**2**

**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

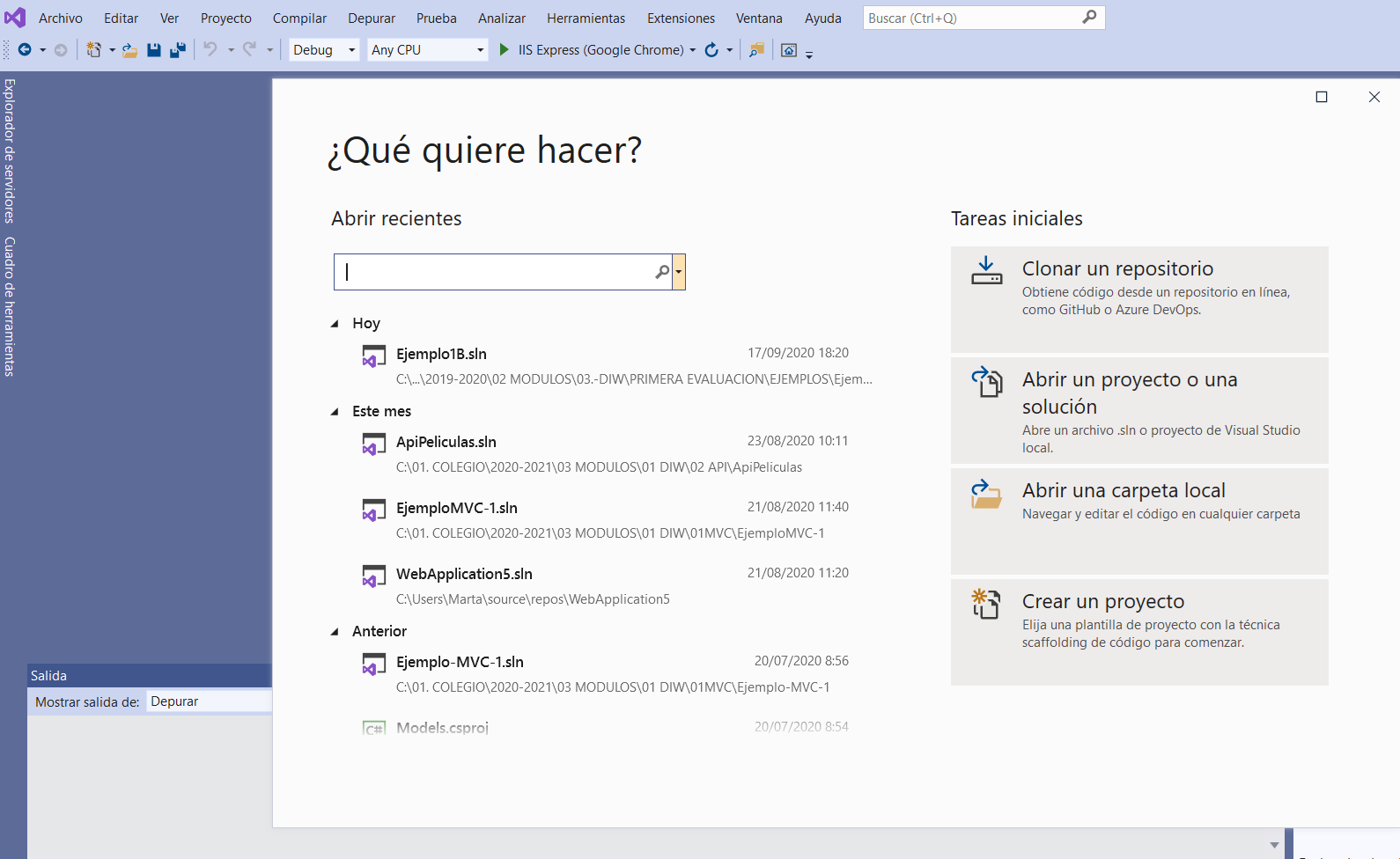
HTML) o en el servidor. Las diferencias fundamentales entre ambas son:

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

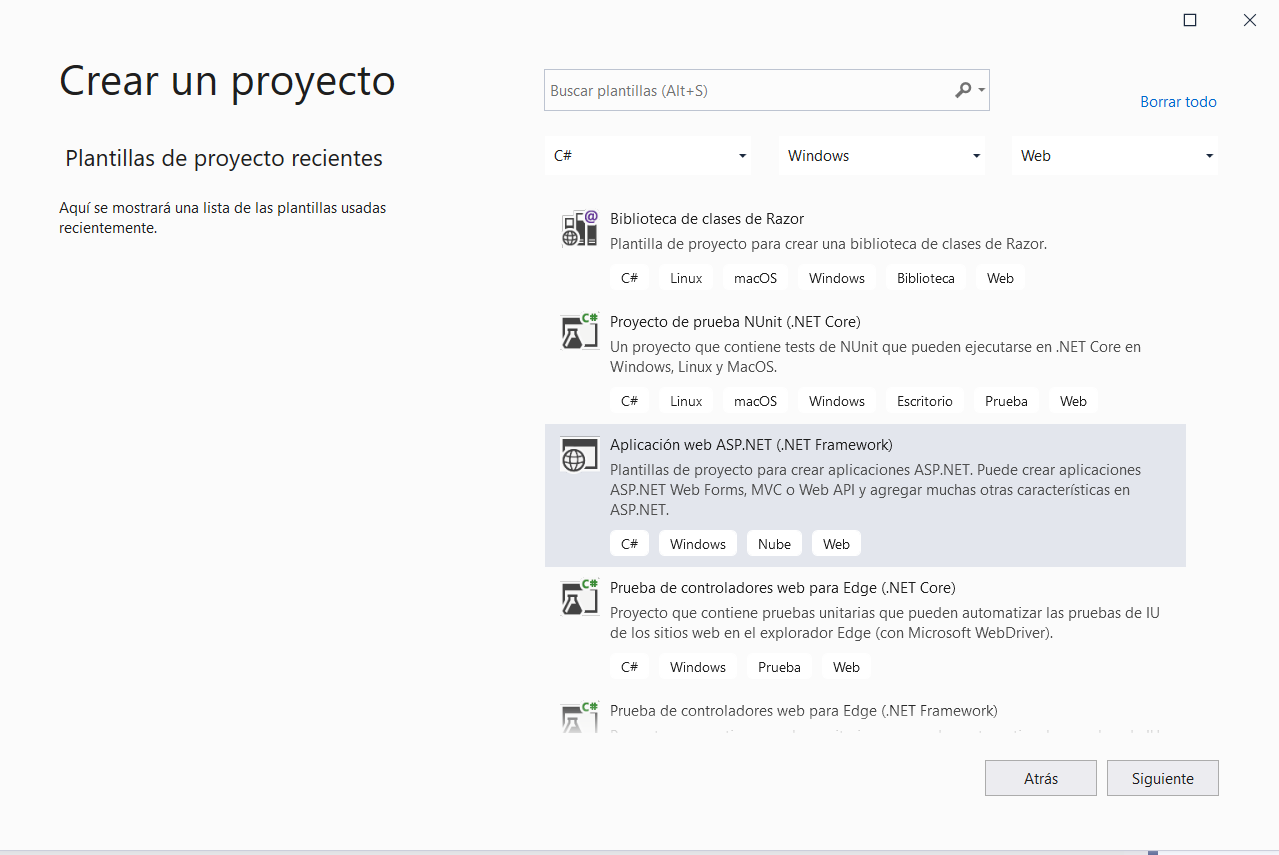
**Crear mi primera aplicación Web**

Vamos a utilizar el programa Visual Studio para poder construir nuestras aplicaciones Web. Nos los bajaremos e instalaremos.

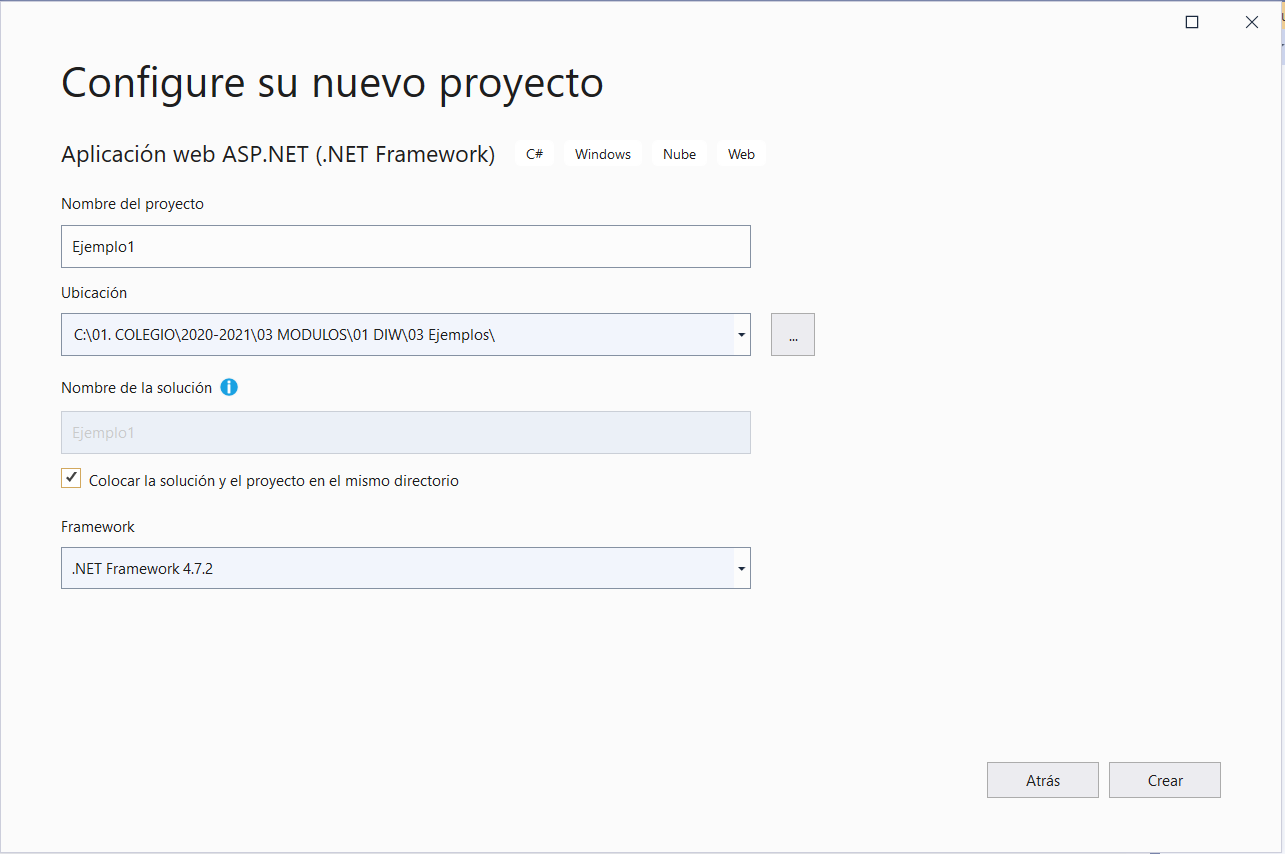
Al arrancar el Visual Studio nos encontramos con la siguiente pantalla:



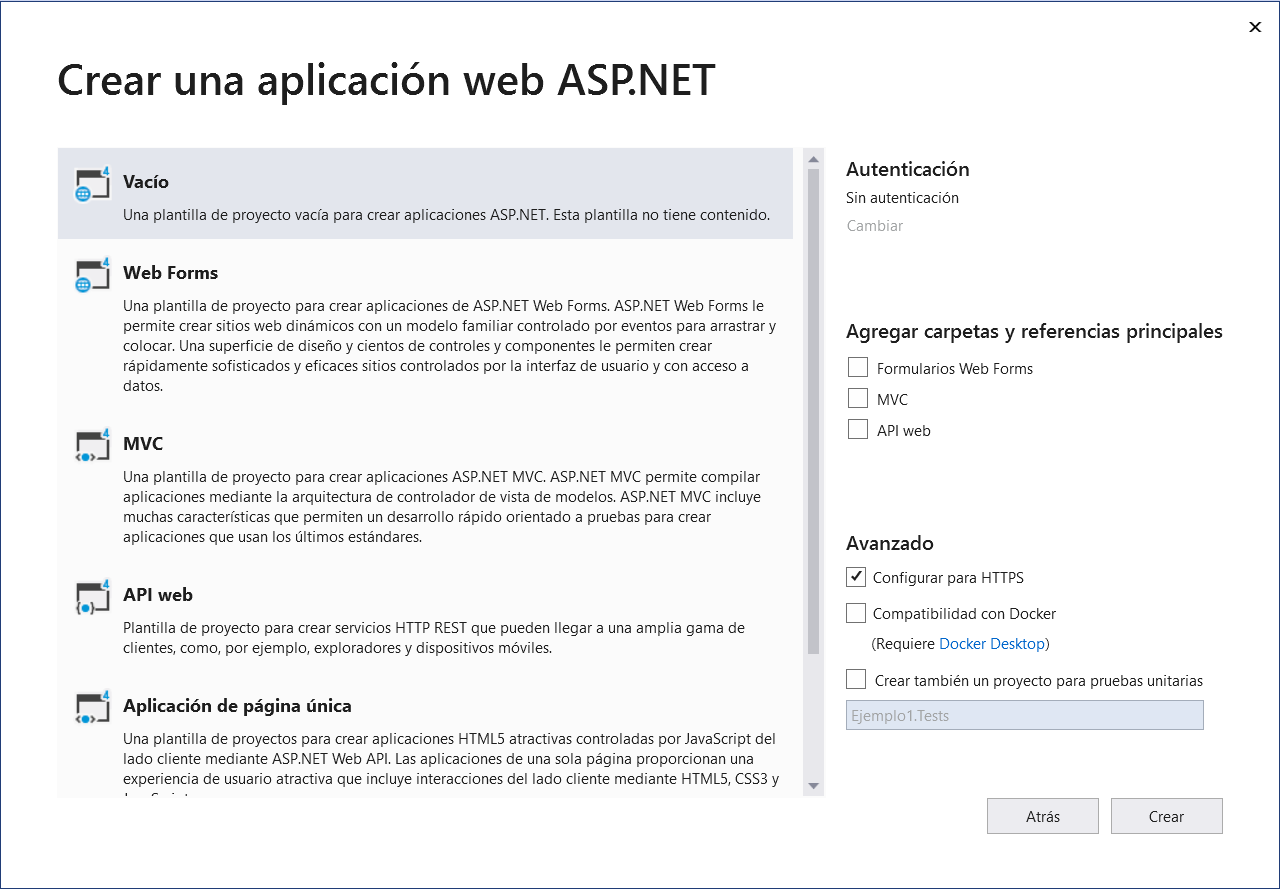
Pulsaremos **Crear un proyecto**.



Seleccionaremos **Aplicación ASP.NET (.NET Framework)** y pulsaremos **Siguiente:**

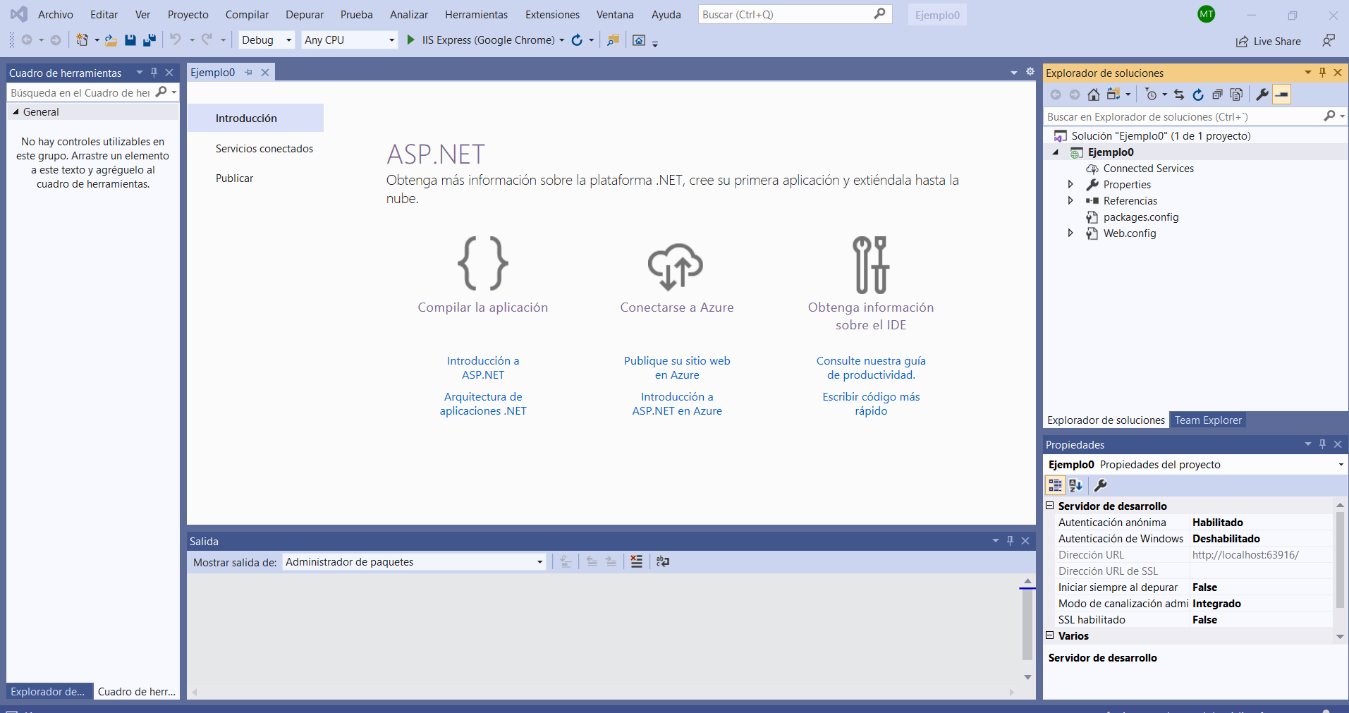


Pondremos **el nombre del Proyecto**, seleccionaremos la **ubicación** para su almacenamiento y pulsaremos la casilla para **colocar la solución y el proyecto en el mismo directorio**.



Elegiremos la plantilla del **proyecto vacío** y pulsaremos **Crear**.

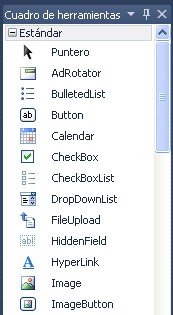
Se mostrará la siguiente pantalla:

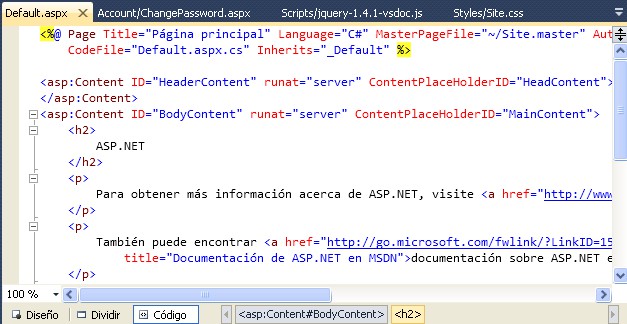


Ahora toca empezar a desarrollar nuestra primera aplicación.

**Vamos a conocer el entorno.** Nos encontraremos las siguientes ventanas:

Empezando de izquierda a derecha:

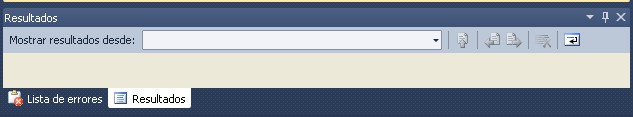
**Cuadro de Herramientas**: donde encontraremos todos los controles disponibles para utilizar en nuestras páginas Web. Están agrupados en diferentes categorías.

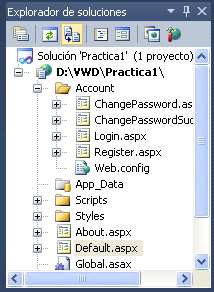
**Ventana Principal**: en donde podremos ver el código o diseño de nuestras páginas. En la parte de debajo de esta ventana están las pestañas Diseño, Dividir y código desde donde podemos cambiar de una a otra visión. La pestaña dividir mostrará la ventana dividida mostrando el diseño y el

código a la vez.

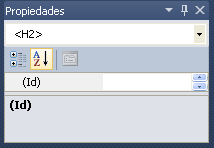
**Ventana de Resultados**: en

donde se mostrará cómo 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, capetas para imágenes,….

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

nuestros controles.

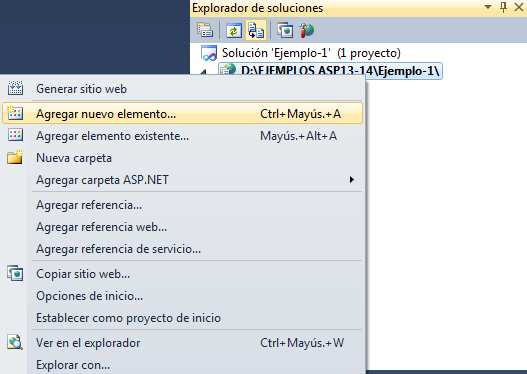
Cada una de estas ventanas podemos ocultarla, cerrarla o volver a abrirla desde el menú Ver

Otras ventanas.

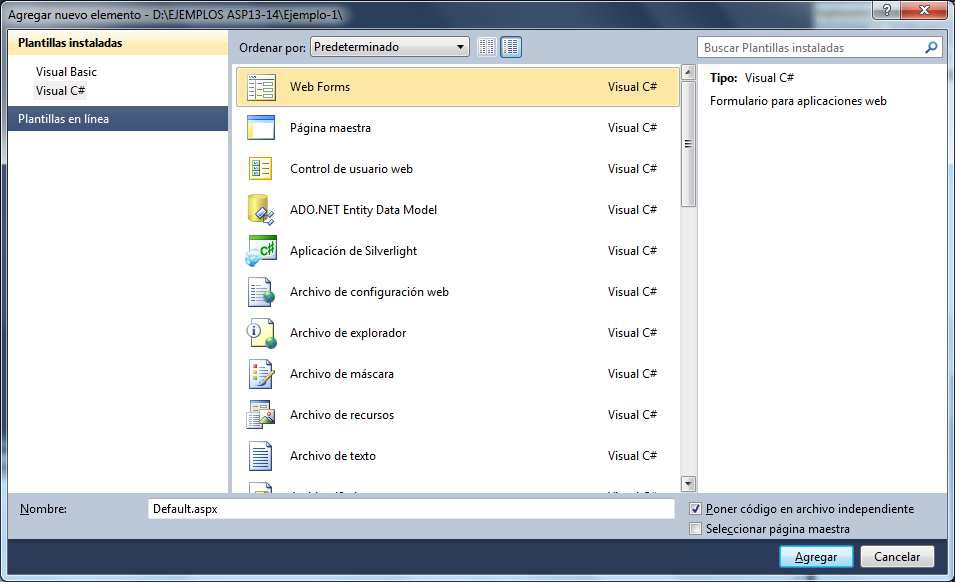
Comencemos con el Proceso. Seguiremos los siguientes pasos:

**1º Añadir un Web Form (Página .aspx)**

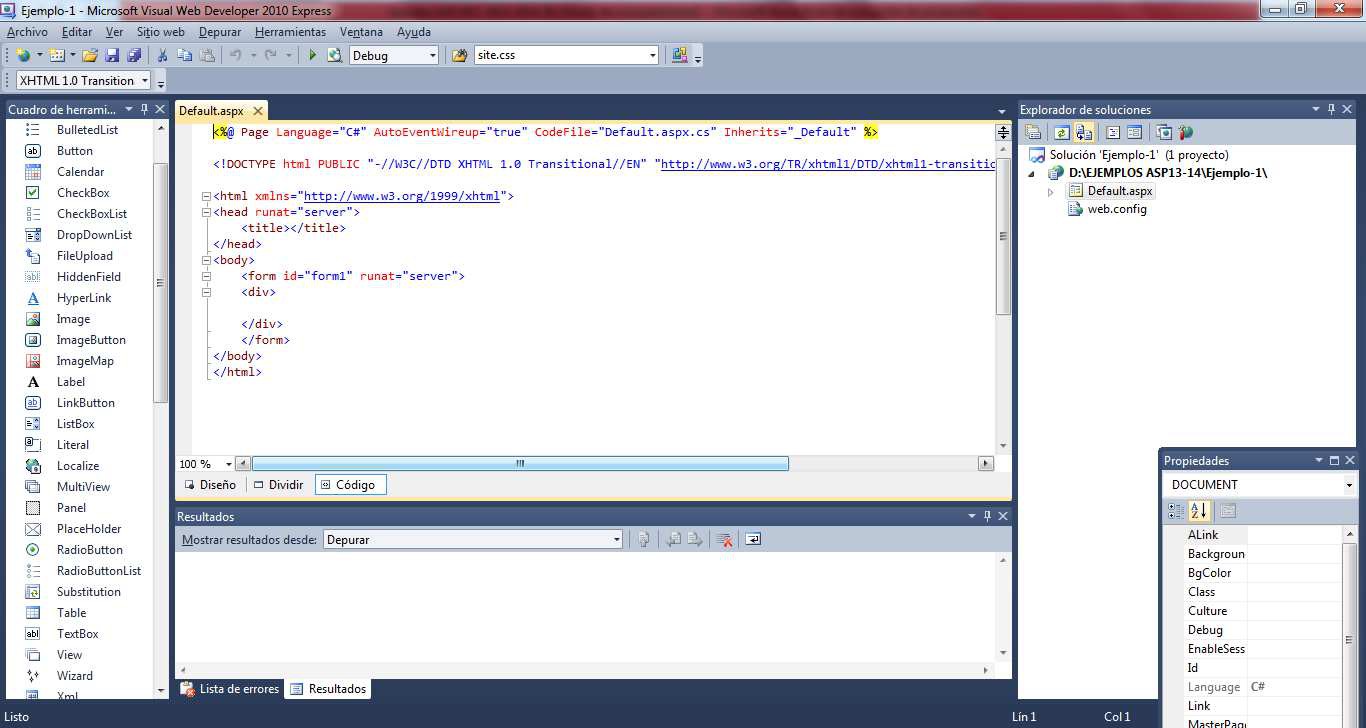
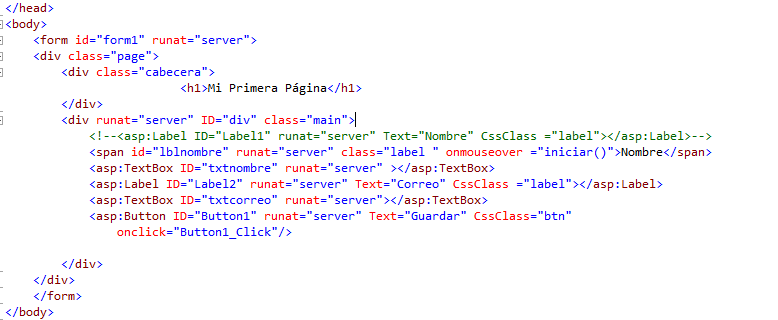
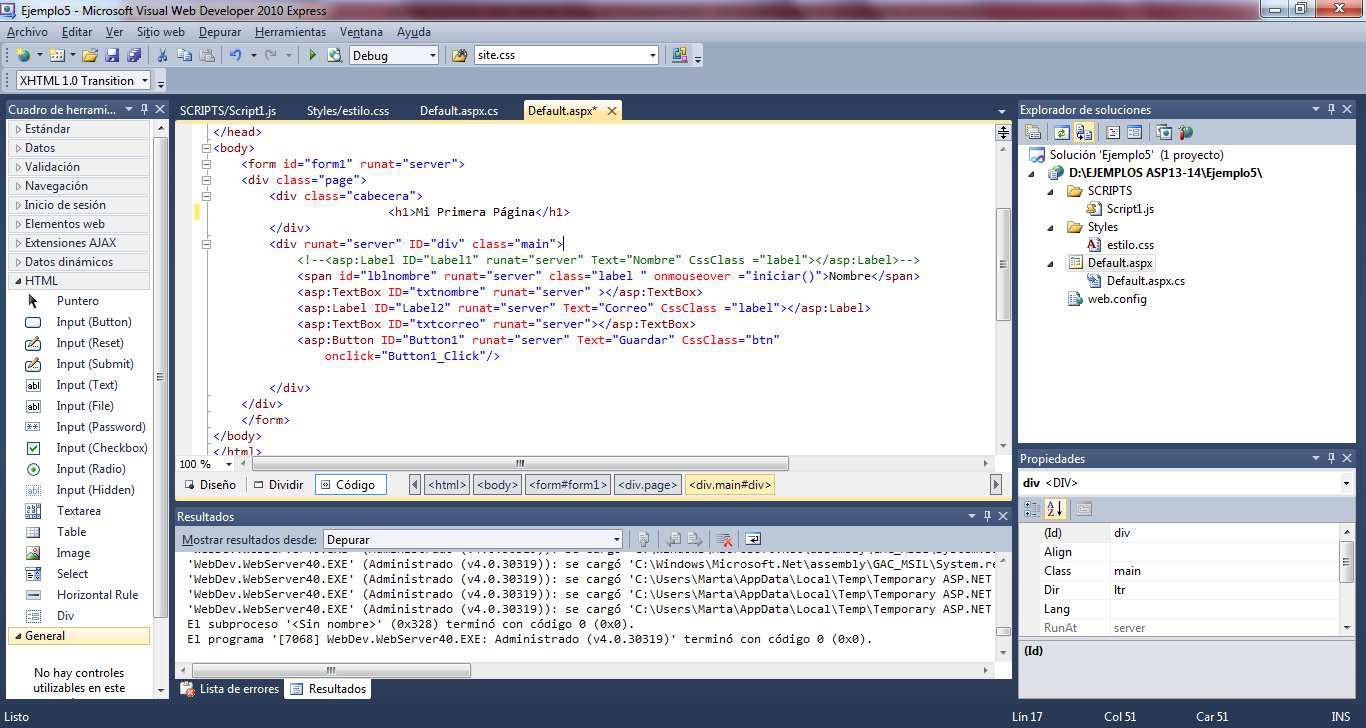
Pulsaremos botón derecho sobre el sitio Web en el explorador de soluciones y pulsaremos la opción de Añadir un nuevo elemento:



Se mostrará el siguiente cuadro de diálogo



En el que pulsaremo 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:

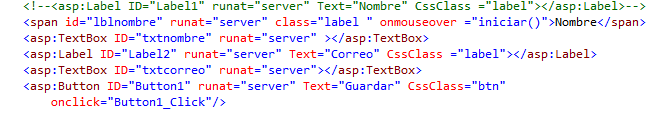
**2º Crear la Hoja de Estilos.**

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

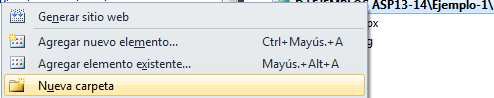
CSS con el atributo **class**.



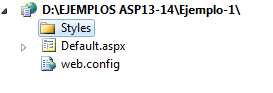
Sin embargo si el control es **ASP** las clases de CSS se asignan mediante el atributo **CssClass**.



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



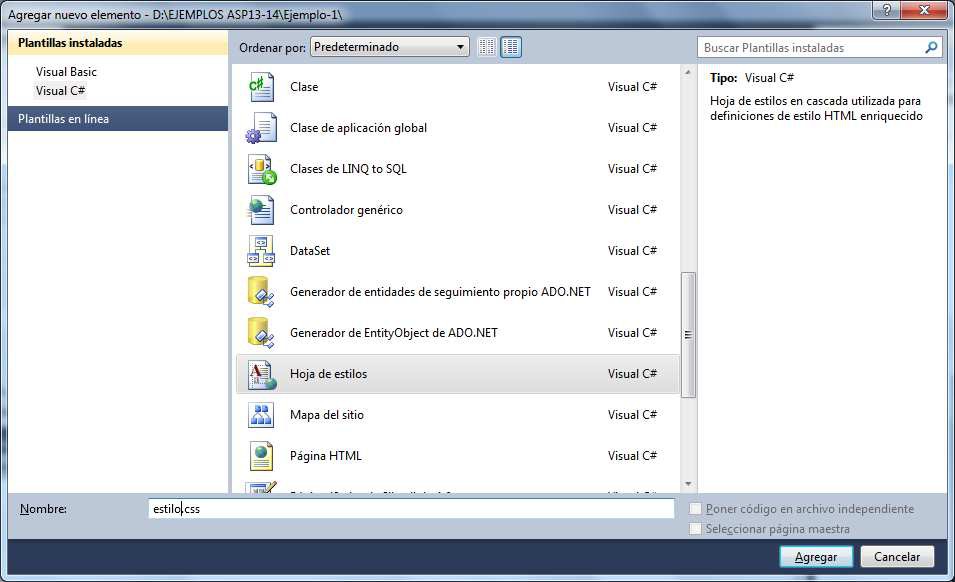
Quedará así:



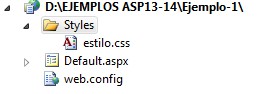
Pulsaremos Agregar nuevo elemento:



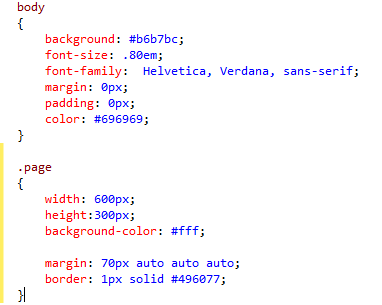
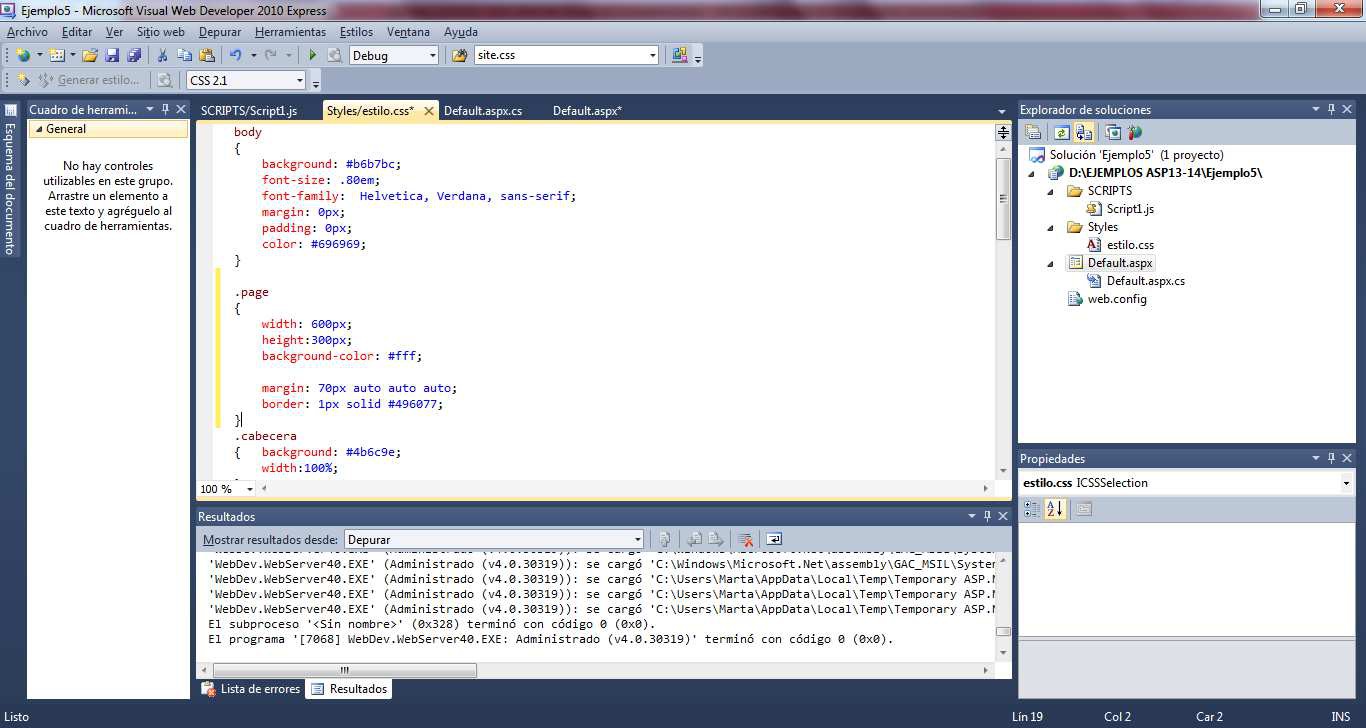
Y cogeremos una hoja de estilos:



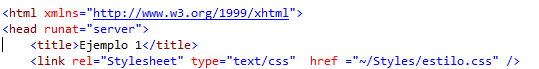
Quedando de la siguiente forma el explorador de soluciones:



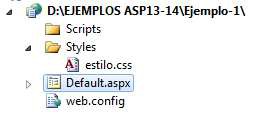
Introduciremos las reglas necesarias



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

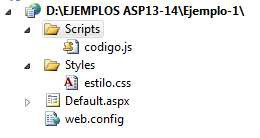
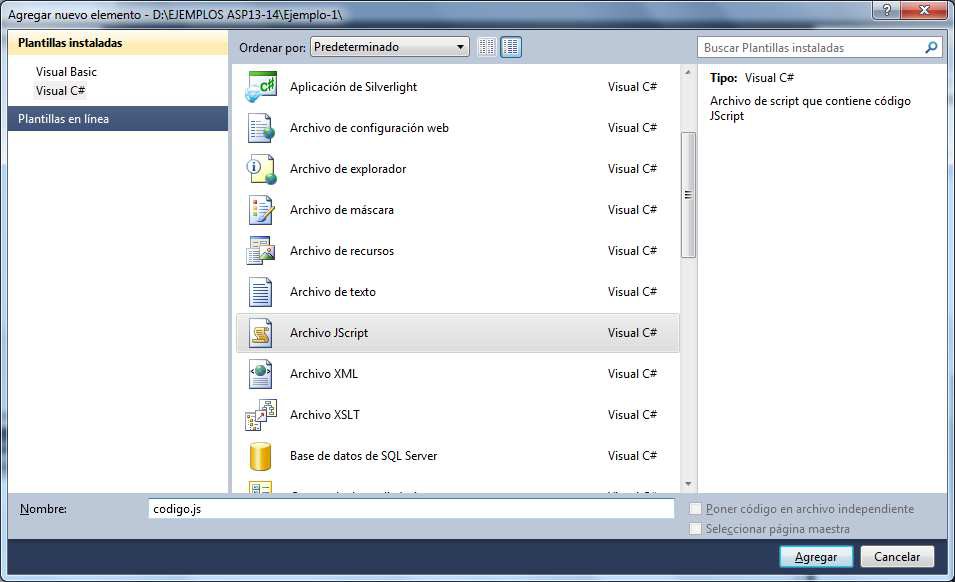


Ahora crearemos un **archivo de Javascript**, si queremos introducir código que sea ejecutado en el cliente. Al igual que en las hojas de estilo, crearemos una carpeta nueva denominada esta vez Scripts:

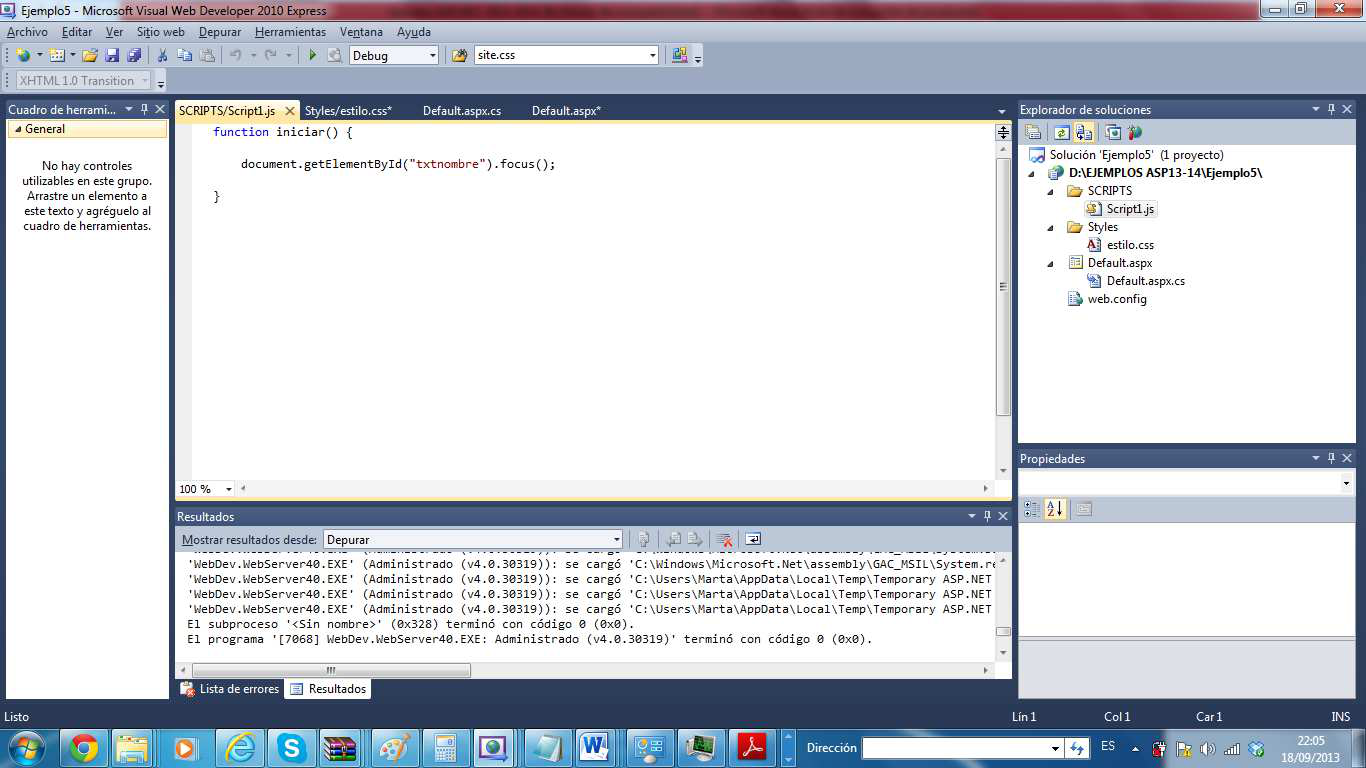


Pulsaremos en la Carpeta **Scripts** y pulsando el botón derecho elegiremos la opción Archivo

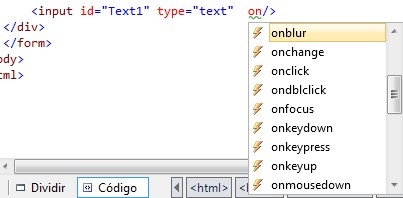
JScript cambiando el nombre si nos interesa.



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.

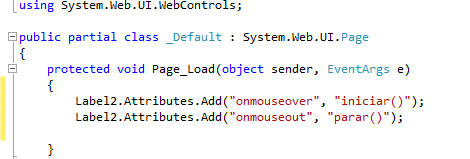


Dentro de mis páginas **aspx**, en nuestro caso **Default.aspx** y dentro de los controles HTML aparecerán varios eventos que pueden ser ejecutados en el cliente **onmouseout**, **onmouseover**, **onkeyup**, **onkeydown**,….



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);**



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 enese archivo. Esto lo haremos con la etiqueta

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

