantalla hay opciones de "<Not Set>". Su uso es muy sencillo... imagina que en este valor de "Italics" has puesto un valor y luego lo quieres borrar, la mejor forma de restablecer a su estado original, es decir vacío, es poniendo este valor de no establecido.

Las opciones para ir creando estilos te las dejo que las vayas curioseando, prueba a generar varios estilos en el fondo de la pantalla, titulares, bordes de tablas, ajustes de textos. En sitios webs muy elaborados es normal ver hojas de estilos muy complejas. Date cuenta que esto luego te aporta mucha riqueza de estilos para crear la página. Por ejemplo si abres con el bloc de notas la página de inicio del Web de Microsoft vemos que tiene un enlace a su propia hoja de estilos: <http://i.microsoft.com/h/en-us/r/hp.css> ábrela para que veas la cantidad de ellos que se pueden crear y eso que esta es pequeña...

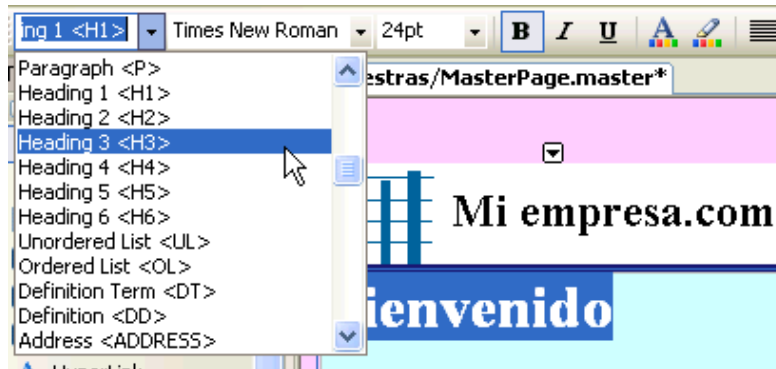
Crea algunos estilos para probarlo ahora... Aplicarlo a las páginas no podía ser algo complicado así que haremos esto. Abrimos la página maestra y arrastramos la página de estilos desde el explorador de soluciones.

Para añadir la hoja de estilos a una página individual que no pertenece a la maestra haremos lo mismo, pero claro, debemos añadirla una a una en esas páginas. La diferencia que veremos en el código HTML es:

```
<link href="../estilos.css" rel="stylesheet" type="text/css" .
```

El IDE, estilos, temas y páginas maestras

Los estilos se aplicarán automáticamente a las etiquetas HTML estándar. Es decir, si definimos antes un estilo para los titulares "H1" automáticamente le aplicará el estilo correspondiente a las páginas afectadas. Para aplicarlos manualmente simplemente marcaremos la zona donde queremos aplicarlo y seleccionamos de la lista:



Pero tenemos un pequeño inconveniente, es una lástima pero le falta algo a esto... los estilos creados con el tipo "class" no aparecen en el sistema de ayuda "intellisense" que son los desplegables que nos muestra los estilos disponibles. Así que no nos queda mas remedio que ponerlo a mano, así que marcaremos el texto o zona donde queremos aplicar esta clase y editar desde la vista de código fuente para escribir el estilo:

```
<span class="miestilo">texto con el formato de miestilo</span>
```

Utilizar estilos DIV

Anteriormente y en otros capítulos comentamos que se pueden crear una especie de divisiones con

```
<DIV> y </DIV>
```

Estas etiquetas son muy interesantes y permiten hacer un cuadro o "división" de texto con un estilo distinto al aplicado en el entorno. Es decir, bloquea la herencia de estilos de las etiquetas y le aplica unos especiales. Un par de etiquetas <DIV>...</DIV> crea un nuevo cuadro o área en el que podemos poner texto, tablas, imágenes o lo que queramos.

Sin embargo si que hereda los estilos de las etiquetas superiores, como <BODY>. Una alternativa para crear un formato manual de <DIV> y así poder aplicarlo tantas veces como queramos es creando un estilo que por ejemplo centre los elementos del cuadro:

```
DIV.centrar
{
    background-color: transparent;

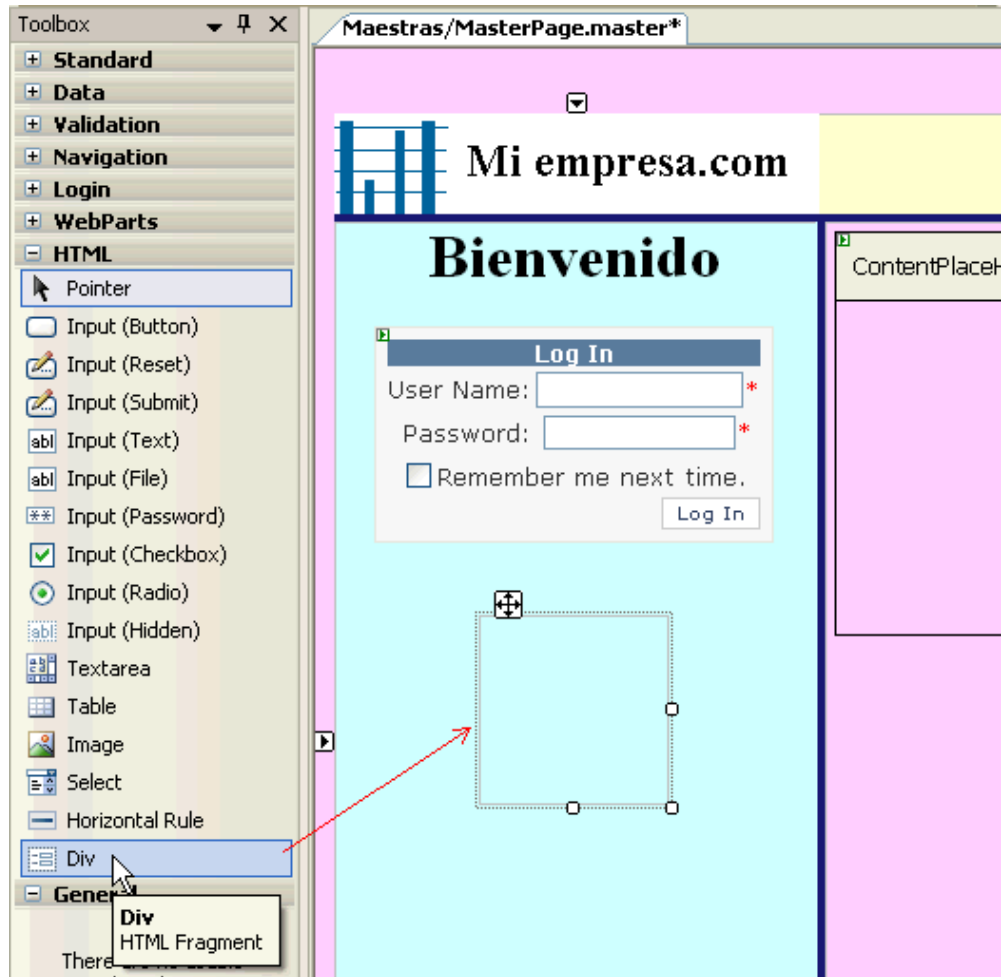
    text-align: center;

    float: none;

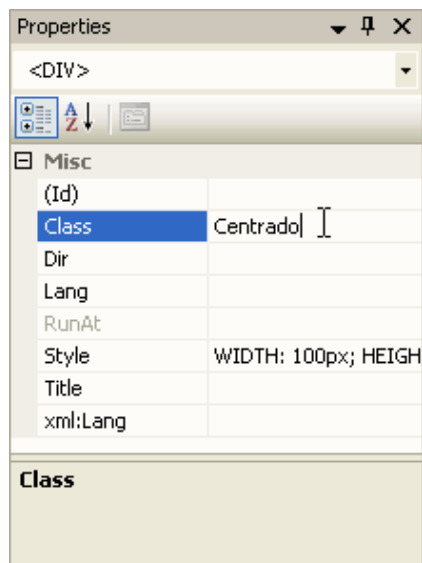
    clear: both
}
```

El IDE, estilos, temas y páginas maestras

Esto nos asegura que este texto quedará centrado pero el texto de alrededor no. Para crear estas zonas las arrastraremos desde la etiqueta HTML del editor:

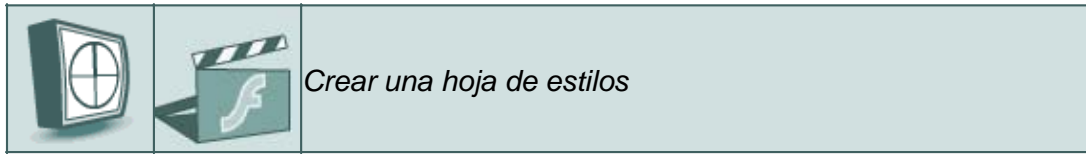


Arrastramos el "DIV" donde queremos y luego en la página de propiedades teniendo esa zona seleccionada elegimos el estilo:



Si arrastramos ahora una imagen a esa zona quedará centrada. Que quedaría de esta forma en HTML:

```
<div class="Centrado">  
    
</div>
```

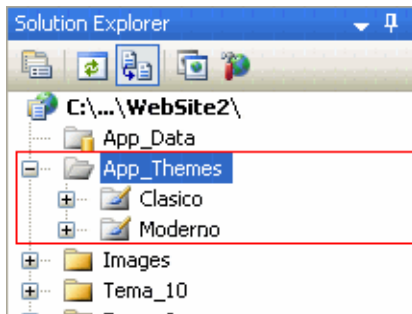


5. Los temas

Vamos a ver por encima una interesante novedad que ya se introdujo en la versión 2.0 de ASP.NET y son los temas. Si te suena, los temas en Windows son un conjunto de estilos: tipos de letra, colores, gráficos, ... que dan un aspecto particular al sistema operativo. En nuestro entorno vamos a disponer de algo parecido para aplicar una serie de estilos a nuestro Web. Sin duda este entorno de desarrollo es lo mejor que se ha creado nunca para edición de código Web porque ningún sistema era tan completo.

Para crear un tema necesitaremos crear una carpeta que lo definirá. Esta carpeta se debe llamar obligatoriamente "App_Themes" que debe estar situada en el primer nivel de nuestra web. Podremos crear varias carpetas dentro de esta y cada una de ellas tendrá su propio estilo.

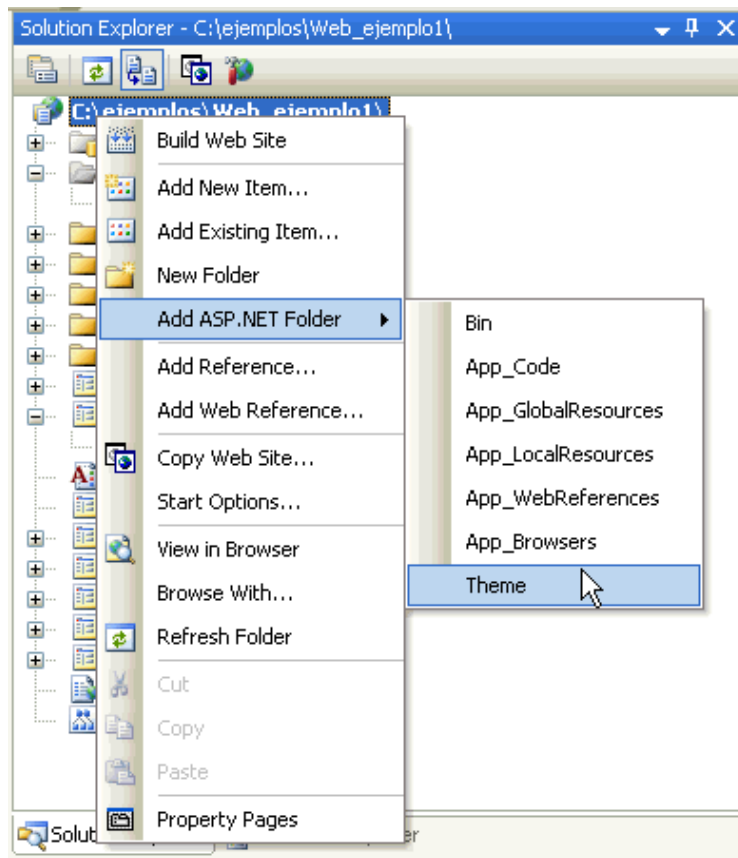
En este ejemplo vamos a crear dos temas, uno clásico y otro moderno. De momento simplemente creamos dos carpetas con esos nombres dentro de la que hemos dicho antes:



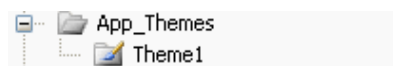
Aunque tengamos definidos varios estilos, sólo habrá uno de ellos activo a la vez. Ahora debemos añadir un fichero de un "skin" que es una piel o textura para nuestros elementos del web. Un fichero "skin" es un fichero de texto con una serie de etiquetas que veremos mas adelante. Veamos ahora paso por paso como crear los temas:

5.1 Crear temas

El primer paso que tenemos que realizar es añadir un tema a nuestro sitio Web así que vamos a hacer lo siguiente. En el explorador de soluciones haz clic en:



Que nos creará esta carpeta y archivo:

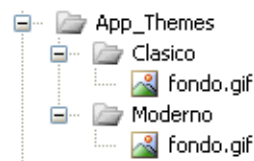


Un tema incluye:

- Imágenes. Por ejemplo para los iconos de navegación y de fondos de pantalla
- Hojas de estilo en cascada CSS. Definirá los textos, colores y otras propiedades definidas por el generador de estilos
- Pielés o "Skins". Se aplican al aspecto de los controles, por ejemplo de los botones.

Imágenes

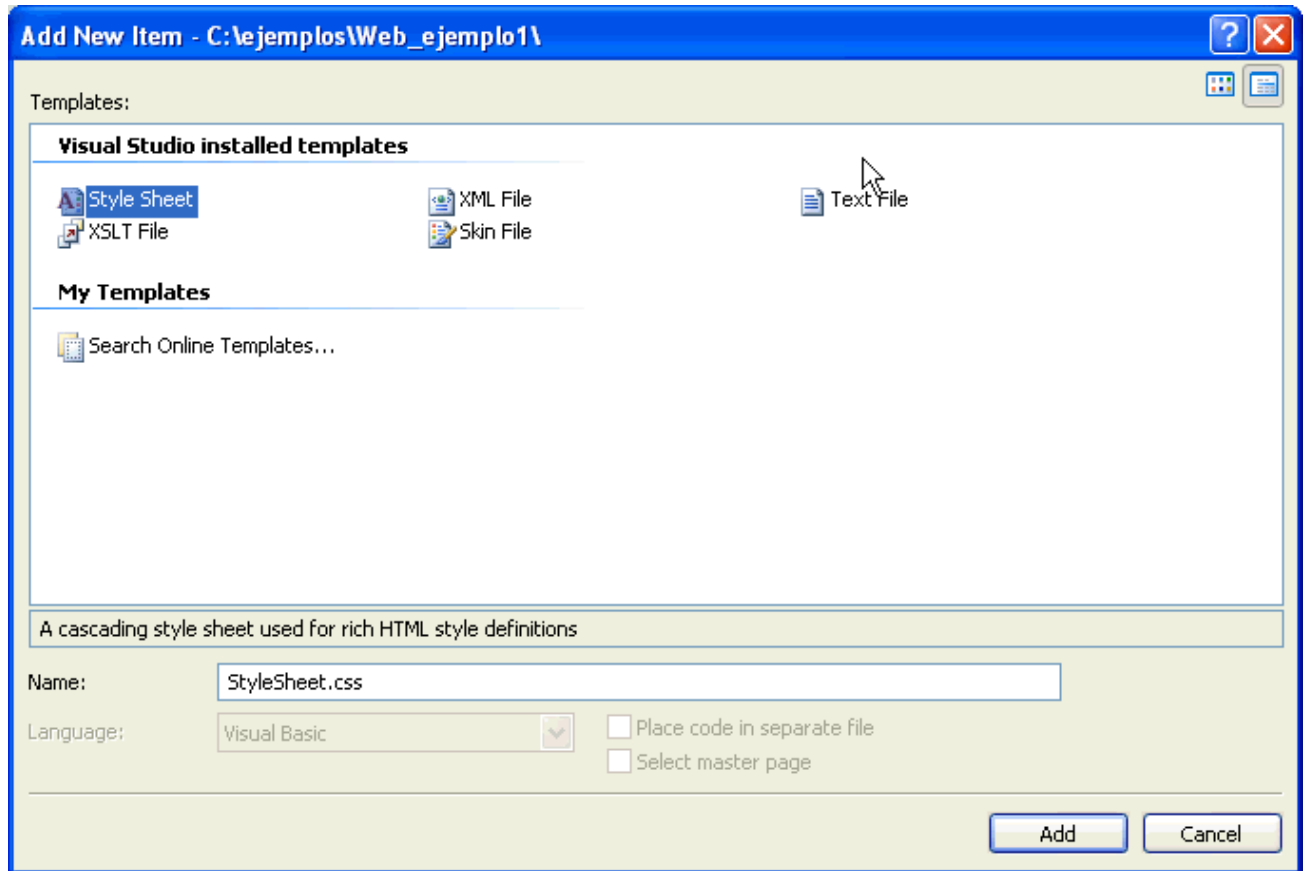
Podemos utilizar cualquier tipo de imagen para las navegación de las páginas, fondos y donde queramos utilizar. Todas estas imágenes deben estar en esta carpeta del tema. Vamos a crear dos carpetas con dos temas distintos y pondremos dos gráficos que harán de fondo en cada una de ellas:



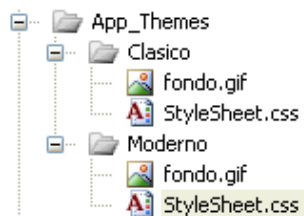
Como ves le he puesto nombres a los dos temas que he creado.

Crear la hoja de estilos

Para dar a nuestras páginas un tipo de letra, color, tamaño, bordes, fondos y otros estilos crearemos una hoja de estilos. Si ya dispones de una puedes copiarla a esa carpeta y sino puedes añadir un nuevo elemento de tipo hoja de estilos como ya conoces:

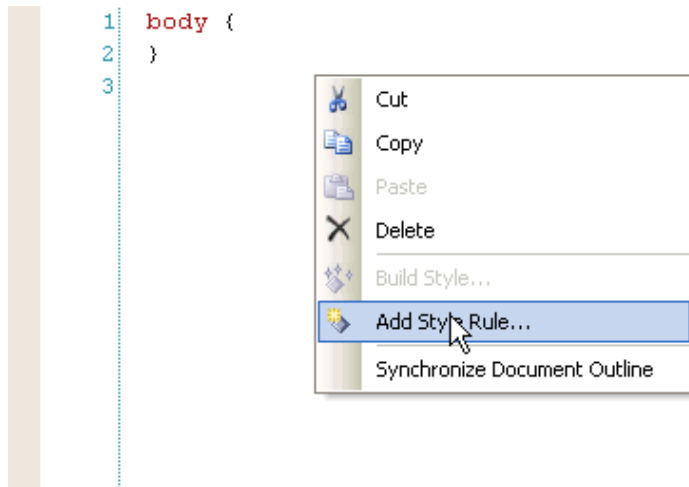


Que quedará:

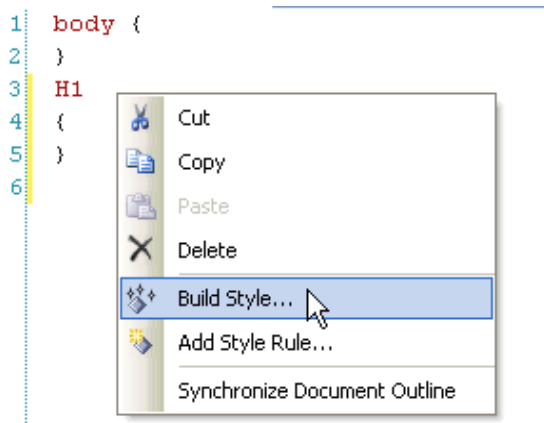


Si no tienes estilos puedes abrir las hojas y seleccionar "Add Style Rule":

El IDE, estilos, temas y páginas maestras



Como ya vimos en otros capítulos. Una vez creado si hacías clic dentro de las llaves { } podías ejecutar el generador de estilos:



Y así poner completar todos los estilos de tu sitio Web. Yo dejaré uno vacío y el otro con todos los estilos de tipos de letra con "verdana" y en color azul.

Ojo, en los dos debemos editar el "body" para ponerle el fondo gráfico:

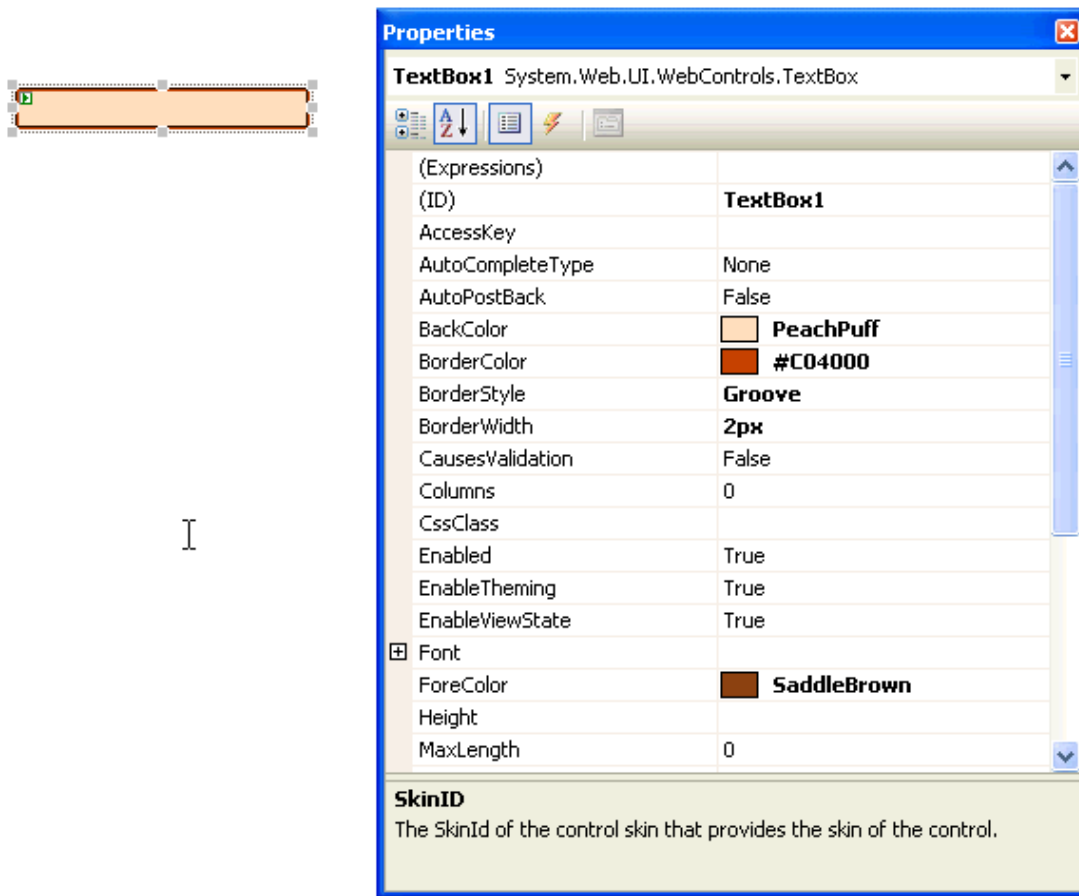
```
1 body
2 {
3     background-image: url(App_Themes/Clasico/fondo.gif);
4     font-family: Arial;
5 }
6
```

Crear las "pieles" o "skin"

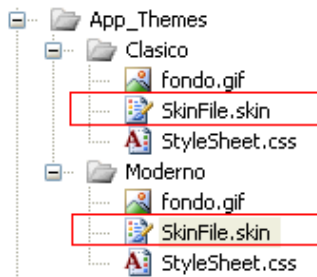
La mejor forma de crearlos es a partir de un formulario. Haz estos pasos: crea un página de formulario "web form" pero que no esté enlazada con una maestra, es para que nos sirva de apoyo para este diseño.

Luego pon un control de cuadro de texto y define las propiedades que quieras para él:

El IDE, estilos, temas y páginas maestras



Cuando hayas terminado con el control pegaremos ese código HTML es nuestra página de "pieles" del tema. Primero vamos a añadir este tipo de página en los temas:



Ahora nos vamos al código HTML de la página anterior, marcamos toda el código que identifica al control:

```
<asp:TextBox ID="TextBox1" runat="server" BackColor="PeachPuff" |
    BorderColor="#C04000" BorderStyle="Groove"
    BorderWidth="2px" ForeColor="SaddleBrown" Style="z-index: 100;
    left: 64px; position: absolute; top: 72px"></asp:TextBox>
```

Abre la página de las pieles creada antes y pega este control:

El IDE, estilos, temas y páginas maestras

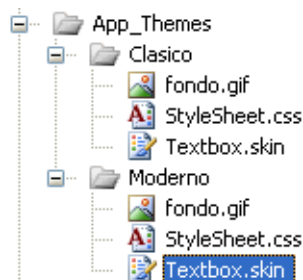
```
1  %--
2  default skin template. The following skins are provided as examples on
3
4  . Named control skin. The SkinId should be uniquely defined because
5    duplicate SkinId's per control type are not allowed in the same theme.
6
7  asp:GridView runat="server" SkinId="gridviewSkin" BackColor="White" :
8    <AlternatingRowStyle BackColor="Blue" />
9  /asp:GridView>
10
11 . Default skin. The SkinId is not defined. Only one default
12   control skin per control type is allowed in the same theme.
13
14  asp:Image runat="server" ImageUrl="~/images/image1.jpg" />
15  -%>
16
17  asp:TextBox ID="TextBox1" runat="server" BackColor="PeachPuff"
18    BorderColor="#C04000" BorderStyle="Groove"
19    BorderWidth="2px" ForeColor="SaddleBrown"
20    Style="z-index: 100;left: 64px; position: absolute; top: 72px">
21  </asp:TextBox>
```

Elimina esos comentarios porque es información innecesaria. Puesto que estas definiciones son solo para el aspecto visual nos sobra toda la información que tenga sobre programación, por ejemplo si fuera un "treeview" tendríamos que quitar la información sobre el origen de datos "Source" y en caso del "textbox" debemos quitar el nombre "ID" que nos ha puesto cuando lo hemos editado para quedar de esta forma:

```
1  <asp:TextBox runat="server" BackColor="PeachPuff"
2    BorderColor="#C04000" BorderStyle="Groove"
3    BorderWidth="2px" ForeColor="SaddleBrown"
4    Style="z-index: 100;left: 64px; position: absolute; top: 72px">
5  </asp:TextBox>
```

Tiene su lógica porque como hemos dicho, esta sección es solo para estilos y no para temas de programación, total no es mas que un copiar-pegar sencillo quitando las etiquetas que les identifican o les proporcionan información externa.

Por último debemos cambiar el nombre es estas pieles para llamarlas como el control que van a dar formato:



Nombres de las "pieles"

Esto es bastante interesante, podemos nombrar de dos formas a estas pieles:

- pieles por defecto. Aplica el aspecto definido a todos los controles del mismo tipo
- pieles con nombre. Aplica el aspecto a los controles que tengan la propiedad "SkinID" igual que la definida.

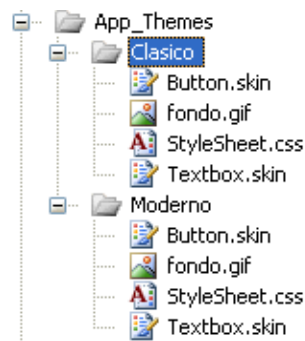
Para crear el primer tipo nos basta con no ponerles esa identificación "SkinID", se aplicará a todos los controles de ese tipo excepto a los que tengas específicamente un nombre de piel distinta. Vamos a hacer un ejemplo, crea dos botones en el que uno tenga algún valor distinto, por ejemplo el color del fondo:



Coge el código de la hoja:

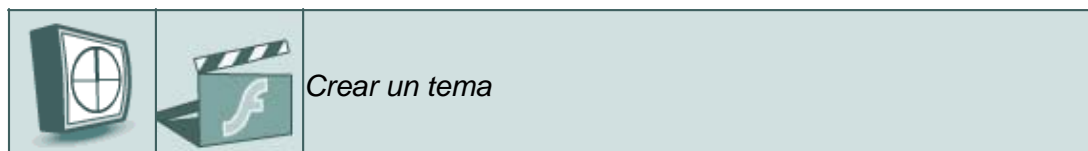
```
<asp:Button ID="Button1" runat="server" Text="Button" />
<asp:Button ID="Button2" runat="server" Text="Button" BackColor="#FFEOCO" />
```

Como ves la diferencia es la propiedad "BackColor". Ahora crea una página de pieles y copia este código pero sin los valores ID (ni texto del botón Text="Button") como dijimos antes:



Cuando pongas el código a uno de ellos le pondremos una etiqueta "SkinID" con un nombre para así distinguirla de la otra, con lo que quedará "button.skin":

```
1 <asp:Button runat="server" />
2 <asp:Button runat="server" SkinID="botoncolor"
3   BackColor="#FFEOCO" />
```



5.2 Utilizar temas en las páginas

Nuestro sitio Web ya estará aplicando estos estilos y pieles, si queremos que algún elemento, como los botones tengan el otro estilo definido lo indicaremos en su propiedad "SkinID":

El IDE, estilos, temas y páginas maestras

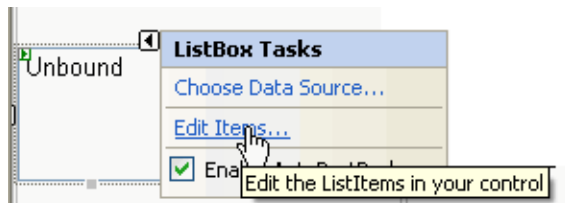
Ahora debemos crear una página para que el usuario seleccione el tema. Crearemos una página de tipo "webform", como es para usuarios autenticados la pondremos en la carpeta de usuarios y la llamaremos "elegirtema.aspx".

La idea es poner unos textos de ejemplo en la pantalla y un cuadro de lista donde el usuario seleccionará un tema. Cuando lo haga lo aplicaremos a toda la página con un método así que escribiremos algo de código. Por ejemplo:

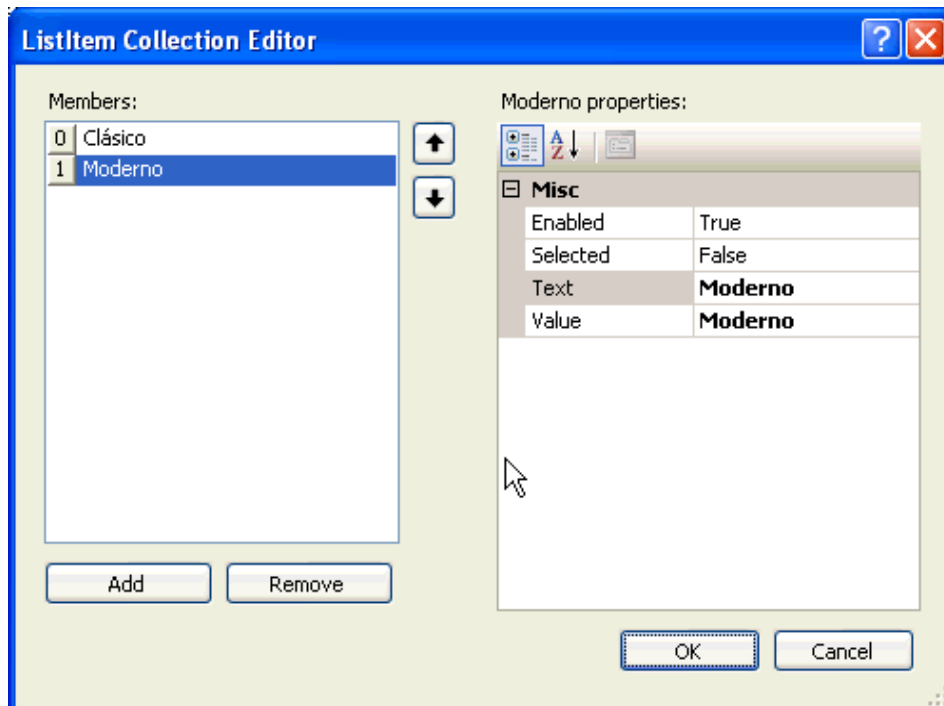
Selecciona un tema	Ejemplo
	<p>Título 1</p> <p>Este es un ejemplo con el tema elegido. Este es un ejemplo con el tema elegido. Este es un ejemplo con el tema elegido. Este es un ejemplo con el tema elegido</p> <p>Título 2</p> <p>Título 3</p> <p><input type="text"/> <input type="button" value="Button"/> <input type="button" value="Button"/></p>

Es solo una tabla con dos files y columnas y con unos cuantos controles y estilos normales. Atento a los botones, en uno de ellos le he indicado el "skinid" del botón de color.

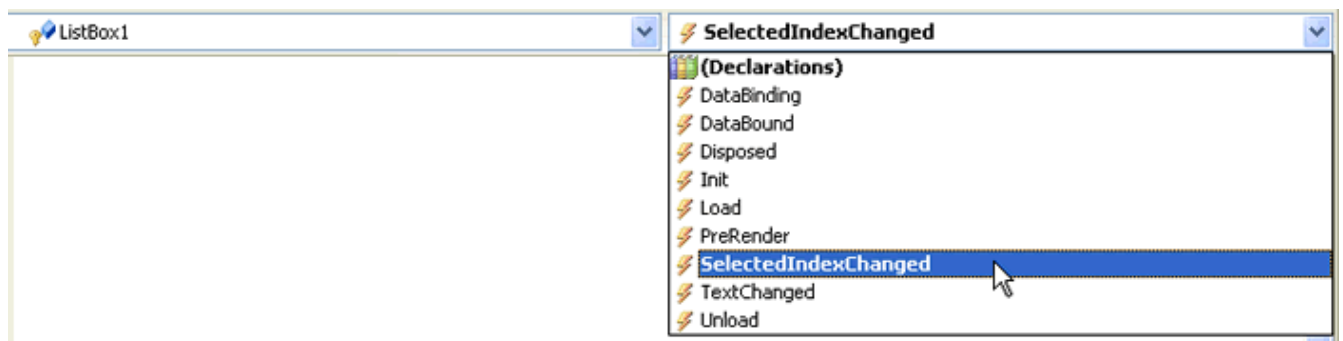
Vamos a añadir un cuadro de lista en la primera columna de la segunda fila y le añadimos unos elementos desde "Edit Items":



Hemos activado el "Enable postback" para que al seleccionar un elemento haga un "page_load"



En la propiedad "Text" puedes poner lo que quieras pero lógicamente en la propiedad "Value" debes poner lo mismo que los temas definidos. Para almacenar el valor debemos irnos a este evento del cuadro de lista que se disparará en cuanto seleccionemos un elemento:



Y ponemos el código:

```
Protected Sub ListBox1_SelectedIndexChanged(ByVal sender As Object,
    Session("tema") = ListBox1.SelectedValue
End Sub
```

Vamos a utilizar una variable de tipo "session" para almacenar la selección del usuarios. Cuando se seleccione una u otra nos aplicará el correspondiente tema. Pero faltan algunas cosas todavía. Al cargar la página debemos actualizar ese cuadro de lista porque debemos indicarle el elemento activo, es decir, el tema que está ahora aplicado. Así que escribimos lo que suponíamos que es en la propiedad del cuadro de lista le decimos el tema del perfil:

El IDE, estilos, temas y páginas maestras

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As  
    If Not Page.IsPostBack Then  
        ListBox1.SelectedValue = Session("tema")  
    End If  
End Sub
```

Bien, al cargar la página nos aplicará el tema... pero aun falta un detalle. Estos temas son hojas de estilos al fin y al acabo y se deben cargar antes que el "page_load" porque al llegar a ese estado ya está pintando la página así que debemos aplicarlo en un estado anterior que es "page_preinit" que significa que en los preparativos de la creación de la página utiliza ese tema, así la cargará asociando las correspondientes hojas de estilos. Le ponemos un pequeño control de que le aplique un tema por defecto si no hay ninguno seleccionado.

Lo mejor es que hubiéramos creado un tercer tema llamado "predeterminado" y así haberlo aplicado ahora pero no pasa nada, utilizaremos uno de los creados:

```
Protected Sub Page_PreInit(ByVal sender As Object, ByVal e  
    If Session("tema") = "" Then  
        Page.Theme = "Clasico"  
    Else  
        Page.Theme = Session("tema")  
    End If  
End Sub
```

Probamos ya la página:



Y nos ha aplicado los estilos definidos. Si seleccionamos otro elemento de esa lista nos actualizará completamente el tema a toda la página.

Otras formas de aplicar temas

Hasta ahora hemos aplicado la instrucción "Page.theme" para aplicar un tema a una página pero tenemos también otra posibilidad:

```
Page.StyleSheetTheme=tema
```

El primero de ellos nos aplicaba el estilo sobrescribiendo los valores de estilo locales. En este nuevo caso el valor establecido en la página para una propiedad tiene preferencia sobre el valor del archivo de máscara en caso de haber coincidencia. Justo al contrario que en el caso habitual.

Una posibilidad poco frecuente es utilizar ambos atributos (Theme y StyleSheetTheme) simultáneamente. En este caso la precedencia lo que ocurre es lo siguiente:

- Se aplican los atributos del tema especificado en StyleSheetTheme.
- Se aplican los atributos definidos en la página que sobrescriben a los del paso anterior en caso de coincidencia.

El IDE, estilos, temas y páginas maestras

- Se aplican los atributos del tema especificado en Theme que en caso de coincidencia tienen por tanto preferencia sobre los anteriores.

Definir un tema para el web

Bien, y la forma mas sencilla de configurar nuestro Web es decirle en alguna parte que debe utilizar un tema, para esto nos iremos a nuestro fichero de configuración "web.config" para definir:

```
<system.web>
  <pages theme="Mi_tema" />
</system.web>
```

Nota: Dado que una aplicación Web puede disponer de varios archivos de configuración (uno por cada carpeta) si queremos que la aplicación emplee un tema diferente en diferentes zonas sólo tenemos que guardar todas las páginas de cada zona en una carpeta diferente y crear archivos web.config específicos para indicar el tema en cada una de ellas. Las carpetas que no dispongan de su propio web.config heredarán el tema principal del sitio.

6. Páginas maestras

Cuando visitamos sitios Web vemos que tienen un aspecto continuo, es decir, parece que estamos en una misma aplicación. Por ejemplo, los periódicos como "El Pais" o "El Mundo" tiene varias secciones a los laterales y una parte central variables. SI navegamos dentro de las noticias seguimos viendo ese aspecto de coherencia donde tanto los encabezados como los laterales son iguales.

Si aprendiste a crearlas con ASP.NET 2.0 estás de suerte: son mucho mas fáciles ahora. En la versión de Visual Web 2005 se podían hacer pero eran muy engorrosas, ahora se ha simplificado y son realmente sencillas...

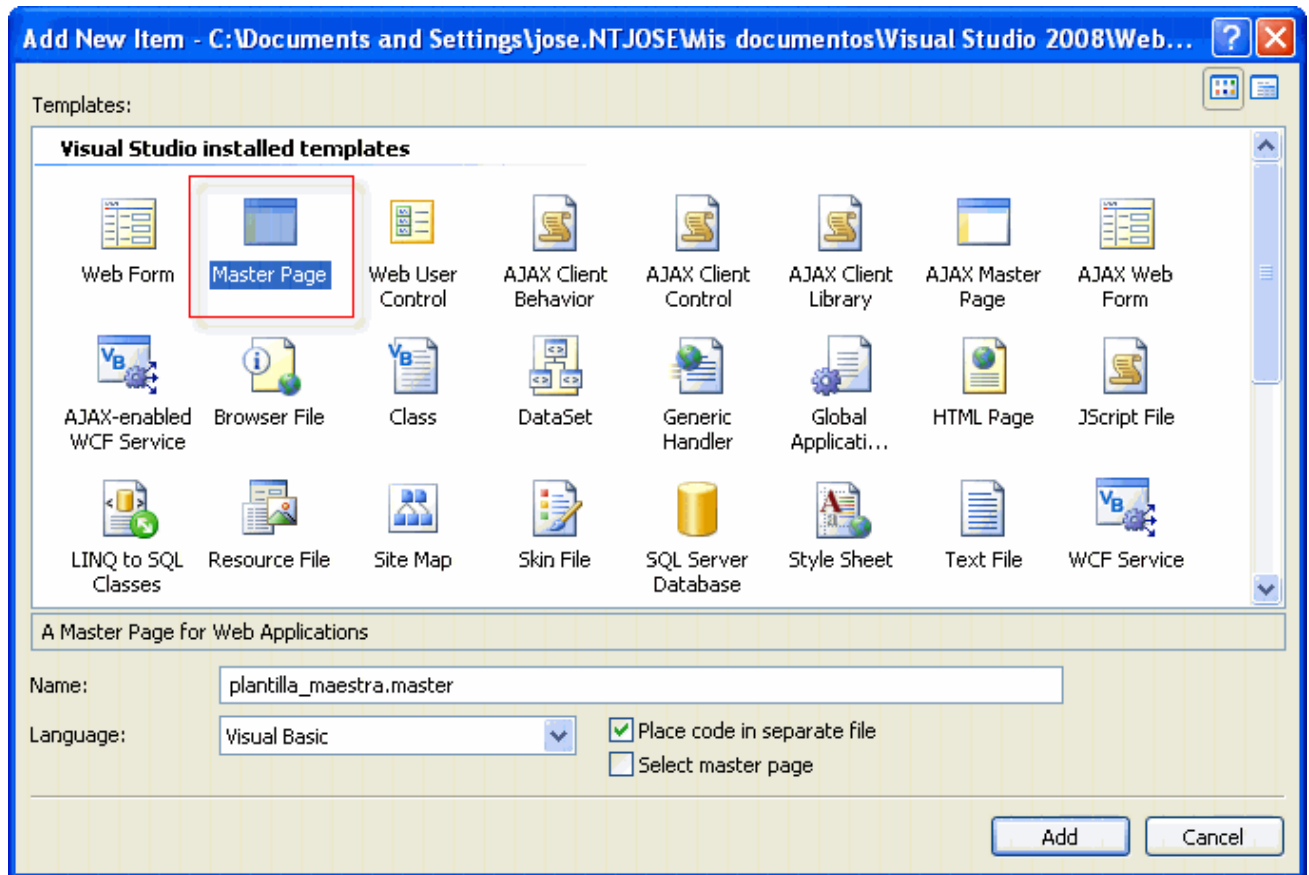
De alguna forma hay una estructura definida en la página donde cada parte está predeterminada. Esto era fácil hacer con los marcos pero no con una página HTML continua. Además los marcos eran engorrosos de mantener ya que cada uno apuntaba a una página web diferente y con estructuras un poco complicadas el funcionamiento se hace complejo. La solución que nos plantea ASP.NET son las páginas maestras que permite crear una serie de plantillas que podremos utilizar en nuestro web de una forma fácil y con resultados realmente buenos como verás ahora...

NOTA: Como te habrás dado cuenta la versión que trata el curso es en inglés pero ya está disponible la española (desde el 15-2-2008, justo el día en que se terminó este curso) así que hemos traducido alguna cosa algo distinta, por ejemplo los "roles" los hemos tratado como grupos cuando la traducción la llama "funciones". En este caso la "page master" las he traducido como páginas maestras pero la traducción pone "páginas principales". Tenlo en cuenta para que sepas que hablamos de ellas. En cualquier caso son muy pocos conceptos y apenas tendrás problemas.

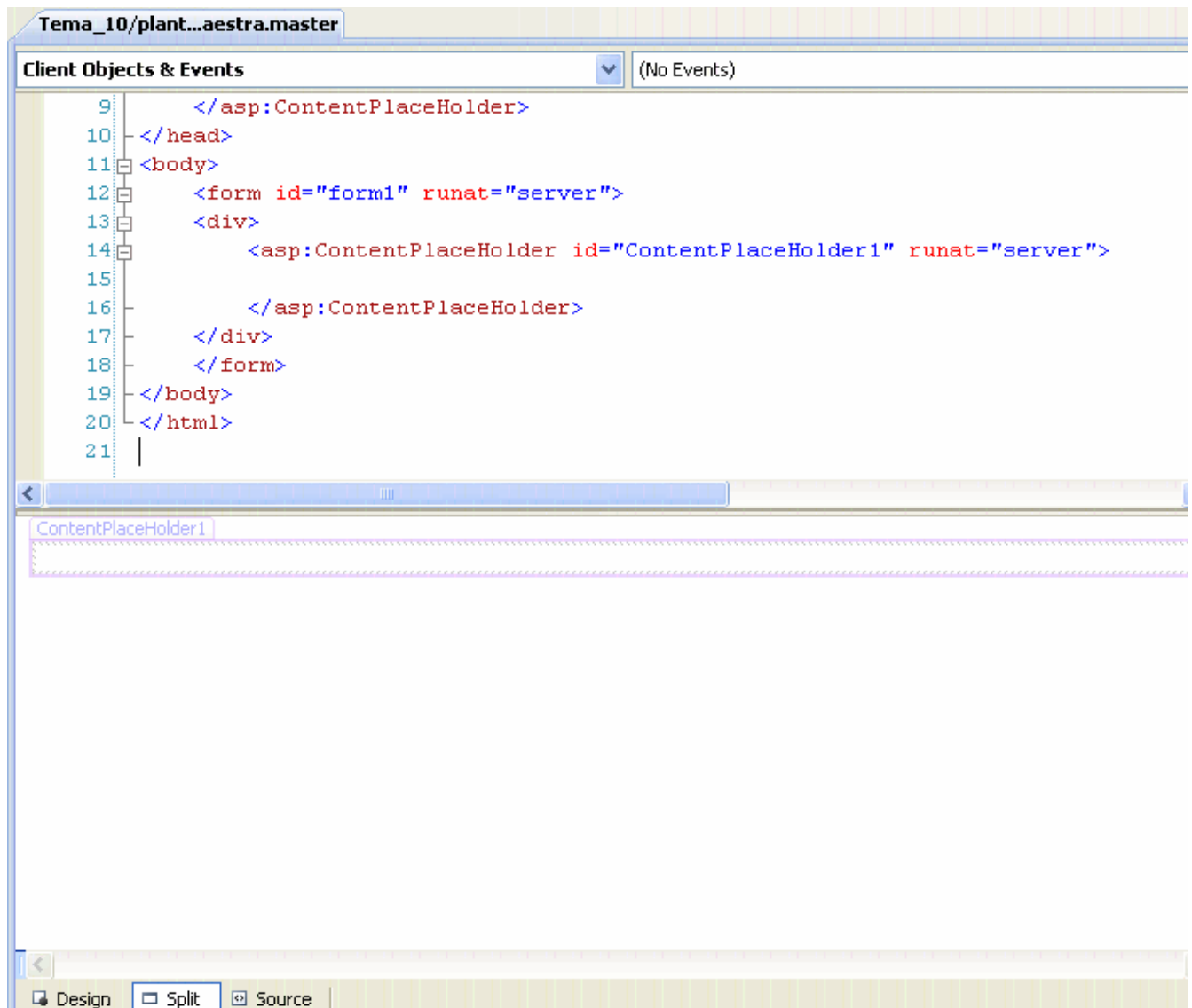
Las páginas maestras (Master Pages) por tanto es una gran forma para proporcionar un aspecto consistente y con diseño que se aplicará a todas las páginas de nuestro sitio Web. Estas páginas nos van a permitir definir por ejemplo un estilo con marcos:

6.1 Una página maestra sencilla con una página de contenidos

Vamos a crear un ejemplo sencillo para que veamos su funcionamiento. Vamos a crear una nueva página de tipo:



Parece que desconcierta un poco porque en lugar de tener un editor visual nos encontramos con que nos muestra un código fuente con una sección "ContentPlaceHolder". Activa la vista mixta de código y página:

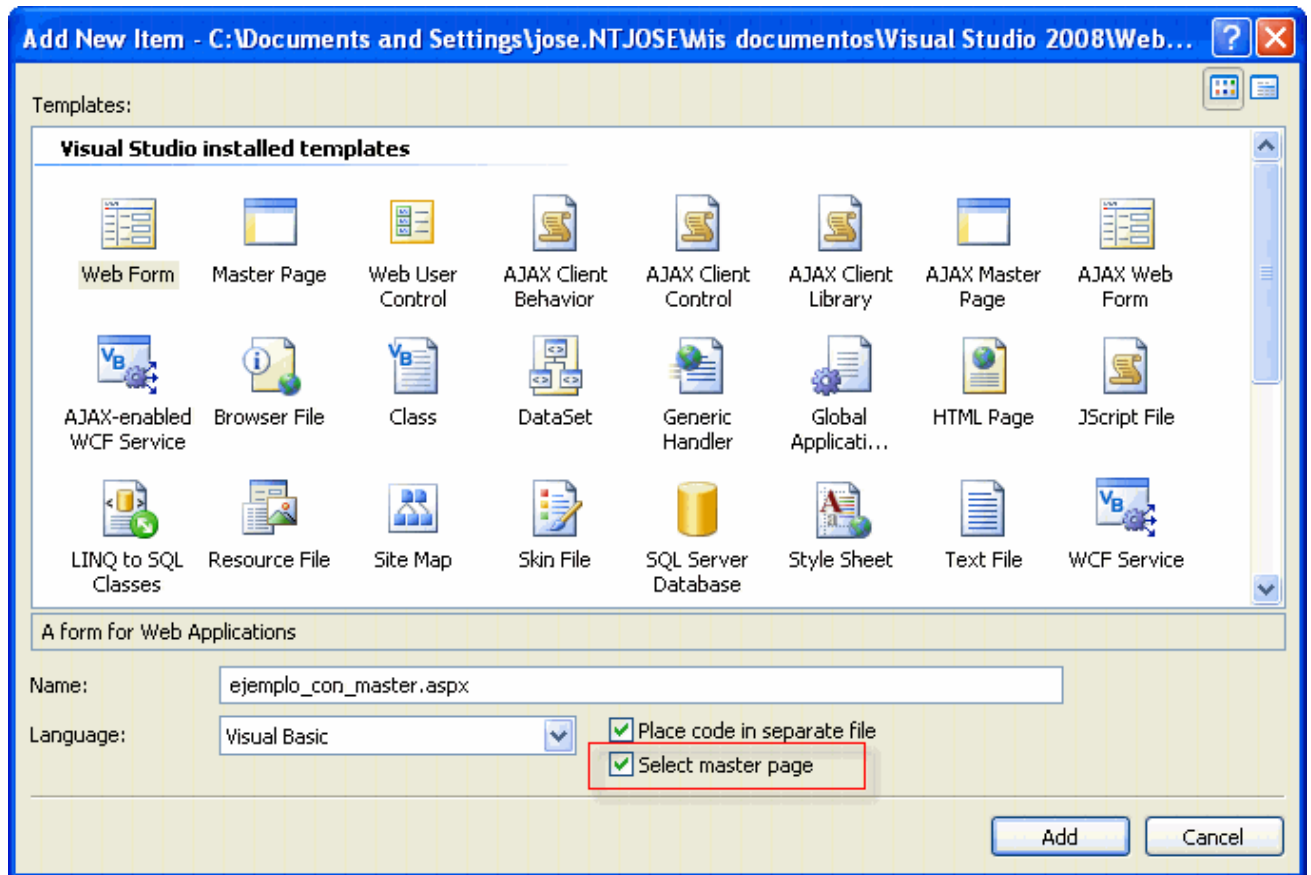


La parte de "ContentPlaceholder" es la única parte que puede cambiar, todo lo demás permanecerá estático. Al crear la página, nos ha creado dos secciones de estas, una en la cabecera, sección "Head" u otra en esta sección de "body". La primera parte contendrá los estilos y datos especiales de las páginas y la segunda tendrá el contenido de la página. Vamos a crear un título y un pie de página en la parte superior e inferior del "ContentPlaceholder":

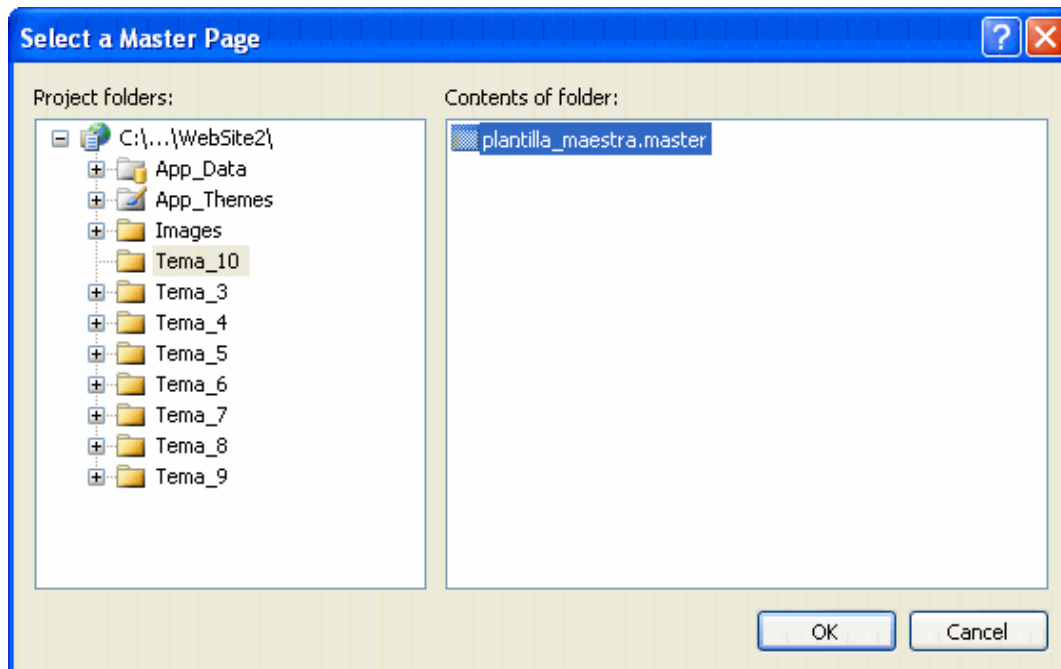


Ya tenemos nuestra página maestra creada. Ahora vamos a crear una página basada en esta página maestra. Crea una página nueva y le indicas que utilice una maestra:

El IDE, estilos, temas y páginas maestras



Al grabarla nos pedirá con que página maestra queremos trabajar:

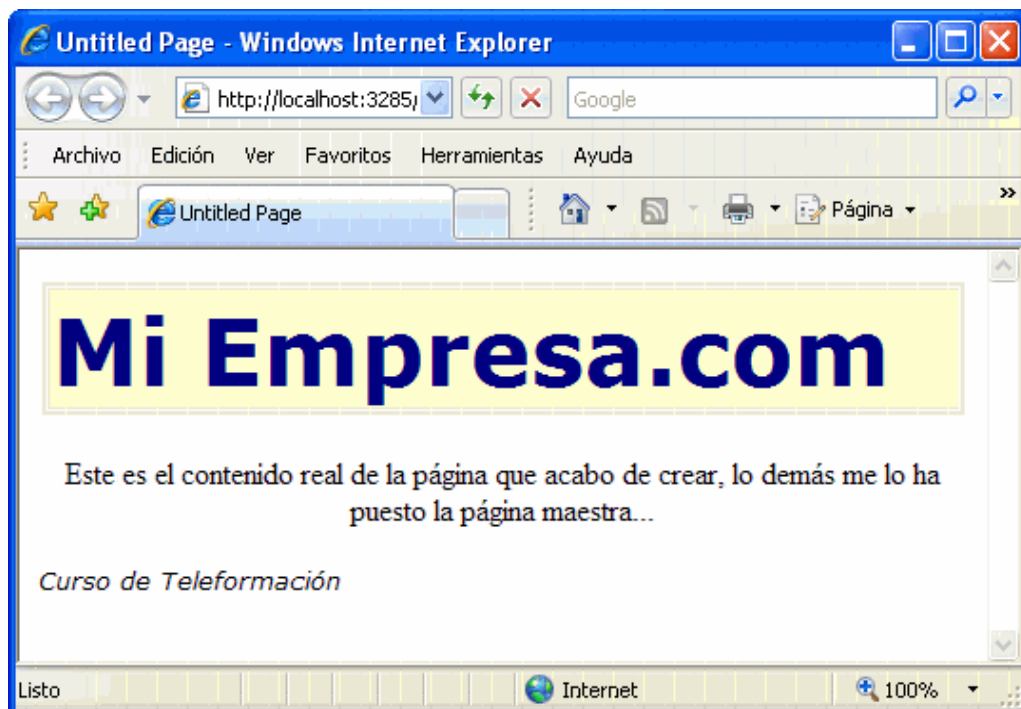


Pon la vista gráfica y verás que:



El aspecto es el siguiente: a pesar de ser una página nueva nos ha incluido todos los aspectos de la página maestra que diseñamos antes. Sólo nos va a dejar trabajar con la zona del centro y verás que al pasar el ratón por la zona de la página maestra el cursor aparece con una señal de prohibido indicando que no se podrá editar esa zona.

Escribimos ahora un texto en la página y la vemos en el navegador:



Conexión de las páginas con las maestras

Veamos ahora el funcionamiento interno de estas páginas. La principal diferencia e estas páginas es que el encabezado empieza con una directiva de "Master" en lugar de "Page". En nuestra página anterior:

```
1 <%@ Master Language="VB" CodeFile="plantilla_maestra.master.vb"  
2 Inherits="Tema_10_plantilla_maestra" %>  
3
```

Ya ves que ahora aparece "Master" en lugar de "Page". El siguiente código interesante es la definición del "ContentPlaceHolder":

```

</body>
<form id="form1" runat="server">
  <div>
    <table class="style1">
      <tr>
        <td style="font-family: Verdana; font-size: 48px; font-weight: bold;">
          Mi Empresa.com</td>
        <td style="width: 100px;">
          <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
          </asp:ContentPlaceHolder>
        </td>
      </tr>
    </table>
    <span class="style2">Curso de Teleformación </span>
  </div>
</form>
</body>

```

Cuando creamos una página de contenido el encabezado es de esta forma:

```

1 <%@ Page Language="VB" MasterPageFile="~/Tema_10/plantilla_maestra.master"
2   AutoEventWireup="false" CodeFile="ejemplo_con_master.aspx.vb"
3   Inherits="Tema_10_ejemplo_con_master" title="Untitled Page" %>
4

```

La página maestra comienza con el carácter "~/ que indica que está en la raíz del sitio web. Esta sintaxis es muy útil porque en los controles cuando indiquemos una ruta que empiece por este carácter le indica que está en la raíz del web.

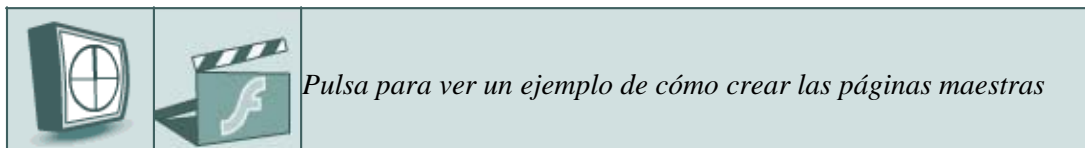
Estas páginas no tienen secciones "head" ni "body" ya que solo podremos poner contenido en su parte de "content". Esta parte en nuestro ejemplo quedaría:

```

<%@ Page Language="VB" MasterPageFile="~/Tema_10/plantilla_maestra.master"
AutoEventWireup="false" CodeFile="ejemplo_con_master.aspx.vb"
Inherits="Tema_10_ejemplo_con_master" title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
  <p style="text-align: center;">
    Este es el contenido real de la página que acabo de crear, lo demás me lo ha
    puesto la página maestra...</p>
</asp:Content>

```



6.2 Página maestra con varias regiones

Las páginas maestras pueden contener varias zonas simplemente añadiendo los "ContentPlaceHolders" que necesitemos y colocarlos en el orden adecuado. En este ejemplo incluye un "ContentPlaceHolders" donde el

El IDE, estilos, temas y páginas maestras

usuario puede incluir contenido y un recuadro sombreado (creado por una etiqueta <div>) que contiene un título de "Otros enlaces" y un segundo "ContentPlaceHolders". La idea es dividir la página en dos secciones lógicas. En la sección del contenido no nos tendemos que preocupar por dar formato a cada sección o de la posición de la sección de otros enlaces. En su lugar simplemente proporcionaremos contenido para cada sección y así ASP.NET lo insertará en la parte correcta de la página maestra.

```
<body>
  <form id="form1" runat="server">
    <div>
      <table class="style1">
        <tr>
          <td style="font-family: Verdana; font-size: 48px; font-weight: bold; background-color: #FFFFCC">
            Mi Empresa.com</td>
        </tr>
      </table>
      <asp:ContentPlaceHolder id="Seccion_principal" runat="server">
        </asp:ContentPlaceHolder>
        <div style="border-right: 1px solid; padding-right: 15px; border-top: 1px solid;
padding-left: 15px; font-size: small; padding-bottom: 15px; margin: 15px; border-left: 1px solid;
width: 417px; padding-top: 15px; border-bottom: 1px solid; font-family: Verdana;
background-color: #ffffcc">
          <strong >
            Otros enlaces<br />
          </strong>
          <asp:ContentPlaceHolder id="OtherLinksContent" runat="server">
            </asp:ContentPlaceHolder>
        </div>
        <span class="style2">Curso de Teleformación </span>
      </div>
    </form>
  </body>
```

Que nos va a producir:



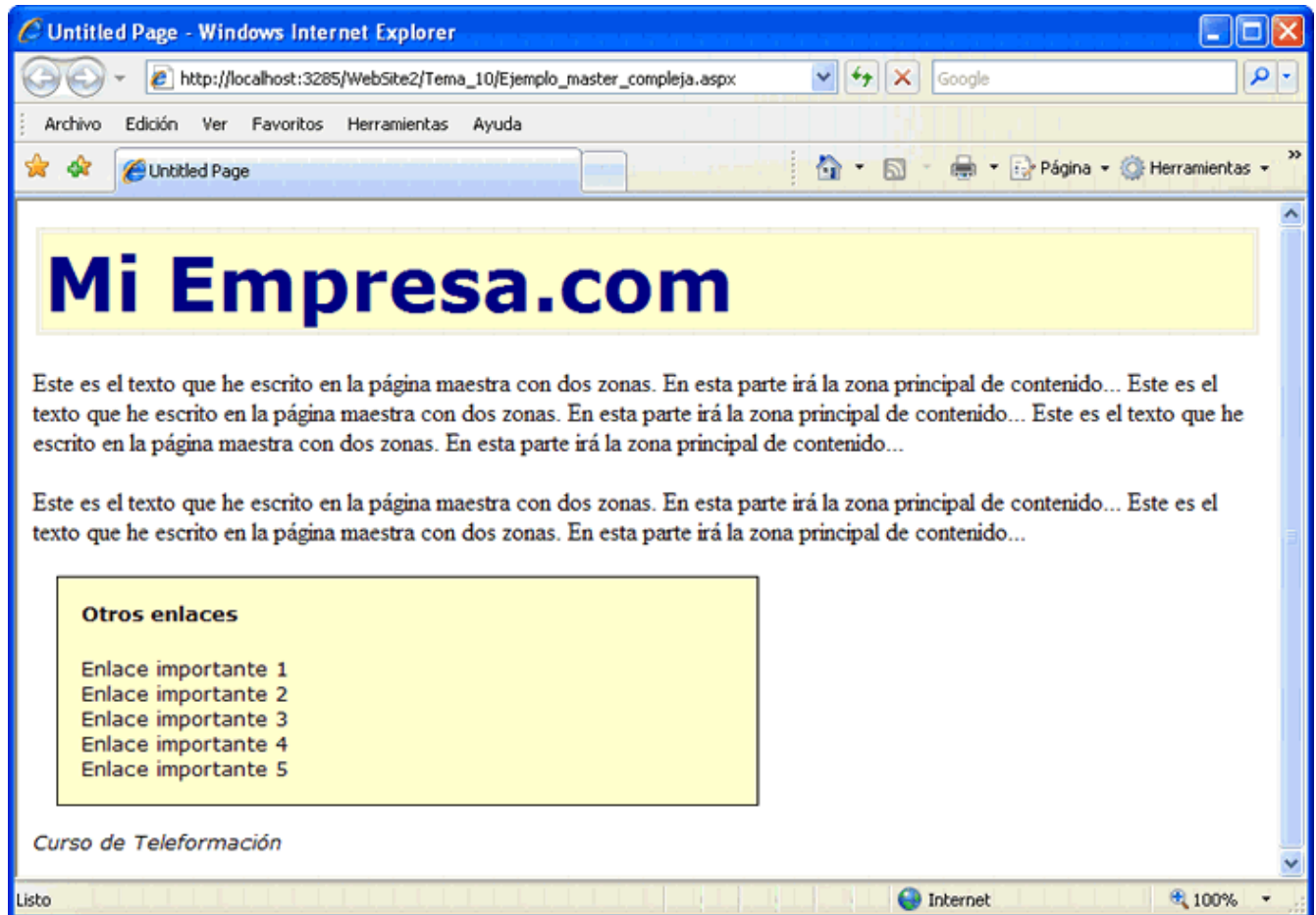
Como ves con dos zonas activas. Ahora creemos la páginas para el contenido, hacemos como antes indicando que va a tener una página maestra, que será la que acabamos de hacer. Veremos un código como este:

El IDE, estilos, temas y páginas maestras

```
<%@ Master Language="VB" MasterPageFile="~/Tema_10/plantilla_maestra_compleja.master"

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="Seccion_principal" Runat="Server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="OtherLinksContent" Runat="Server">
</asp:Content>
```

Si la editamos tendremos dos zonas activas donde pondremos nuestro contenido para obtener algo como esto:



6.3 Páginas maestras y rutas relativas

Tenemos que comentar cómo manejan las páginas maestras las rutas relativas. Si añadimos texto fijo no pasará nada pero si añadimos imágenes debemos tener cuidado. EL problema se da si ponemos la página maestra en otra carpeta distinta a la de la imagen. Esto se nos puede dar porque Microsoft recomienda almacenar las páginas maestras en una carpeta, pero que si no tenemos cuidado con las rutas podríamos tener problemas al utilizar rutas relativas. Por ejemplo, imagina que tenemos las páginas maestras en una carpeta llamada "paginas_maestras" y añadimos esto a una página de estas:

```

```

El IDE, estilos, temas y páginas maestras

Suponiendo que "\\paginas_maestras\miempresa.gif" existe todo parecería normal. Sin embargo, si creamos una página de contenido en otra carpeta, la ruta de esta imagen se interpreta como relativa en esa carpeta. Si el gráfico no existe ahí entonces nos mostrará una página con la marca de la imagen no cargada "X", o nos podría mostrar otra imagen si existiera una con el mismo nombre en esa carpeta.

Este problema se produce porque la etiqueta es una etiqueta ordinaria de HTML y por esto, ASP.NET no la modifica. Lo mismo puede suceder con enlaces, o con etiquetas "<a>" que proporcionan enlaces relativos a otras páginas con el elemento "link". Para resolver el problema podemos hacer dos cosas, decirle a las imágenes que se "ejecuten" en el servidor:

```

```

Y así ASP.NET sí que lo procesará de la forma correcta. Además, estaría perfectamente escrita si utilizamos el carácter que vimos antes y que indica la raíz de nuestra aplicación web: ~

```

```

Recuerda que este símbolo solo lo reconoce ASP.NET así que no lo utilices en otros controles que estén procesados por ASP.NET.

7. Páginas maestras avanzadas

Las páginas maestras que hemos visto serán mas que suficientes para la gran mayoría de nuestras situaciones. Sin embargo se nos pueden quedar cortas si queremos estructuras complejas. Vamos a aprender mas trucos para construir estas páginas que nos permitirán mas versatilidad, aunque ya no serán tan sencillas.

7.1 Esquemas basados en tablas

La idea es basarnos en tablas para crear las páginas maestras y las distintas zonas para los contenidos. Como las tablas son filas y columnas, la idea es poner contenido en una celda, así podremos limitar y dar formato al contenido, sabiendo que ocupará esa zona en la pantalla. Desgraciadamente no tenemos forma de hacerlo visualmente, sino que tendremos que poner etiquetas HTML en el fichero .aspx.

Veamos un ejemplo, imaginemos una zona con un aspecto como este:



Que tiene una tabla con una primera fila como encabezado, una segunda fila con dos columnas: una para los enlaces de navegación y otra para el contenido. Finalmente otra fila para el pie de página. Está claro que debemos crear una tabla con las etiquetas de fila y columna adecuadas <td> y <tr> y dentro de ellas pondremos nuestros "ContentPlaceHolder". Para empezar una tabla con anchura de 100%:

El IDE, estilos, temas y páginas maestras

```
<table style="width: 100%">

    ...

</table>
```

El siguiente caso poner una fila con la cabecera:

```
<table style="width: 100%">

    <tr>

        <td colspan="2">Cabecera de mi página</td>

    </tr>

    ...

</table>
```

Y así hasta terminar la página de esta forma:

```
<table style="width: 100%">
    <tr>
        <td colspan="2"><h1>Cabecera de mi página</h1></td>
    </tr>
    <tr>
        <td style="width: 100px">
            <asp:TreeView ID="TreeView1" runat="server">
                <Nodes>
                    <asp:TreeNode Text="Página 1" Value="Pagina 1"></asp:TreeNode>
                    <asp:TreeNode Text="Página 2" Value="Pagina 2"></asp:TreeNode>
                </nodes>
            </asp:TreeView>
            <br />
        </td>
        <td>
            <asp:contentplaceholder ID="conteniudo" runat="server">
            </asp:contentplaceholder>
        </td>
    </tr>
    <tr>
        <td colspan="2"><i>Pié de página</i></td>
    </tr>
</table>
```

Que nos dará una tabla formateada con el "contentplaceholder" en su sitio:



Hemos utilizado un control que veremos mas adelante para navegación que corresponde a una lista jerárquica de enlaces, como las que aparecen en el explorador de ficheros.

7.2 Código en la página maestra

En estos ejemplos hemos puesto código estático. Pero como estamos programando, sería interesante poner código para que estas páginas maestras nos respondan a ciertos eventos. Por ejemplo, que en el "Page_Load" nos inicie la página maestra. O detectar los clics de los controles para actualizar el contenido de las zonas que queramos.

Por tanto las páginas maestras pueden respondernos a eventos o proporcionarnos propiedades a nuestras páginas para cambiar su aspecto o comportamiento. Esto nos va a permitir interactuar el contenido de la página con la página maestra.

Por ejemplo, imagina que queremos que el usuario quiera ocultar la zona de la izquierda de los enlaces para que tenga mas área de trabajo. No queremos implementar esto en la página maestra porque no queremos que se ejecute en todas y no podemos ponerla en la del contenido porque éstas no pueden modificar las páginas maestras. La solución es crear una forma para que la página contenido interactué con la página maestra, así podremos preguntar a la página maestra si quiere ocultar los controles o enlaces de navegación.

Pondremos en nuestro contenido dos botones para mostrar u ocultar el control:



Una buena forma es añadiendo una nueva propiedad llamada "control_navegacion" a la clase de la página maestra:

```
Public Property control_navegacion () as Boolean  
  
Get
```

El IDE, estilos, temas y páginas maestras

```
Return treeview1.Visible

End Get

Set (ByVal Value as Boolean)

    treeview1.Visible=Value

End Set

End Property
```

Fíjate la versatilidad que tenemos con la programación orientada a objetos que en cualquier momento podemos exponer una propiedad de la gran clase que es nuestra aplicación web. Por un lado es pública, para poderse acceder desde el exterior, es decir, por otras clases o páginas. Por otro lado establece la conocida propiedad de Visible (disponible en casi todos los controles de ASP.NET) en el control jerárquico de la página maestra. Cualquiera que sea el valor que le pasemos a "control_navegacion" se lo aplicará a la propiedad "Visible" del control "treeview1". Esto debe ser así porque normalmente la propiedad visible de este control no es accesible directamente desde la página contenido.

Para acceder a esta página la página de contenido utiliza una propiedad suya llamada "Page.Master" que siempre devuelve el objeto enlazado en la página maestra. Sin embargo podremos acceder a la propiedad "Page.Master".control_navegacion" directamente puesto que la propiedad Page.Master utiliza la clase "MasterPage". Para acceder a estas propiedades personalizadas, como la que hemos creado, debemos hacer un casting (indicarle el tipo de datos) del objeto "Page.Master" con el tipo apropiado.

El controlador del evento clic de los botones de ocultar o mostrar que pondremos en la página serán ahora muy sencillos:

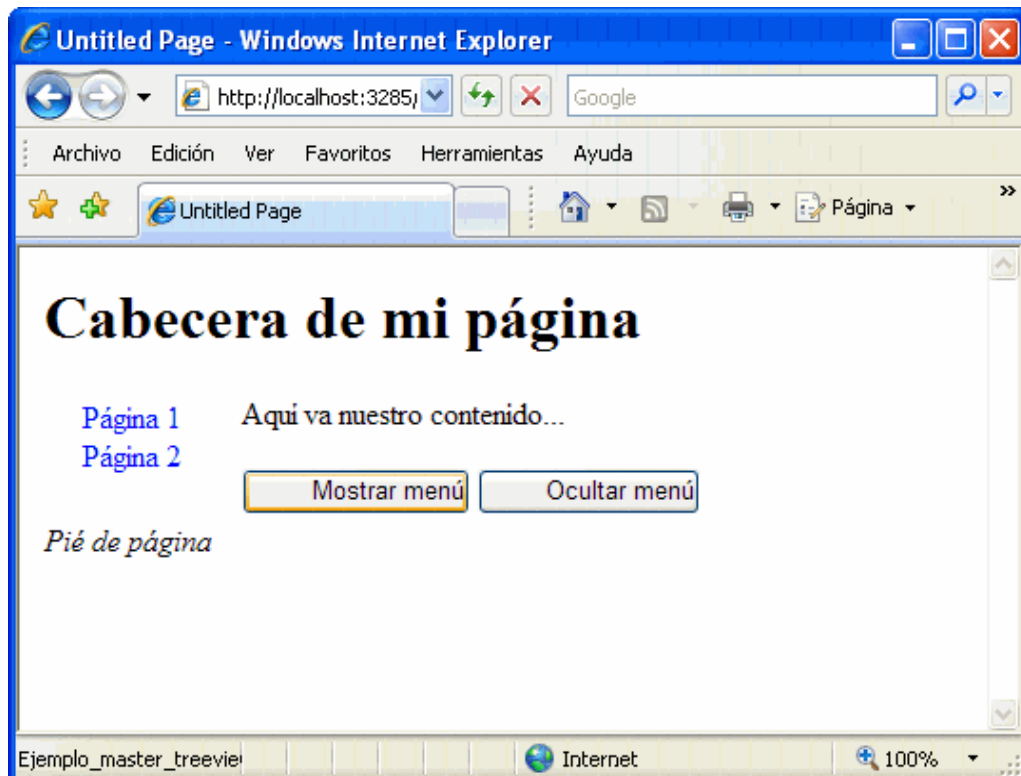
```
Protected Sub btn_ocultar_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim maestra As Tema_10_Ejemplo_master_con_tabla = _
        CType(Me.Master, Tema_10_Ejemplo_master_con_tabla)

    maestra.control_navegacion = False
End Sub
```

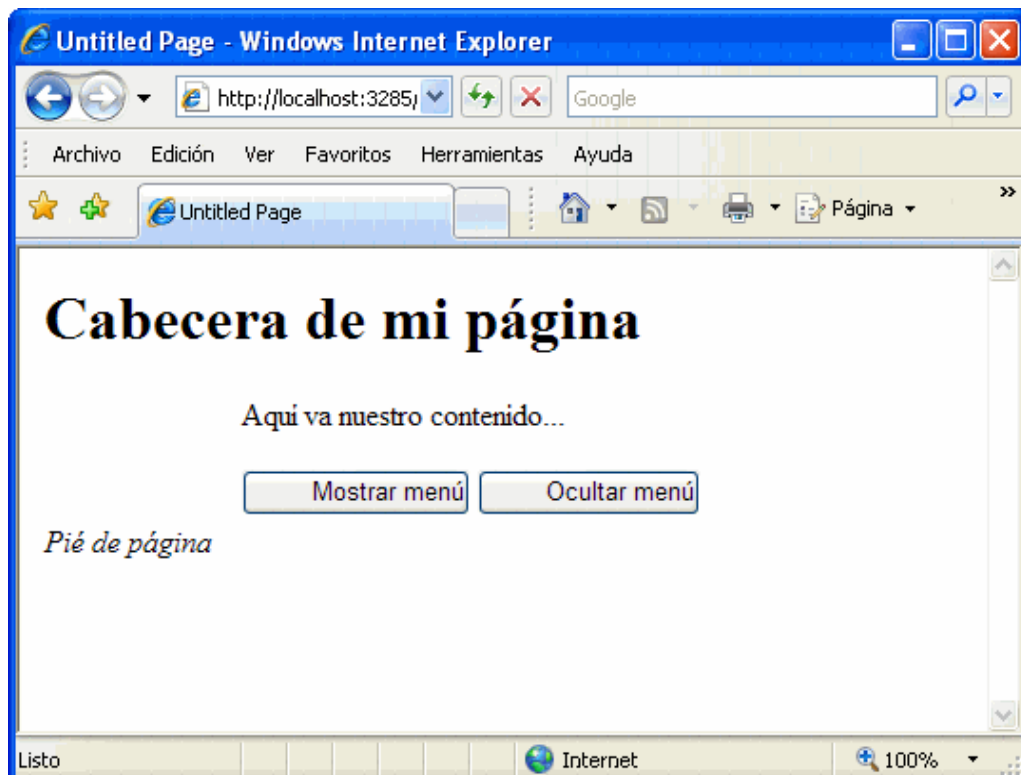
```
Protected Sub btn_mostrar_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim maestra As Tema_10_Ejemplo_master_con_tabla = _
        CType(Me.Master, Tema_10_Ejemplo_master_con_tabla)

    maestra.control_navegacion = True
End Sub
```

Y el resultado el siguiente:



Y pulsamos clic en el botón:



Fabuloso. No te imaginas lo que hemos conseguido con esto: desde otra página distinta (la del contenido) hemos conseguido mostrar y ocultar un control. Ahora te das cuenta de porqué hemos tenido que publicar esa propiedad desde la página maestra. Pues sí, para poder manipular la propiedad de un control desde otra

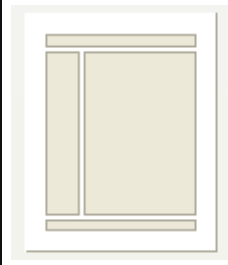
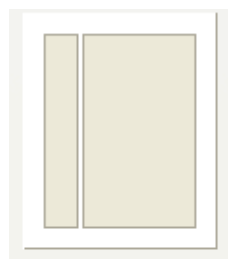
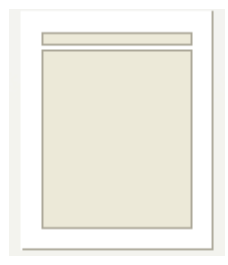
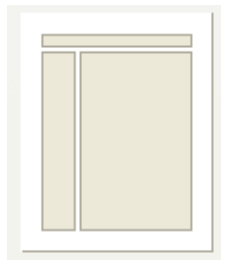
El IDE, estilos, temas y páginas maestras

página. Y esa otra página en el contenido, que al acceder a la propiedad del control contenido en la otra página hemos conseguido manipular su estado.

Esto tiene mas importancia de la que parece porque te hace pensar ya en la aplicación web como en un "todo" en el que los objetos (páginas) pueden acceder a los control ASP.NET de otras y esto es mas que potente.

Después de esta grata sorpresa que nos abre multitud de posibilidades vamos a ver los controles que tenemos para la navegación o exploración de las páginas de nuestra aplicación web.

Y para terminar mostrarte simplemente otros estilos de webs muy utilizados que puedes utilizar en las páginas maestras con tablas:

Estilos de marcos con encabezados, índices y pies de página			
			

Cualquier diseño de estos es bueno, unos con índices a la izquierda, otros con encabezados que serán fijos en todo el Web... esto lo dejamos ya a tu elección.

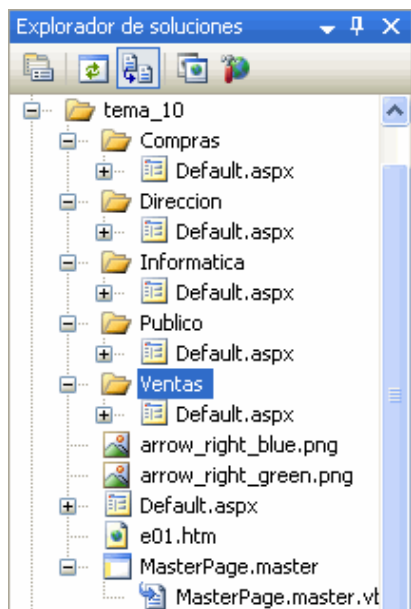
[Pulsa aquí para descargar los ejemplos de este tema](#)

Ejercicios

Ejercicio 1

Crea una estructura de carpetas para varias secciones en nuestra web:

- Compras
- Ventas
- Administración
- Informática
- Público (para acceso libre)



Crea una página maestra con un menú a la izquierda y el contenido en la parte de la derecha debajo de un título. Para el diseño inicial ayúdate si quieres de FrontPage:

Ojo, si lo haces inicialmente con FrontPage asegúrate al pegarlo en el IDE que las imágenes y los enlaces del menú sean controles ASP (image y hyperlink)



Dentro de cada carpeta pon una página por defecto y que utilicen todas la página maestra anterior. Para dar como resultado:



Ejercicio 2

Crea una hoja de estilos y la asocias con los archivos anteriores.