
Introducción a ASP.NET

ASP.NET es la herramienta de Microsoft que facilita la creación de sitios Web dinámicos con acceso a bases de datos.

En un principio, los sitios Web estaban basados en páginas estáticas (no permitían ningún tipo de comunicación con el usuario) escritas en lenguaje HTML. Posteriormente se introdujeron los formularios como modo sencillo de comunicación con los usuarios. Esta comunicación se basaba habitualmente en cuestionarios simples que eran procesados por programas escritos en algún lenguaje como PERL, C, etc. Mediante estos programas se generaba una respuesta (en HTML) que se retornaba al usuario.

Este modo de trabajar sigue el denominado modelo **cliente-servidor** en el cual los equipos intercambian información para realizar una tarea actuando uno de ellos como cliente (el que solicita la información) y el otro como servidor (el encargado de proporcionarlo).

En Internet, un cliente habitualmente es un dispositivo con un navegador de páginas Web, mientras que un servidor es un equipo (o un conjunto de equipos) que contiene información sobre un sitio Web (páginas HTML, imágenes, scripts, etc) y un programa encargado de procesar las peticiones de los clientes y enviar las respuestas a los mismos.

Las páginas devueltas por el servidor a las peticiones de los clientes pueden estar almacenadas en el servidor y reenviadas sin más (páginas estáticas) o pueden ser generadas por el propio servidor en función de las peticiones del cliente mediante la ejecución de programas CGI o páginas ASP clásicas (páginas dinámicas).

Además, actualmente ha surgido una tercera vía de comunicación con el cliente (representada por ASP.NET y otras tecnologías similares) en la que el servidor además puede responder a **sucesos que se producen en el cliente**. Por tanto, la diferencia fundamental entre CGI y ASP y ASP.NET es que con ASP.NET el servidor puede reaccionar ante lo que sucede en el cliente sin esperar a que éste envíe los datos. Por ejemplo, con ASP clásico si necesitamos validar un campo es necesario enviar todos los datos para ser procesados por el servidor (a no ser que se emplee código de parte del cliente como comentaremos más adelante). Sin embargo, con ASP.NET el servidor puede ejecutar el código de validación en el momento en que el cliente introduzca los datos en el campo a validar.

El código fuente que genera una página o aplicación ASP.NET es muy similar al que genera ASP. Lo único que cambia es el proceso intermedio y final de conversión de las llamadas ASP

correspondientes a la página Web y que será devuelta al cliente así como algunas estructuras y llamadas internas.

La arquitectura ASP.NET ha sido reprogramada por completo por lo que ASP.NET podría ser considerado un nuevo lenguaje y no una nueva versión de ASP.

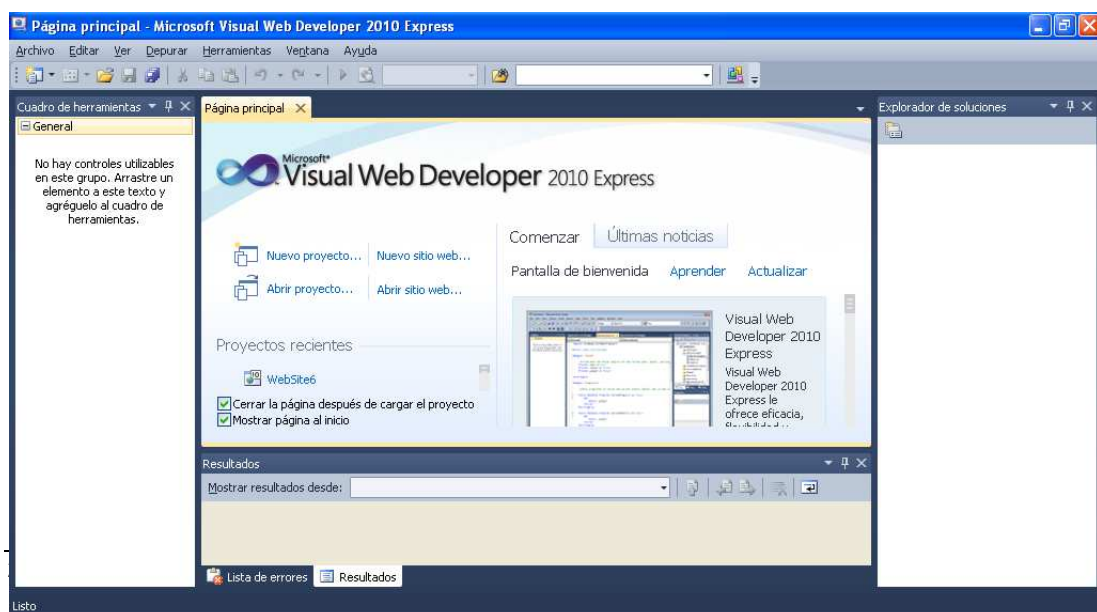
Como ya se ha comentado, la única posibilidad que había tradicionalmente de responder a eventos en el lado cliente era mediante la inclusión de scripts en el lado cliente (mediante JavaScript o VBScript). Por tanto podemos tener código en el cliente (embebido en las páginas HTML) o en el servidor. Las diferencias fundamentales entre ambas son:

En el cliente	En el servidor
Generalmente se usa para producir efectos dinámicos como variaciones de imagen ante una acción del usuario, uso de menús y de cuadros de mensajes...	Generalmente se usa para el procesamiento de datos, acceso a bases de datos y devolución de los resultados obtenidos.
El código es ejecutado por el navegador en el cliente	El código se ejecuta en el servidor
El código es interpretado por el navegador	El código es compilado por ASP.NET (en nuestro caso).
Un problema es la compatibilidad de los diferentes navegadores con el lenguaje de script utilizado.	Al devolver la páginas HTML estándar no existen incompatibilidades con ningún navegador
El código va incluido en la página HTML bajo unas marcas especiales que indican que debe ser procesado.	Los resultados deben ser convertidos a HTML, XML, antes de ser enviados al cliente

Crear mi primer sitio Web

Vamos a utilizar el programa Visual Web Developer Express (parte de Visual Studio) para poder construir nuestros sitios y aplicaciones Web. Nos los bajaremos e instalaremos.

Al arrancar el Visual Web Developer (utilizaré VWD a partir de este momento) nos encontramos con la siguiente pantalla:

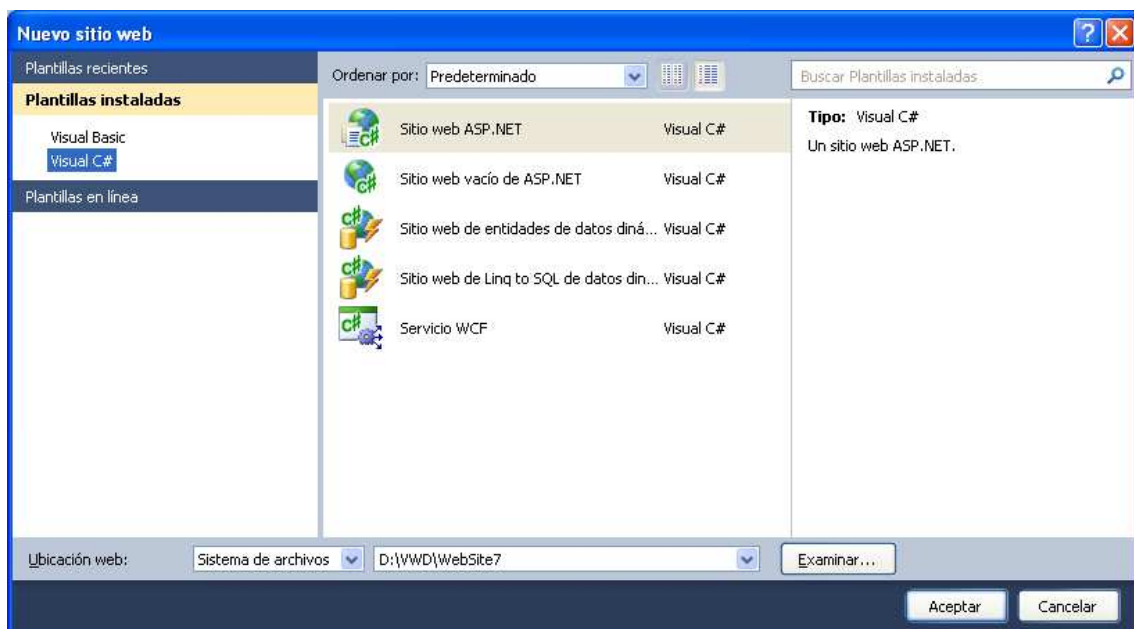


Podremos pulsar **Archivo→ Nuevo Proyecto** o **Archivo→ Nuevo Sitio Web**. Dos de las diferencias principales entre ambos tipos es que:

- En las "aplicaciones Web" todo el código se compila en una DLL y posteriormente se publica esa DLL.
Los "sitios Web" usan DLLs dinámicas, es decir, no se crea una sola DLL con todo el código, sino que puedes crear una DLL por cada carpeta (el propio VS se encarga de eso), además de que se crean las carpetas de código que puedes "publicar" y será el propio entorno de ejecución del servidor el que se encargue de compilarlo, etc.
- En el Sitio Web aparece ya el archivo **Global.asax** que no aparece en las aplicaciones Web para controlar lo que se debe hacer al arrancar una sesión de usuario y finalizarla y al arrancar la aplicación y finalizarla. En el sitio Web no aparece por defecto, pero siempre lo podemos incorporar si tenemos necesidad de introducir código en estos momentos.
- Además aparece alguna carpeta más en las aplicaciones Web, pero siempre vamos a poder incluirlas en el sitio Web.

Como crear un nuevo sitio Web:

Pulsaremos Nuevo Sitio Web:

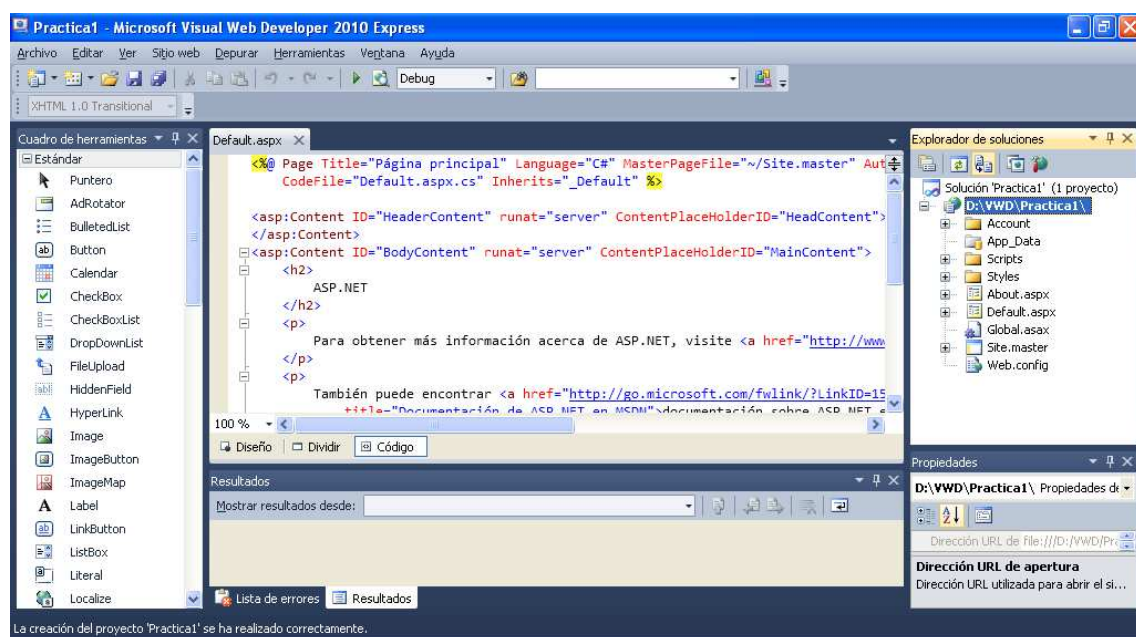


Podremos elegir entre un Sitio Web ASP.NET y un sitio Web vacío de ASP.NET. Además en la barra izquierda tenemos el lenguaje con el que programar nuestros eventos de servidor: Visual Basic o Visual C#. En nuestro caso elegiremos Visual c#.

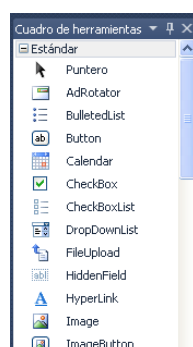
Vamos a poder con el botón Examinar elegir la ruta en la que queremos guardar nuestros sitios web y la ubicación que vamos a utilizar en clase va a ser nuestro Sistema de archivos. Tenemos también los valores **FTP** y **http**, si utilizamos estos protocolos a la hora de guardar nuestros sitios. Además si tenemos instalado el Servidor Web IIS aparecerá una opción más, que es **IIS** para que vaya creando nuestros sitios en directorios virtuales dentro de la carpeta c:\inetpub\wwwroot.

Pondremos el nombre del proyecto o sitio y pulsaremos Aceptar.

Si hemos elegido en la pantalla anterior la primera opción **Sitio Web ASP.NET** nos aparecerá una ventana como la siguiente:

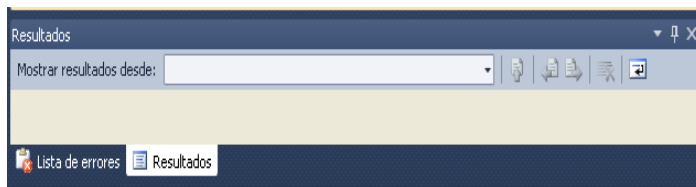
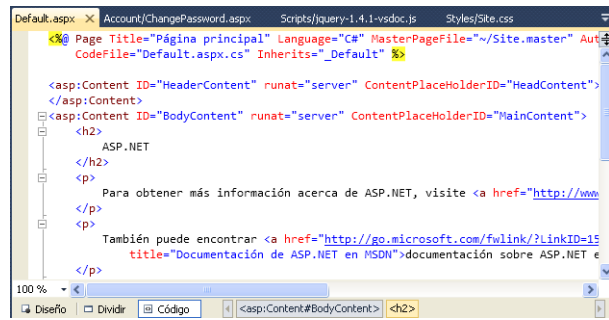


Vamos a encontrar la pantalla dividida en una gran cantidad de ventanas: Empezando de izquierda a derecha:



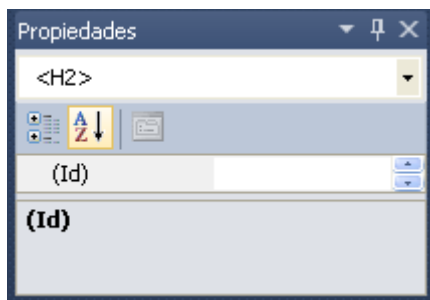
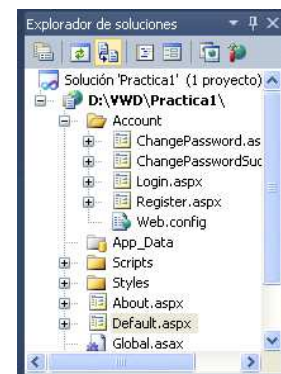
Cuadro de Herramientas: donde encontraremos todos los controles disponibles para utilizar en nuestras páginas Web. Están agrupados en diferentes categorías.

Ventana Principal: en donde podremos ver el código o diseño de nuestras páginas. En la parte de debajo de esta ventana están las pestañas Diseño, Dividir y código desde donde podemos cambiar de una a otra visión. La pestaña dividir mostrará la ventana dividida mostrando el diseño y el código a la vez.



Ventana de Resultados: en donde se mostrará como ha ido la compilación de nuestro sitio web y posibles errores.

Explorador de Soluciones: donde encontraremos el proyecto o sitio web escrito en negrita y colgando de él todos los ficheros que forman parte del sitio bien organizados en carpetas. Carpetas para Hojas de Estilo en cascada, carpeta para scripts que se ejecutarán en el cliente, carpetas para imágenes,....

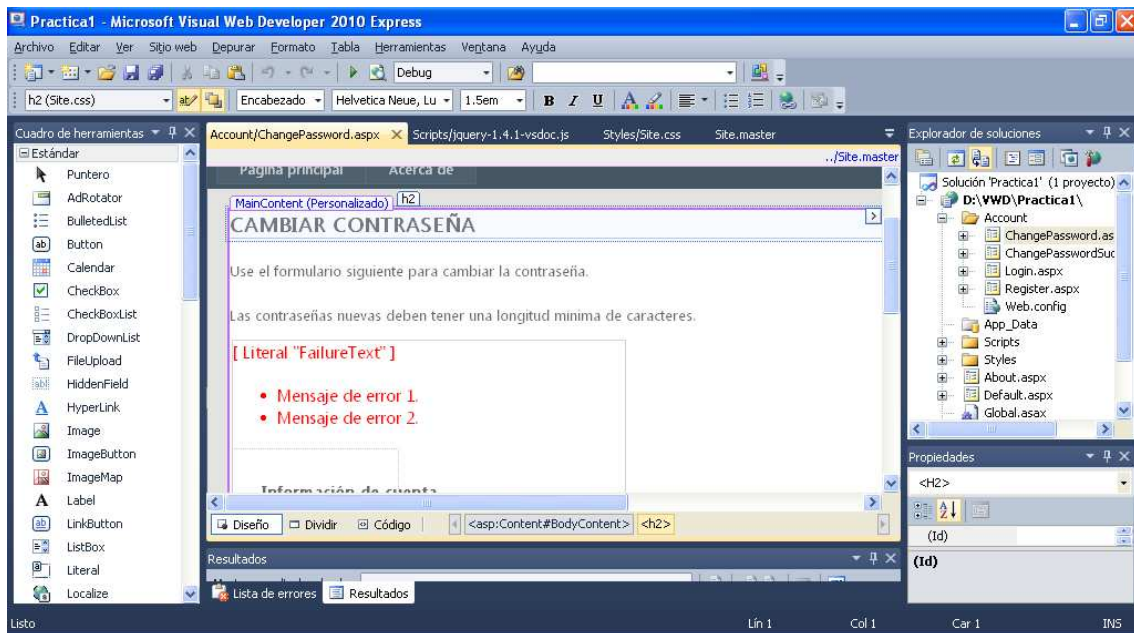


Ventana de Propiedades: donde en diseño podremos ver y modificar cada una de las propiedades que tienen nuestros controles.

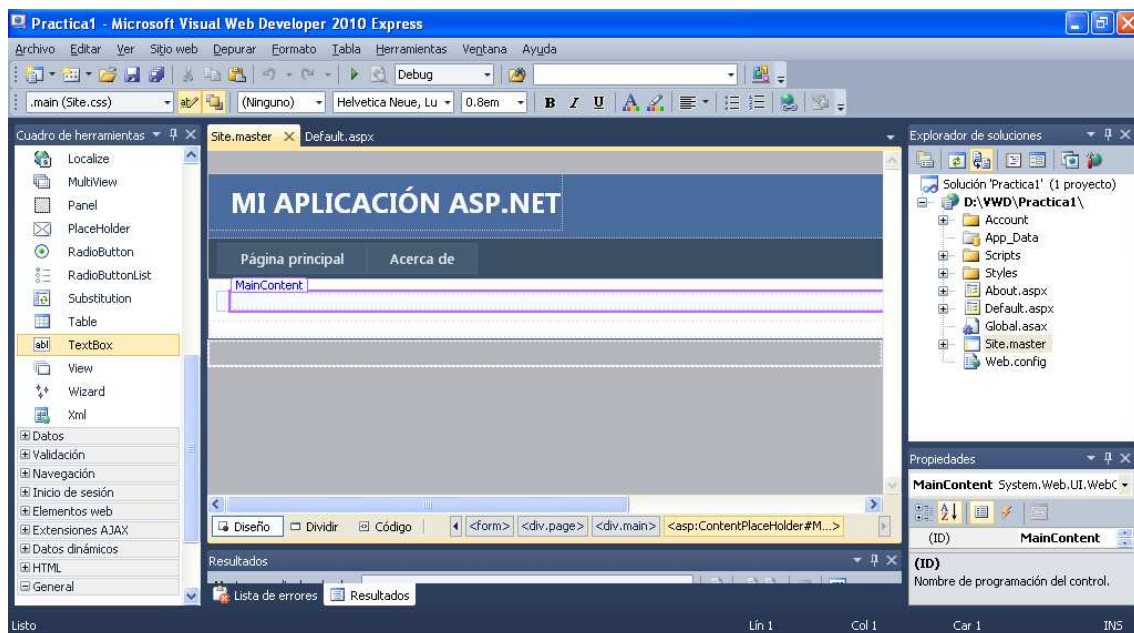
Cada una de estas ventanas podemos ocultarla, cerrarla o volver a abrirla desde el menú Ver → Otras ventanas.

Entendiendo el Sitio Web

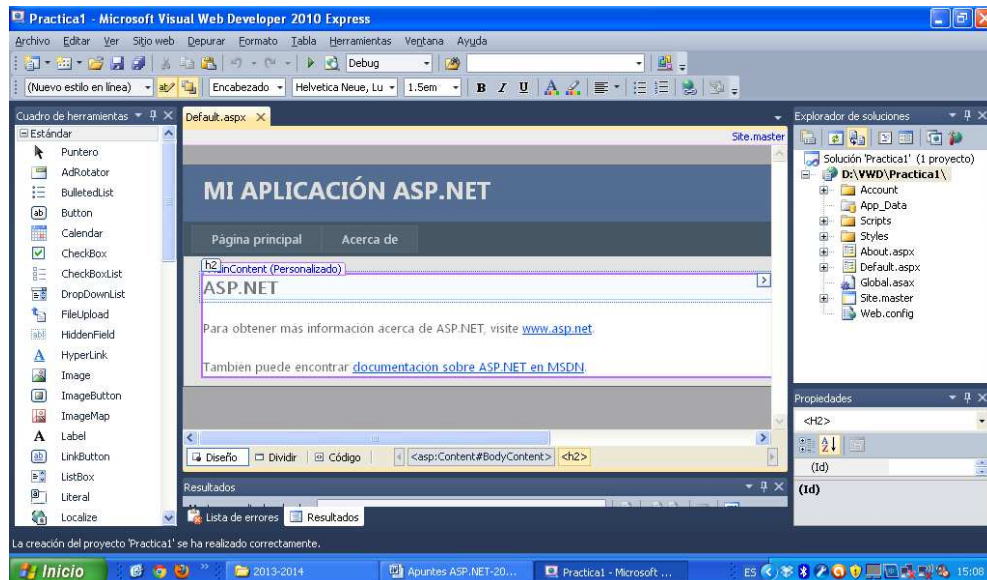
Si comenzamos mirando el **Explorador de Soluciones** nos encontraremos con una gran cantidad de ficheros. Generalmente las páginas **aspx** están en el directorio principal del proyecto, aunque en este ejemplo también hay páginas dentro de una carpeta denominada **Account** que contiene todas aquellas páginas dedicadas a controlar la sesión de un usuario (identificación o login, cambio de contraseña,...).



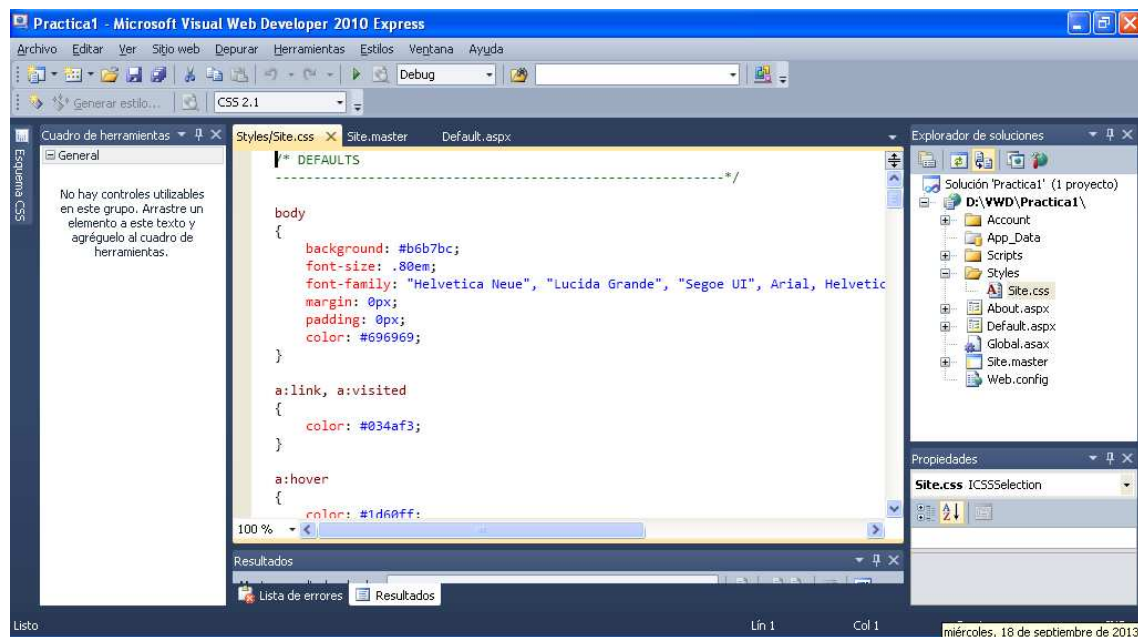
Además nos encontraremos con un archivo denominado **Site.master**.



Este tipo de páginas que tienen como extensión **.master** son las denominadas **Master Pages** (Páginas Principales). Estas páginas van a contener el diseño y la programación que queremos que se encuentre en todas (o la mayoría de) las páginas de nuestro sitio web. Las **Master Pages** tienen uno o varios controles denominados **MainContent** que son espacios en los que las **páginas aspx** pueden incluir los controles propios de la página.



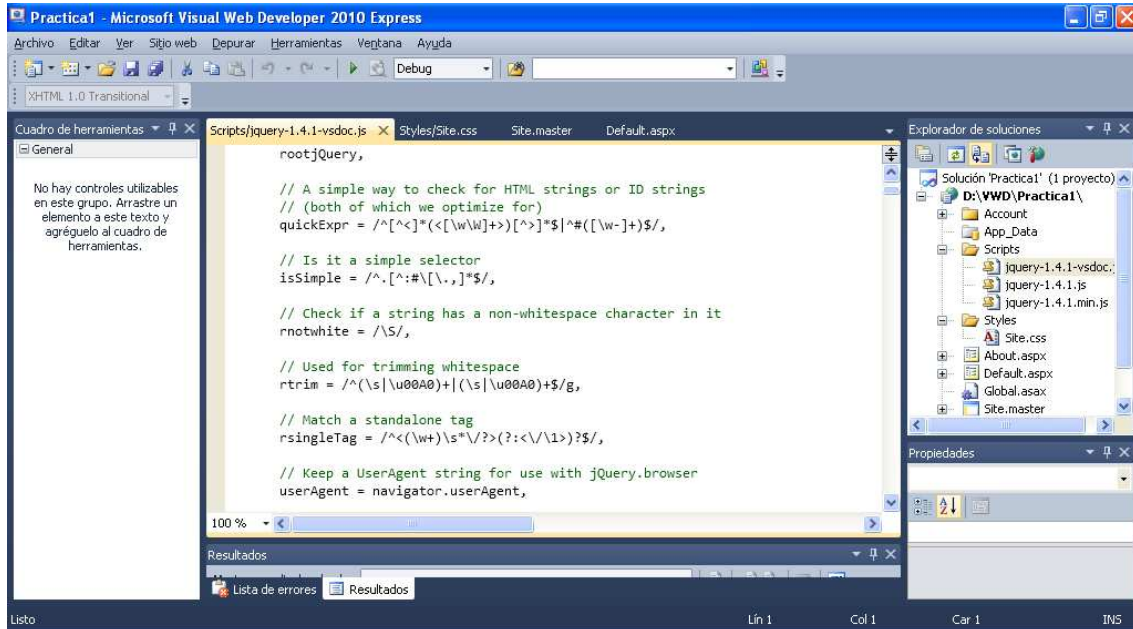
Como vemos la **página Default.aspx** tiene el mismo diseño que la **Page Master** de nuestro sitio web (**Site.master**), pero incorpora dentro del control **MainContent** sus propios controles, en este caso muestra unas etiquetas con texto e hipervínculos.



Además, como muestra la imagen anterior, podemos encontrar una carpeta **Styles** en donde encontraremos una hoja de estilos **Site.css**, que contendrá las reglas que seguirán los

controles y etiquetas de mis páginas. En la página maestra o en las páginas **aspx** que utilicen estas reglas, debe haber una referencia a esta hoja de estilos mediante la etiqueta:

<link rel="stylesheet" type="text/css" href="Styles/Site.css" />



Encontraremos también la carpeta **Scripts**, en donde meteremos nuestro código en JavaScript o cualquier otro lenguaje de Script para ejecutar código en el cliente.

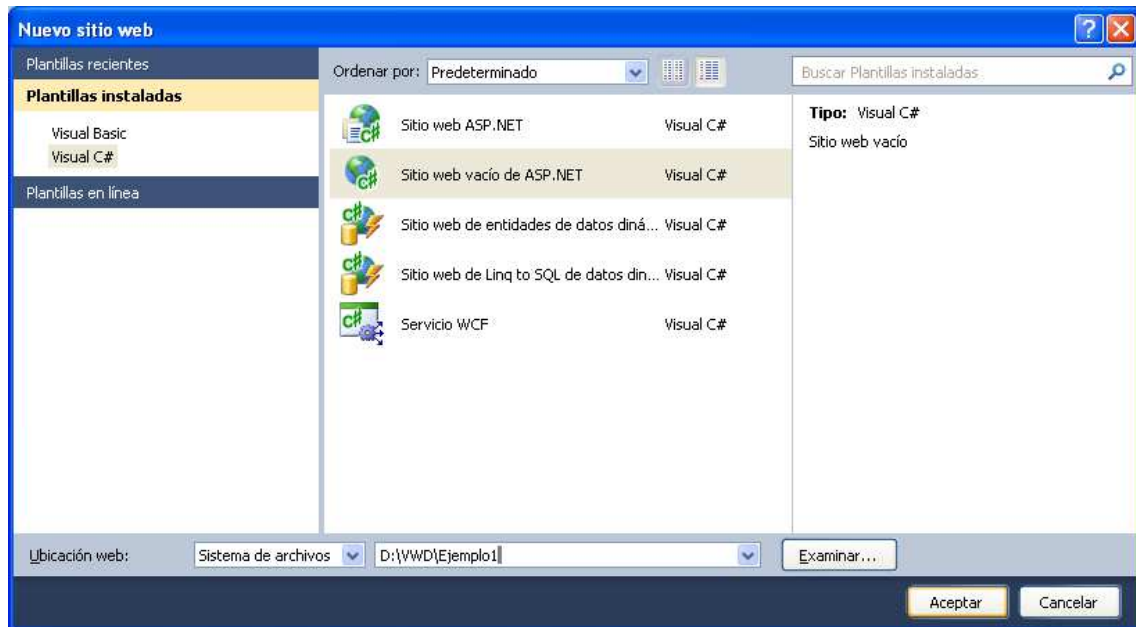
Debemos incluir una referencia a estas páginas de código mediante la etiqueta:

<script language="javascript" type="text/javascript" src="Scripts/archivo.js"></script>

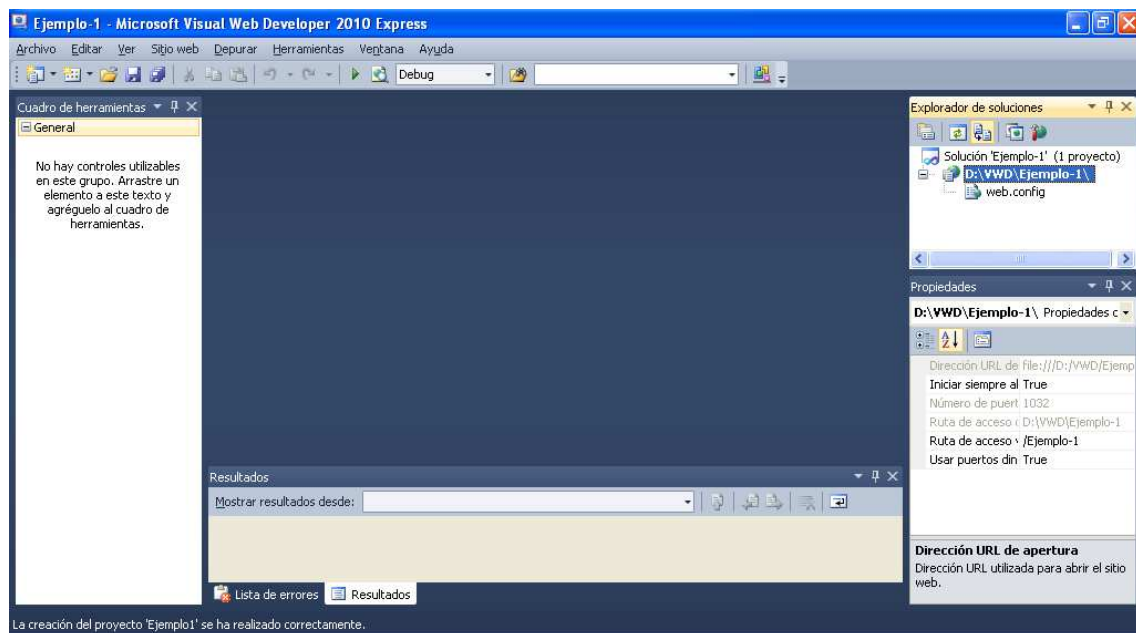
Por ultimo nos encontraremos con el archivo **web.config**. Este es el archivo principal de opciones de configuración para una aplicación web en **ASP.NET**. El archivo es un documento **XML** que define información de configuración concerniente a la aplicación web. El archivo **web.config** contiene información que controla la carga de módulos, configuraciones de seguridad, configuraciones del estado de la sesión, opciones de compilación y el lenguaje de la aplicación. Los archivos **web.config** pueden contener también objetos específicos tales como **cadenas de conexión** a la **base de datos**.

Creación de un nuevo sitio Web vacío:

Al crear un nuevo sitio web, también tenemos la opción de crear el sitio web vacío:



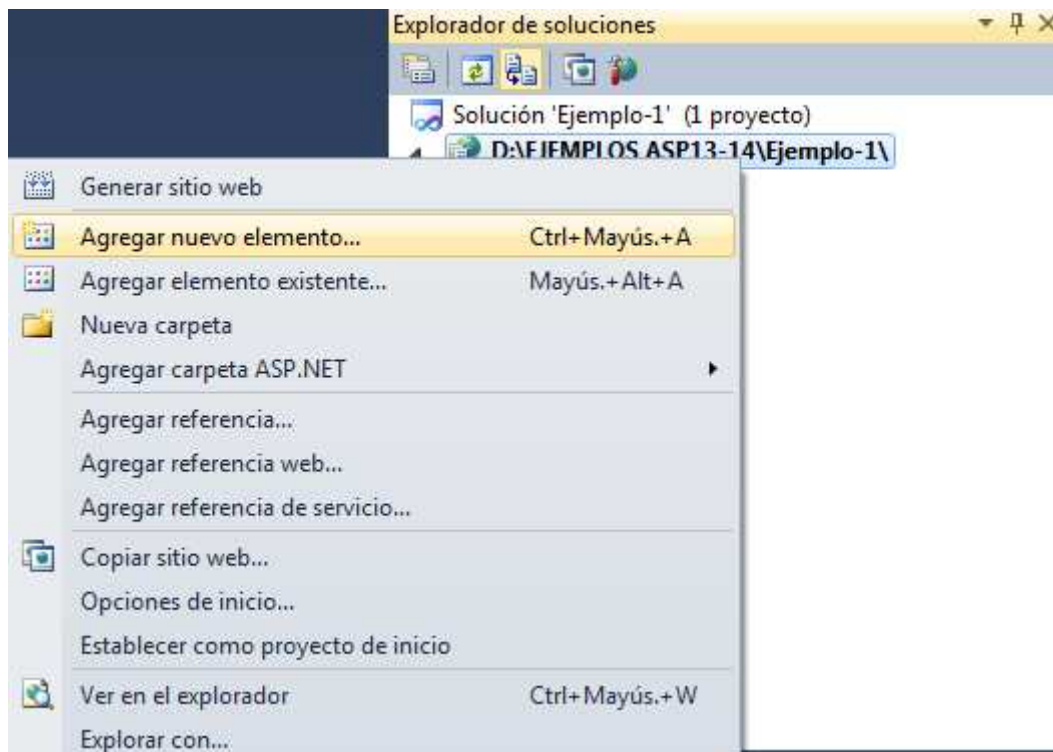
Una vez puesto el nombre del sitio web y pulsar aceptar se nos mostrará la siguiente pantalla:



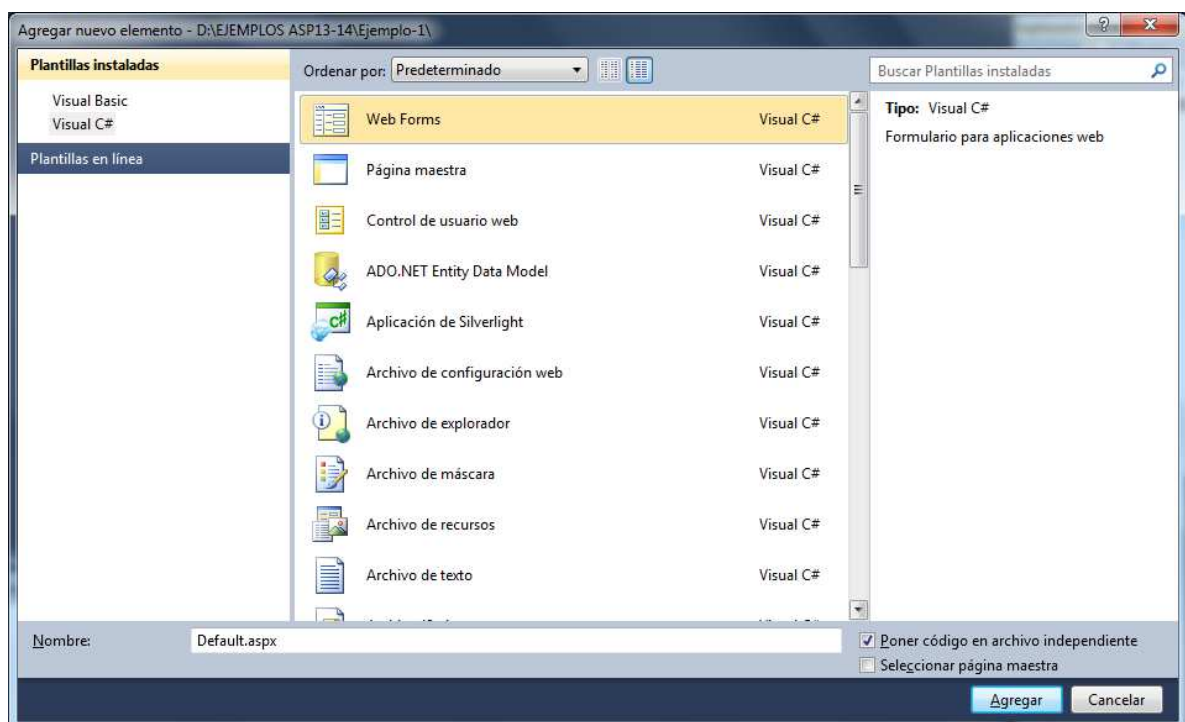
Como podemos observar en el Explorador de Soluciones, solo se crea bajo el sitio web el archivo Web.config. Es cosa nuestra empezar a crear nuestro sitio completo.

1º Añadir un Web Form (Página .aspx)

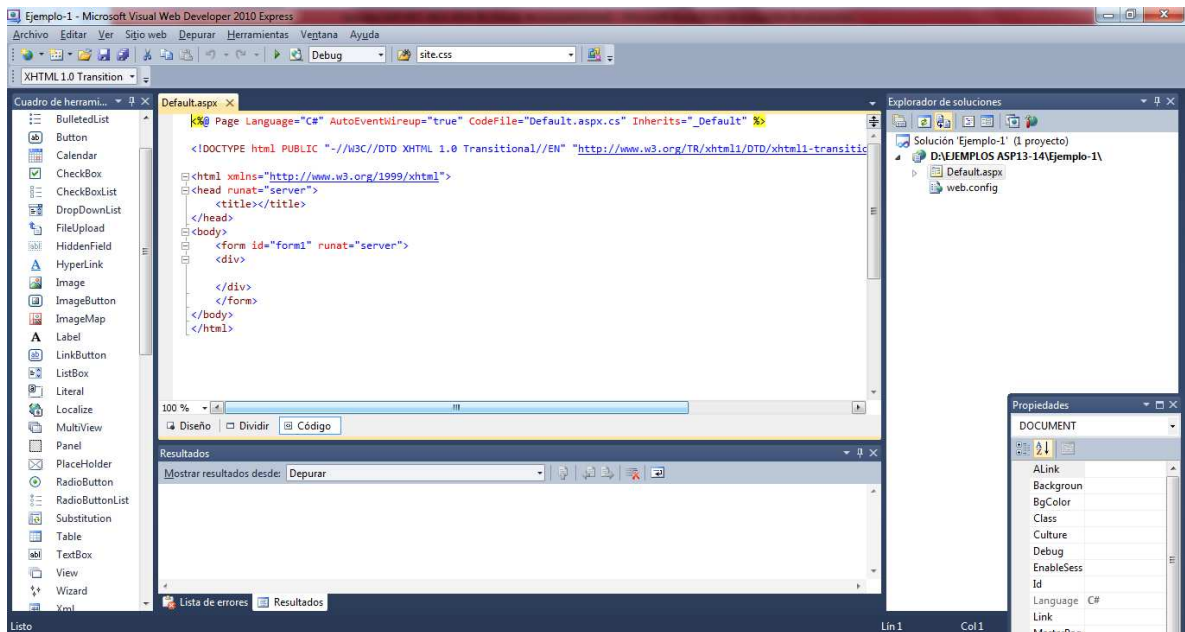
Pulsaremos botón derecho sobre el sitio Web en el explorador de soluciones y pulsaremos la opción de Añadir un nuevo elemento:



Se mostrará el siguiente cuadro de diálogo

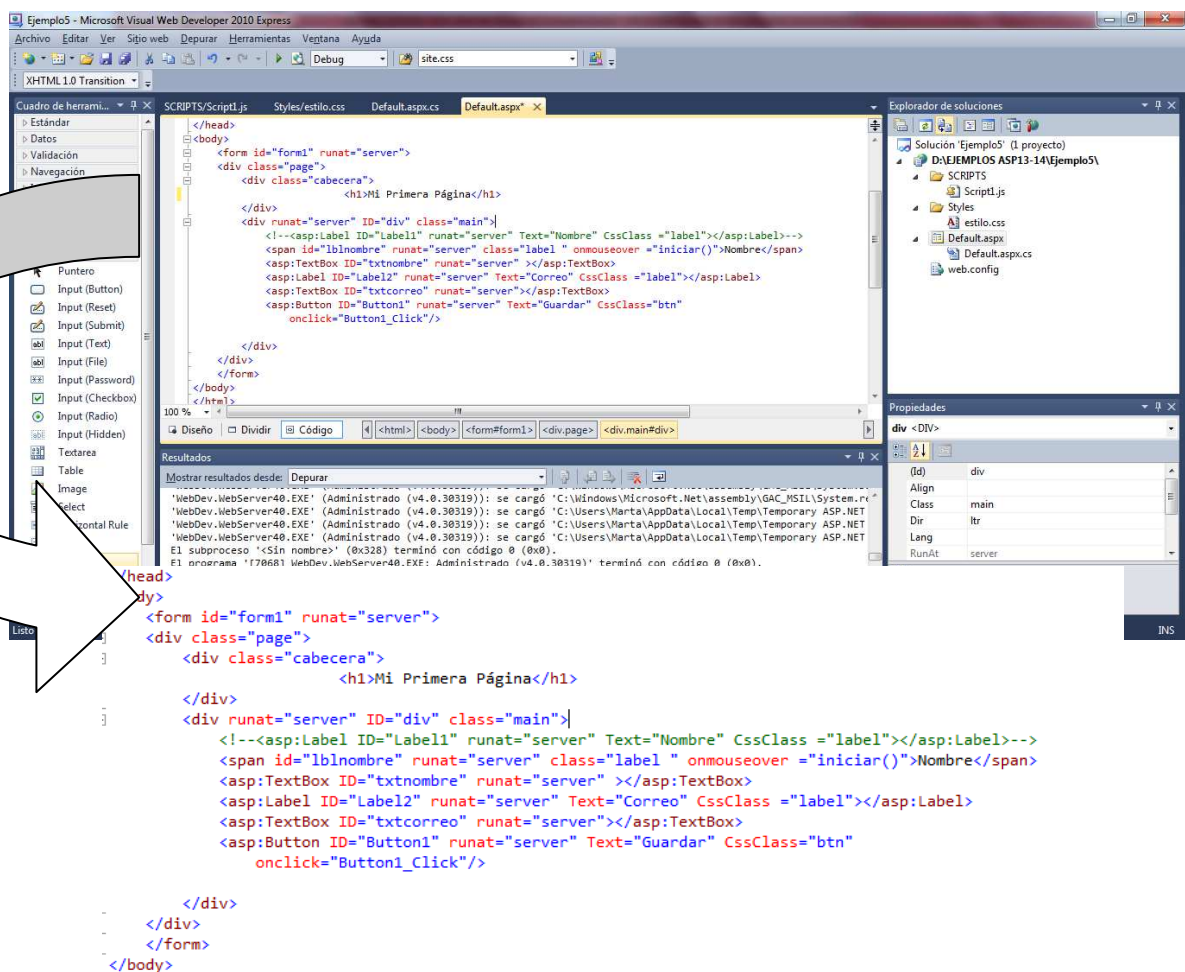


En el que pulsaremos la opción Web Form y si queremos le cambiaremos el nombre. Por



defecto se denominará a esta página como Default.aspx.

Ahora se pasará a introducir los controles (bien HTML o ASP) que deseemos. Por ejemplo:



2º Crear la Hoja de Estilos.

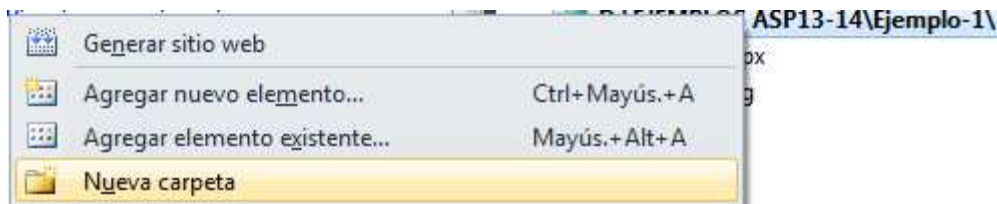
A continuación se creará **la hoja de estilos** correspondientes a nuestros controles. Debemos recordar que si estamos tratando con **etiquetas de HTML** podemos asignarles una clase en CSS con el atributo **class**.

```
<div class="cabecera">
    <h1>Mi Primera Página</h1>
</div>
<div runat="server" ID="div" class="main">|
```

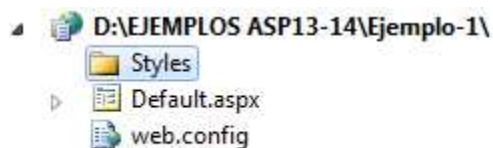
Sin embargo si el control es **ASP** las clases de CSS se asignan mediante el atributo **CssClass**.

```
<!--<asp:Label ID="Label1" runat="server" Text="Nombre" CssClass ="label"></asp:Label-->
<span id="lblnombre" runat="server" class="label " onmouseover ="iniciar()">Nombre</span>
<asp:TextBox ID="txtnombre" runat="server" ></asp:TextBox>
<asp:Label ID="Label2" runat="server" Text="Correo" CssClass ="label"></asp:Label>
<asp:TextBox ID="txtcorreo" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Guardar" CssClass="btn"
    onclick="Button1_Click"/>
```

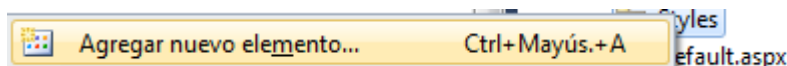
Pulsando con el botón derecho sobre **el Sitio Web** pulsaremos **Nueva Carpeta** y le pondremos el nombre de **Styles**.



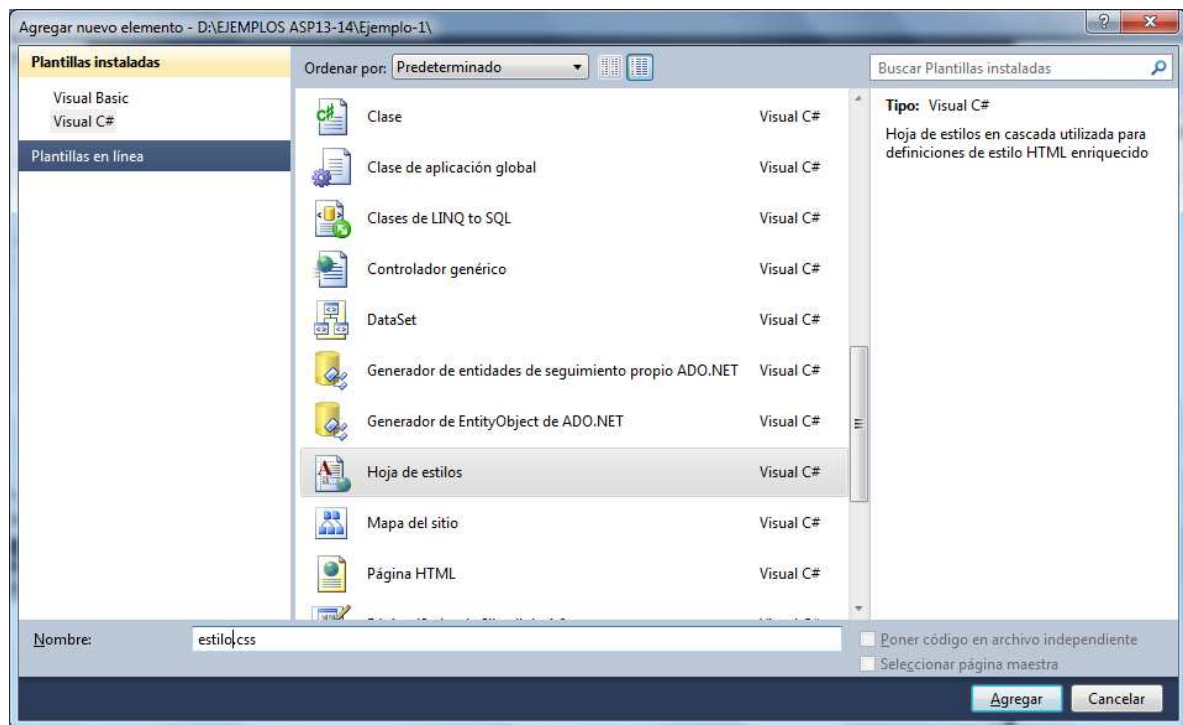
Quedará así:



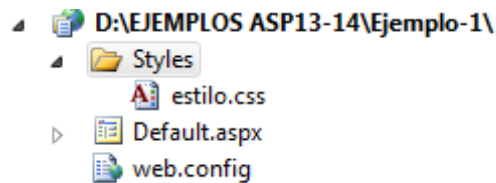
Pulsaremos Agregar nuevo elemento:



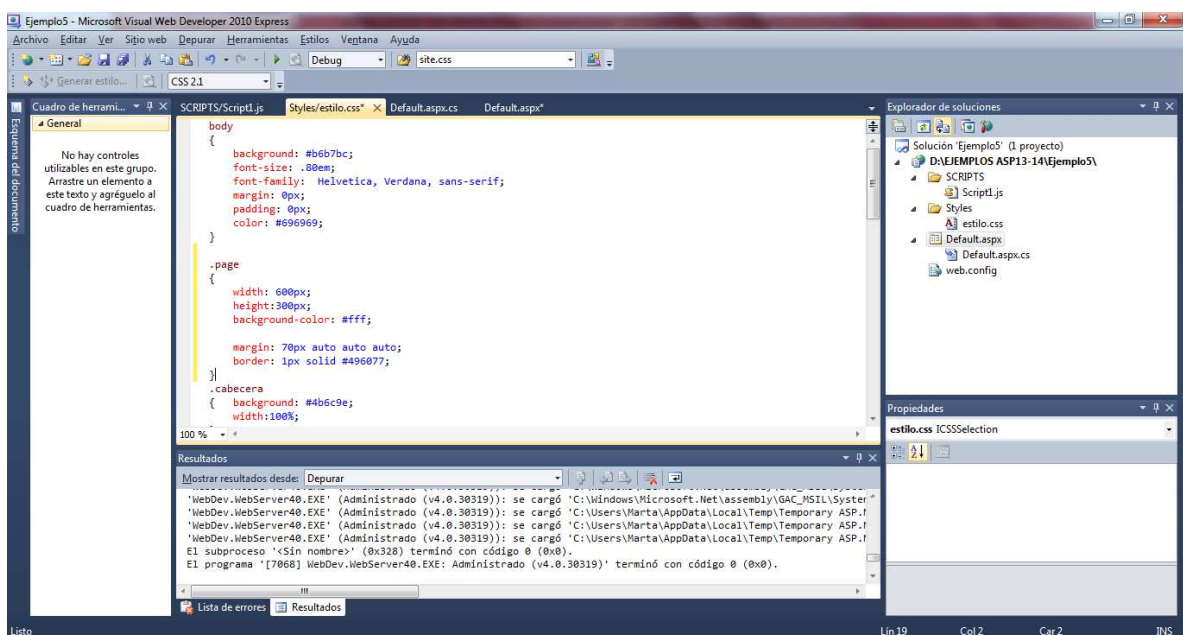
Y cogeremos una hoja de estilos:



Quedando de la siguiente forma el explorador de soluciones:



Introduciremos las reglas necesarias




```

body
{
    background: #b6b7bc;
    font-size: .80em;
    font-family: Helvetica, Verdana, sans-serif;
    margin: 0px;
    padding: 0px;
    color: #696969;
}

.page
{
    width: 600px;
    height: 300px;
    background-color: #fff;

    margin: 70px auto auto auto;
    border: 1px solid #496077;
}

```

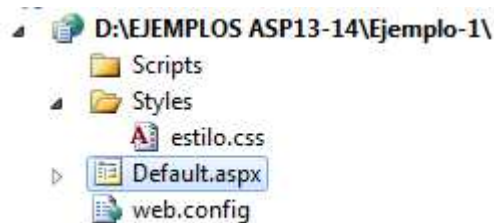
En nuestra página Default.aspx debemos introducir dentro del encabezado (<head></head>) la etiqueta que referencie a nuestra hoja de estilos en cascada:

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
|   <title>Ejemplo 1</title>
    <link rel="stylesheet" type="text/css" href = "~/Styles/estilo.css" />

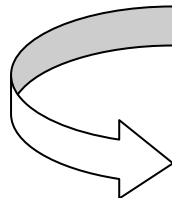
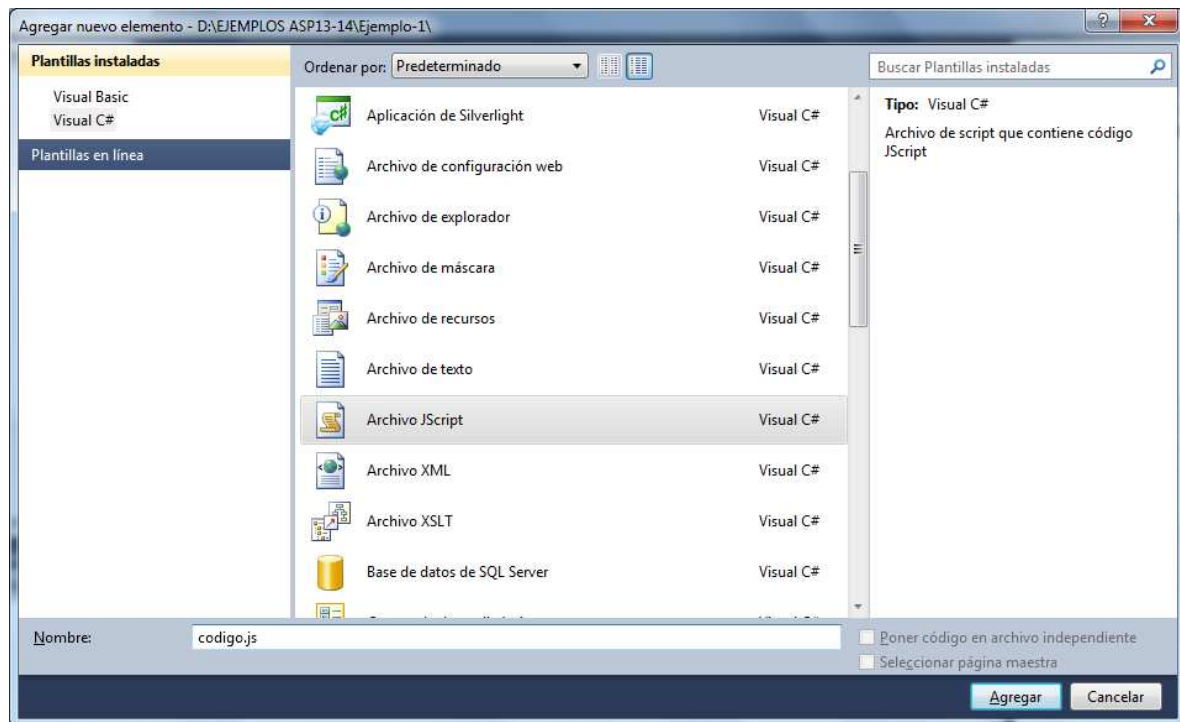
```

Ahora crearemos un **archivo de Javascript**, si queremos introducir código que sea ejecutado en el cliente. Al igual que en las hojas de estilo, crearemos una carpeta nueva denominada esta vez Scripts:

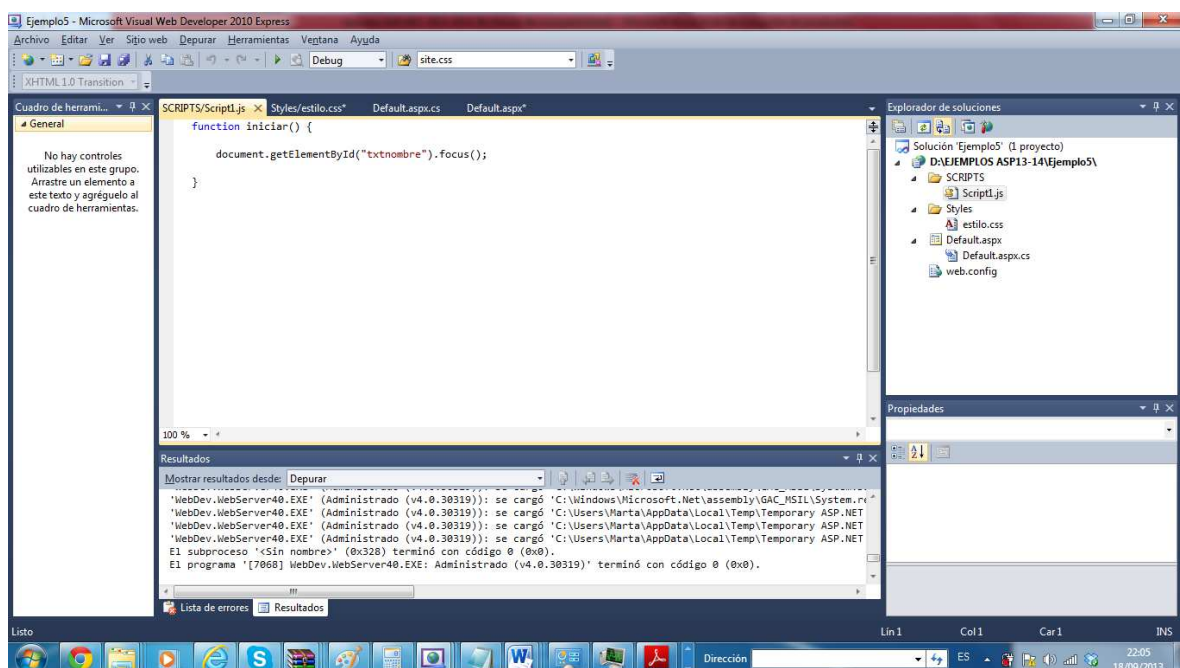


Pulsaremos en la Carpeta **Scripts** y pulsando el botón derecho elegiremos la opción Archivo JScript cambiando el nombre si nos interesa.

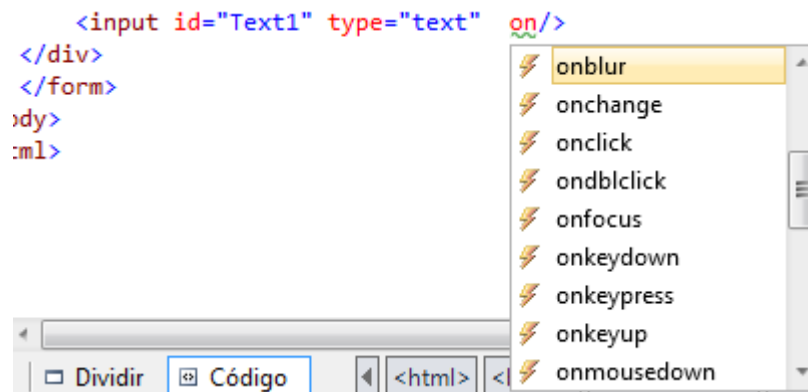




Dentro de este archivo escribiremos las funciones de **javascript** que van a ejecutarse cuando realice algún evento sobre los controles que quiero que sea ejecutado en el cliente.



Dentro de mis páginas **aspx**, en nuestro caso **Default.aspx** y dentro de los controles HTML aparecerán varios eventos que pueden ser ejecutados en el cliente **onmouseout**, **onmouseover**, **onkeyup**, **onkeydown**,....



Si estamos trabajando con controles ASP, no tendremos estos eventos dentro de la etiqueta como en los controles HTML, pero podremos añadirlos. El proceso para añadirlos es el siguiente. En la página de código **Default.aspx.cs** dentro del evento **Page_Load** (que es el evento cuando la página es cargada) debemos añadir el **evento** que nos interese al **control** que nos interese y decirle con que **función** de **javascript** lo vamos a relacionar. Esto se hace de la siguiente forma:

Control ASP.Attributes.Add (“evento”,”función-javascript”);

```
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label2.Attributes.Add("onmouseover", "iniciar()");
        Label2.Attributes.Add("onmouseout", "parar()");
    }
}
```

En este ejemplo lo que queremos conseguir es que al **pasar el ratón por encima** de la etiqueta con **ID Label2** se ejecute el código **javascript** que hayamos escrito dentro de la **función iniciar()**. Y al **sacar el ratón** de la etiqueta con **ID Label2** que se ejecute el **código javascript** de la **función parar()**.

Además en la página **Default.aspx** debemos indicar dentro del encabezado (<head></head>) que las funciones de javascript están en ese archivo. Esto lo haremos con la etiqueta

<script language="javascript" type="text/javascript" src="fichero.js"></script>

```
<head runat="server">
| <title>Ejemplo 1</title>
| <link rel="stylesheet" type="text/css" href = "~/Styles/estilo.css" />
| <script language="javascript" type="text/javascript" src = "SCRIPTS/Script1.js"></script>
| </head>
```