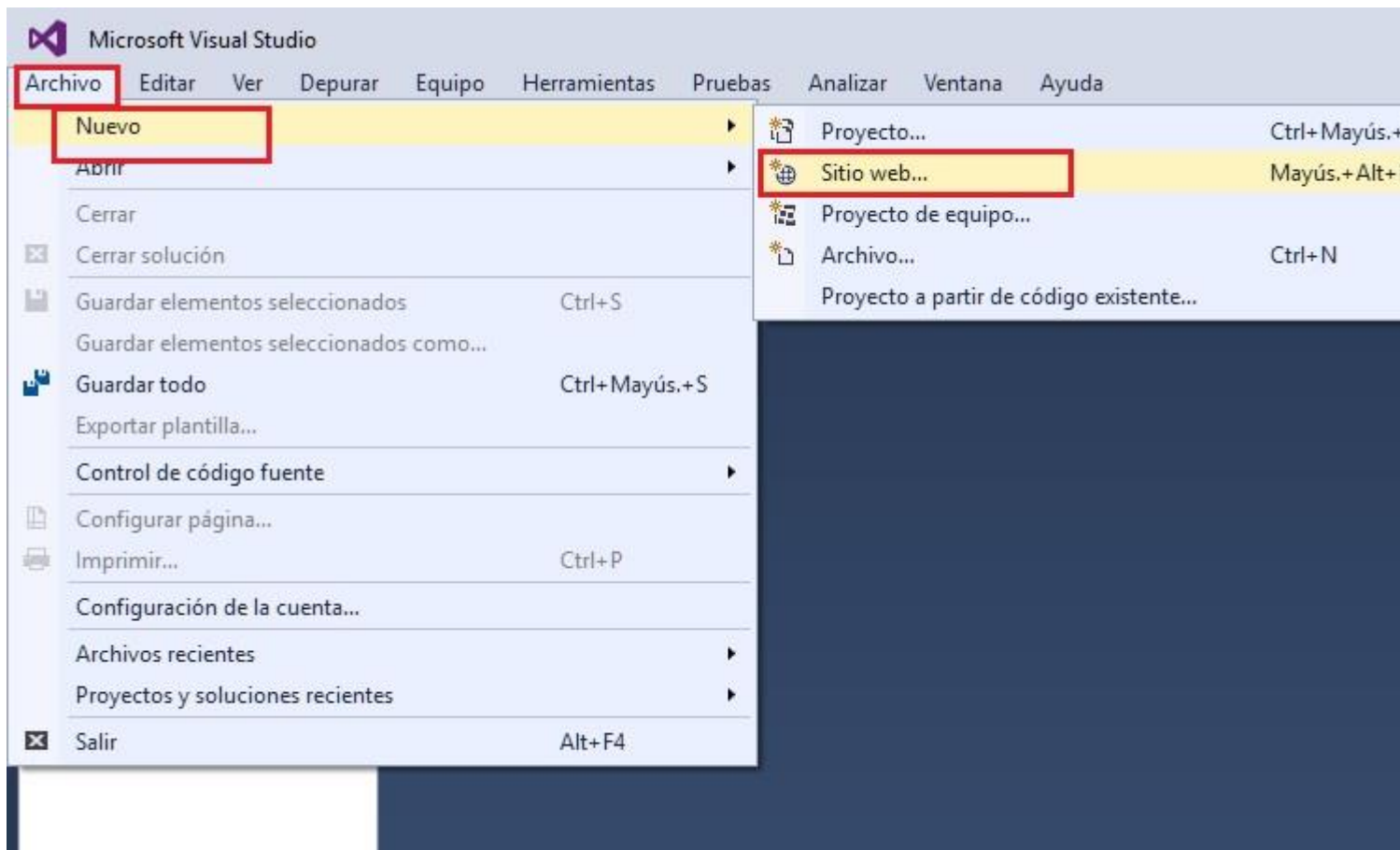


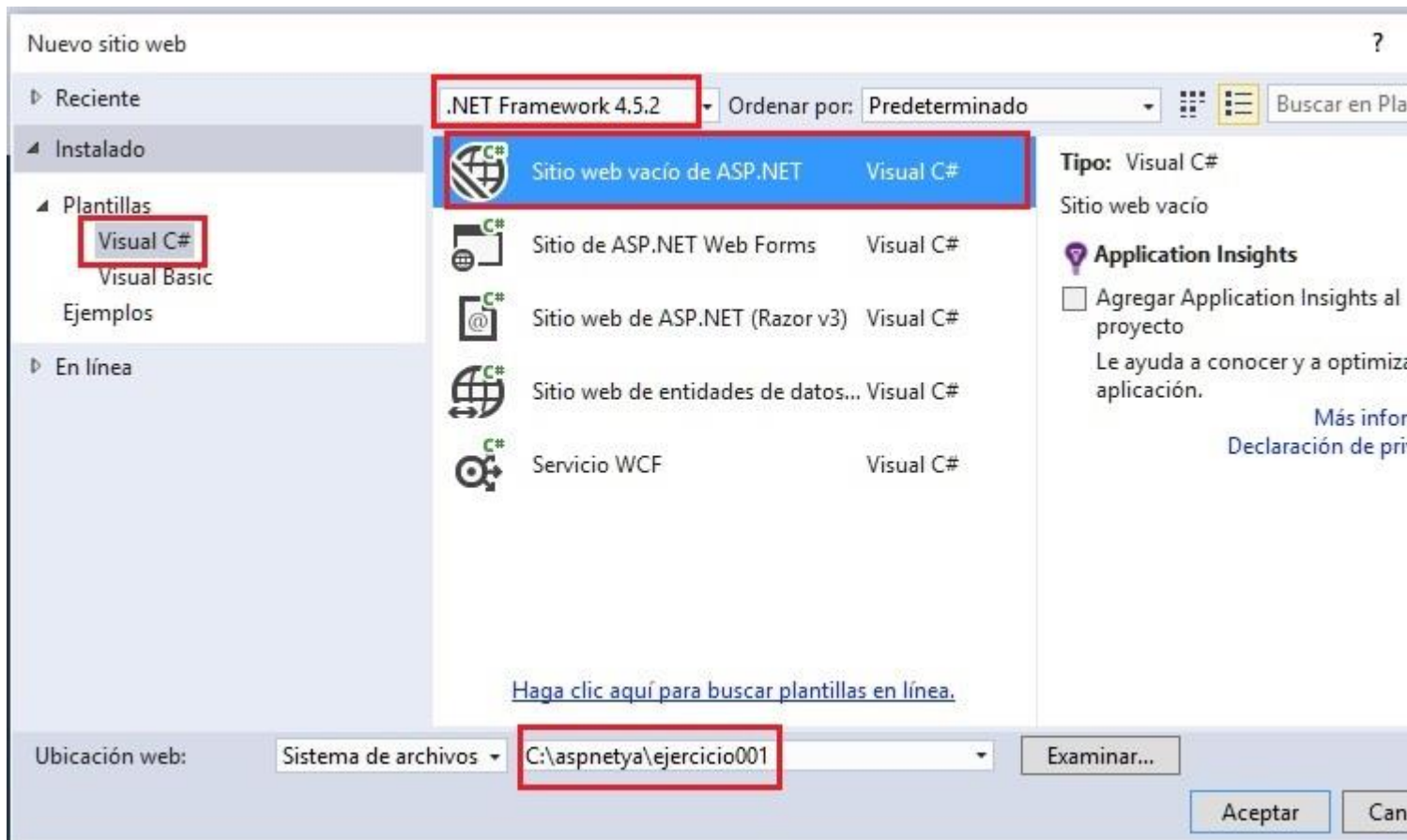
Pasos para crear un sitio web ASP.Net con el Visual Studio 2015 con C#

Conceptos.

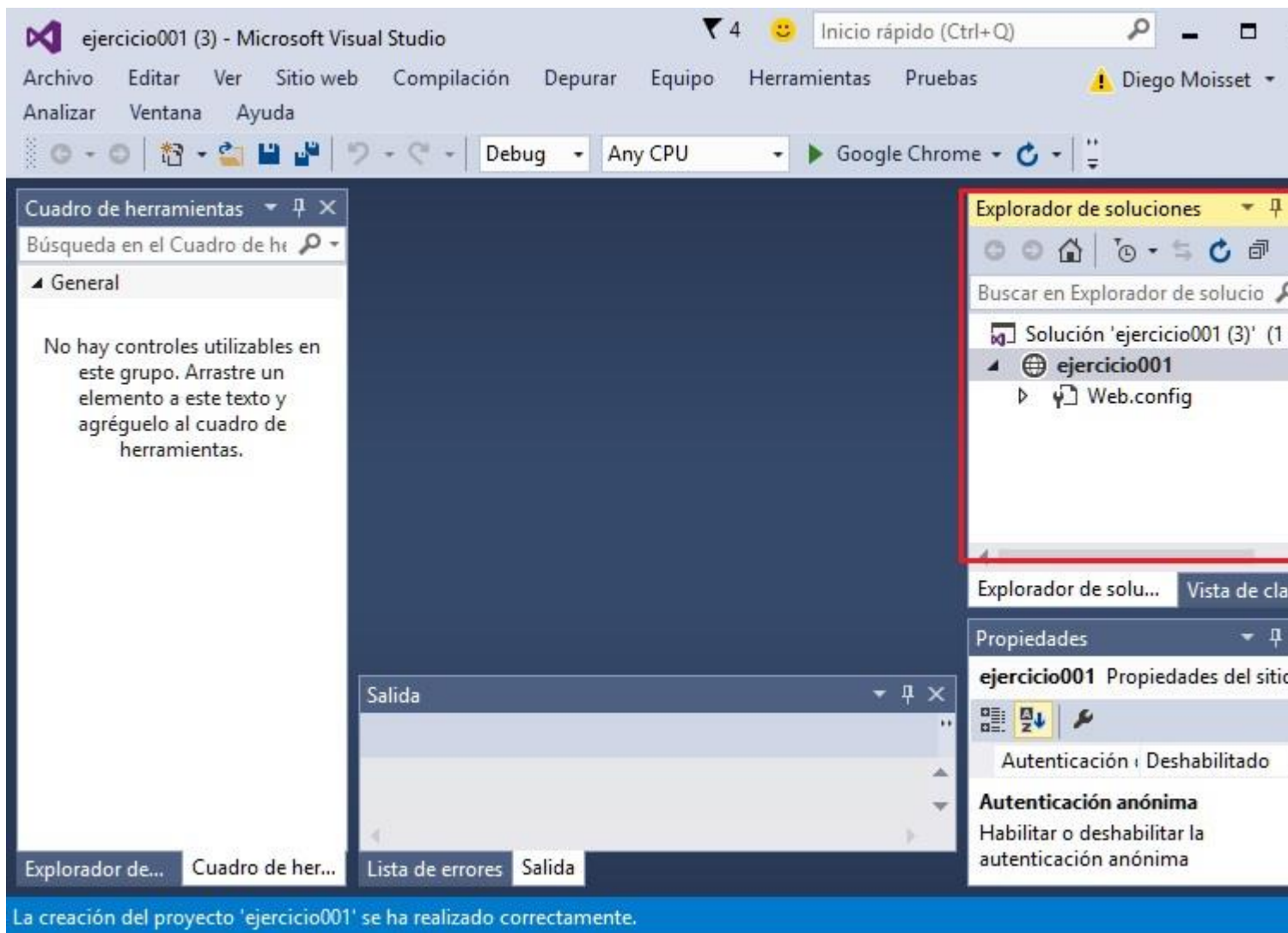
Para crear un sitio web debemos seleccionar desde el entorno del Visual Studio 2015: Archivo ->Nuevo sitio web:



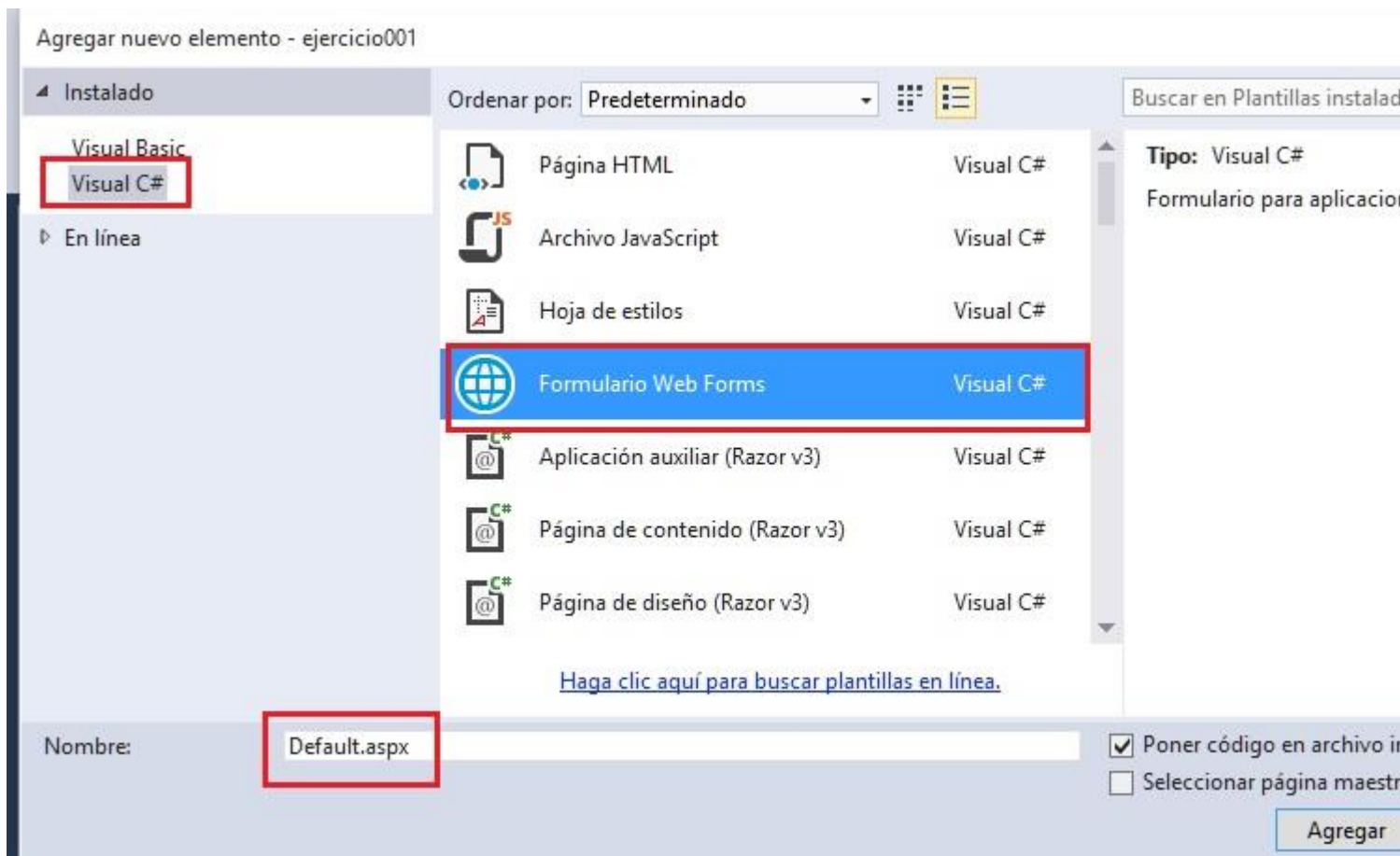
Inmediatamente aparece un diálogo donde debemos seleccionar el lenguaje a utilizar (Visual C#) la versión de Framework (por ejemplo 4.5.2), elegiremos crear "Sitio web vacío de ASP.NET" y la carpeta donde se almacenarán todas las páginas de nuestro directorio (por ejemplo c:\aspnetya\ejercicio001):



Ahora tenemos nuestro primer sitio web vacío, solo se ha creado un archivo de configuración llamado Web.config:



Creemos nuestra primer "Web Form" para esto debemos presionar el botón derecho del mouse donde aparece el nombre de nuestra aplicación "ejercicio001" y en este menú seleccionar Agregar -> Agregar nuevo elemento...:



En la parte izquierda tenemos el ?Cuadro de herramientas?, en esta aparecen las componentes visuales (Label, TextBox, Button etc.) que tenemos disponibles para crear nuestro formulario Web.

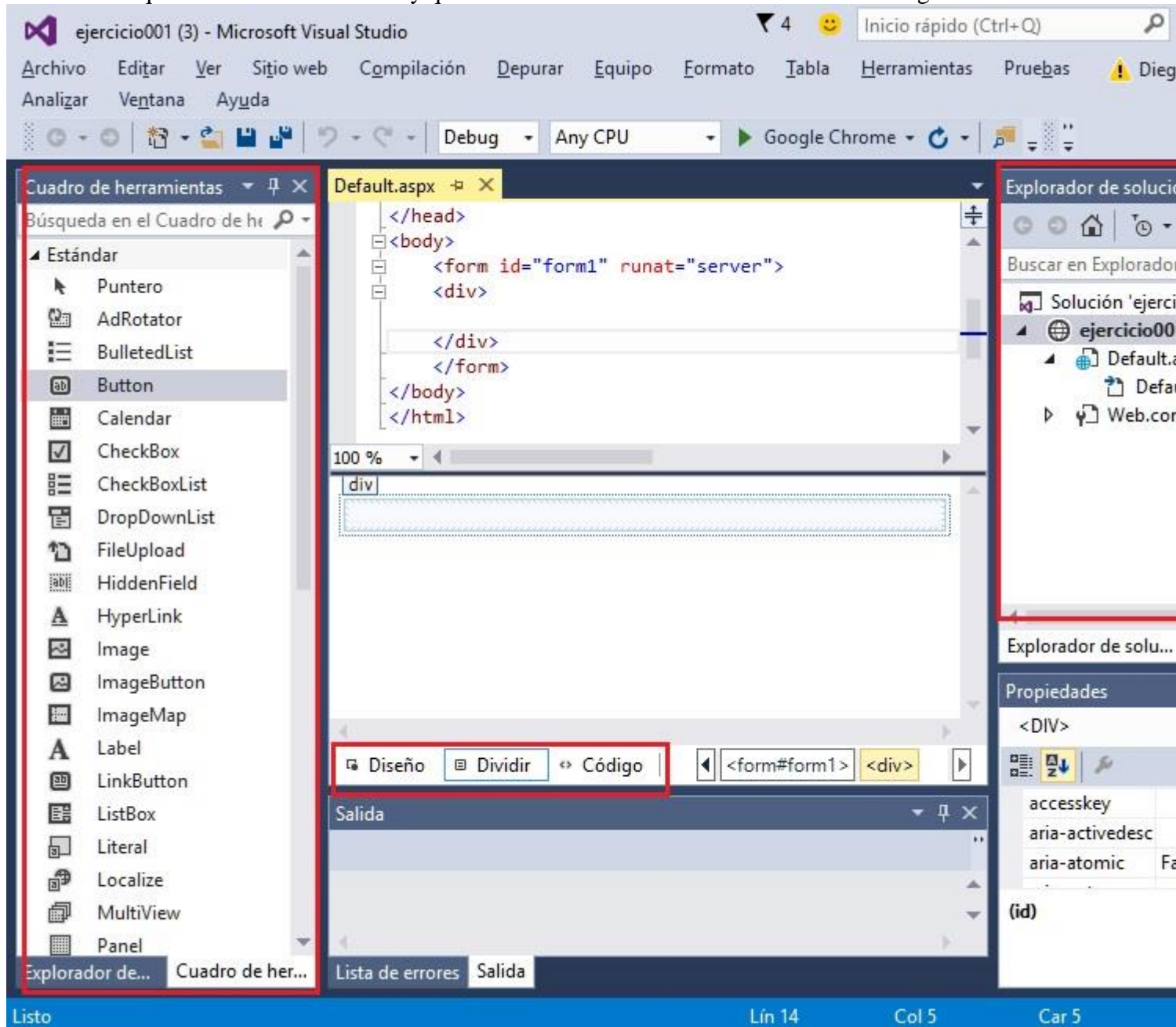
En el centro aparece la página en vista de código (se puede ver en ?vista de diseño?, ?Dividir? y ?Código?)

En la parte derecha disponemos del ?Explorador de soluciones? donde podemos identificar el nombre de nuestro sitio web y los archivos contenidos en dicho sitio.

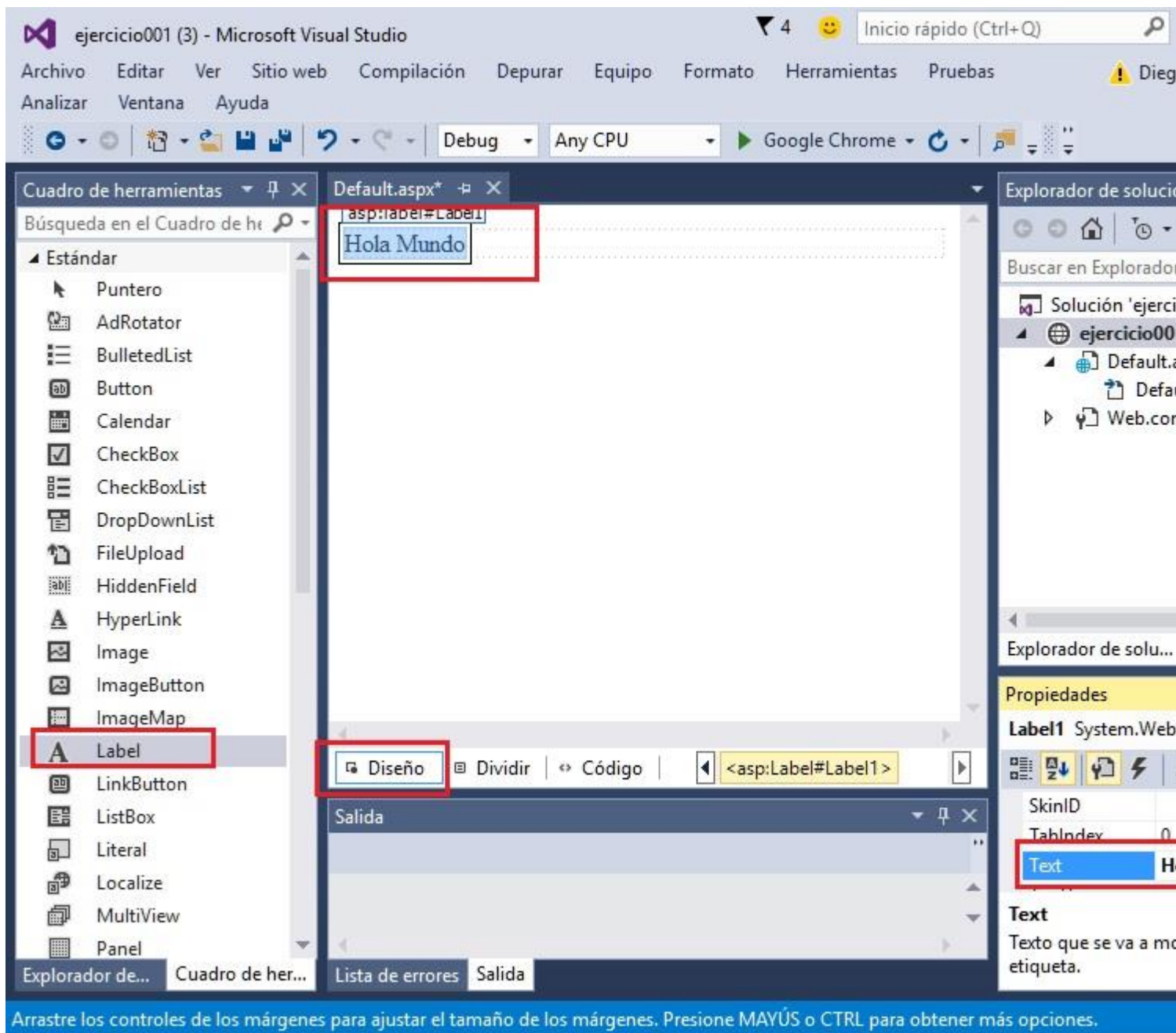
Siempre que creamos una aplicación web debemos crear al menos un formulario web inicial y lo almacenamos en el archivo Default.aspx (la extensión aspx indica que se trata de una página dinámica ASP.Net, así como la extensión php indica que su contenido está programado en PHP)

Además del archivo Default.aspx se crea otro archivo llamada Default.aspx.cs (éste archivo contiene la codificación en C# de los eventos que definamos a los controles del formulario)

Otro archivo que veremos más adelante y que se crea en forma automático es el Web.config.



Para nuestro primer ejemplo implementaremos el '¿Hola Mundo?' para ello en el recuadro central seleccionamos la pestaña 'Diseño' y desde el cuadro de herramientas arrastramos un control de tipo Label. Seguidamente desde el cuadro de propiedades ubicado en la parte inferior derecha de la pantalla inicializamos la propiedad text con el mensaje que queremos que muestre nuestra '¿Label?', en nuestro caso '¿Hola Mundo?'. Una vez modificada la propiedad Text con el mensaje que queremos mostrar y presionada la tecla Enter podemos ver como se actualiza la ventana de Diseño en la parte central de nuestra pantalla:



Para probar nuestra pequeña aplicación desarrollada debemos presionar el triángulo verde que se encuentra en la barra de botones, o desde el menú de opciones: Depurar->Iniciar depuración, o presionar la tecla ?F5?. Podemos ver que inmediatamente aparece el navegador configurado por defecto con el resultado de la ejecución de la página:



El Visual Studio 2015 instala un servidor web propio que está escuchando en un puerto desocupado. Luego de cerrar la ventana del navegador debemos detener la depuración de nuestra aplicación para poder modificarla, para esto podemos seleccionar desde el menú Depurar -> Detener Depuración o presionar desde la barra de botones el cuadradito morado (luego de esto aparece el ?Cuadro de herramientas?)

Eventos

Modificaremos ahora nuestra aplicación para que muestre la fecha del servidor en una Label. Cuando desde el navegador solicitamos una página aspx lo primero que se ejecuta es el evento Page_Load. Para poder definir un método para dicho evento hacemos doble clic sobre el WebForm con lo que inmediatamente se abre el archivo Default.aspx.cs donde tenemos dicho método:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

Luego codificamos dentro del método Page_Load el algoritmo que muestra la fecha actual del servidor:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
```



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

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Today.ToString("d");
    }
}
```

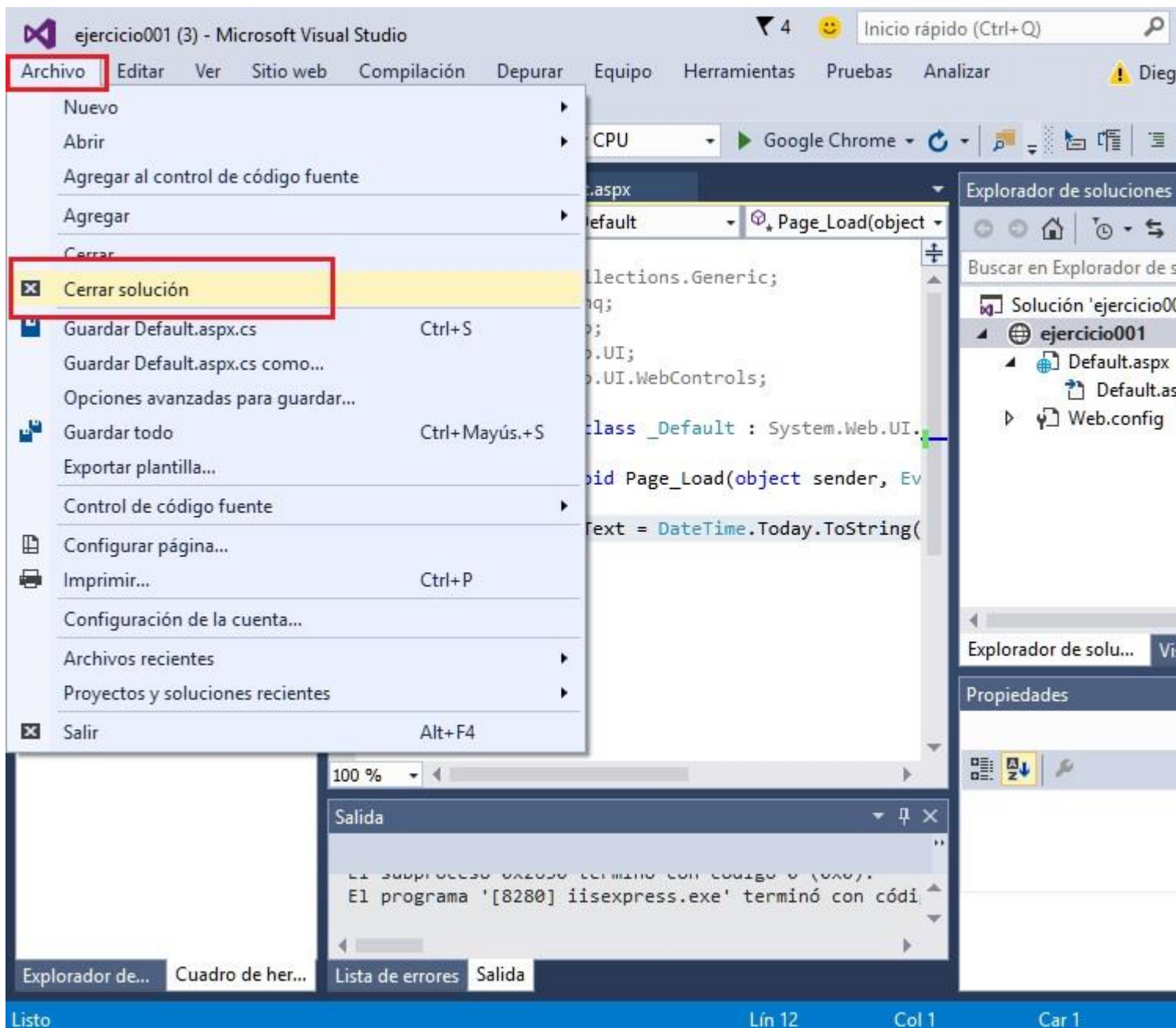
Mediante el objeto `DateTime` y accediendo a la propiedad `Today` y llamamos al método `ToString` pasando un string con una "d" lo que hace que dicho método nos retorne el día, mes y año.

Si ejecutamos ahora la aplicación en el navegador se verá reflejada la hora fecha definida en el servidor web:

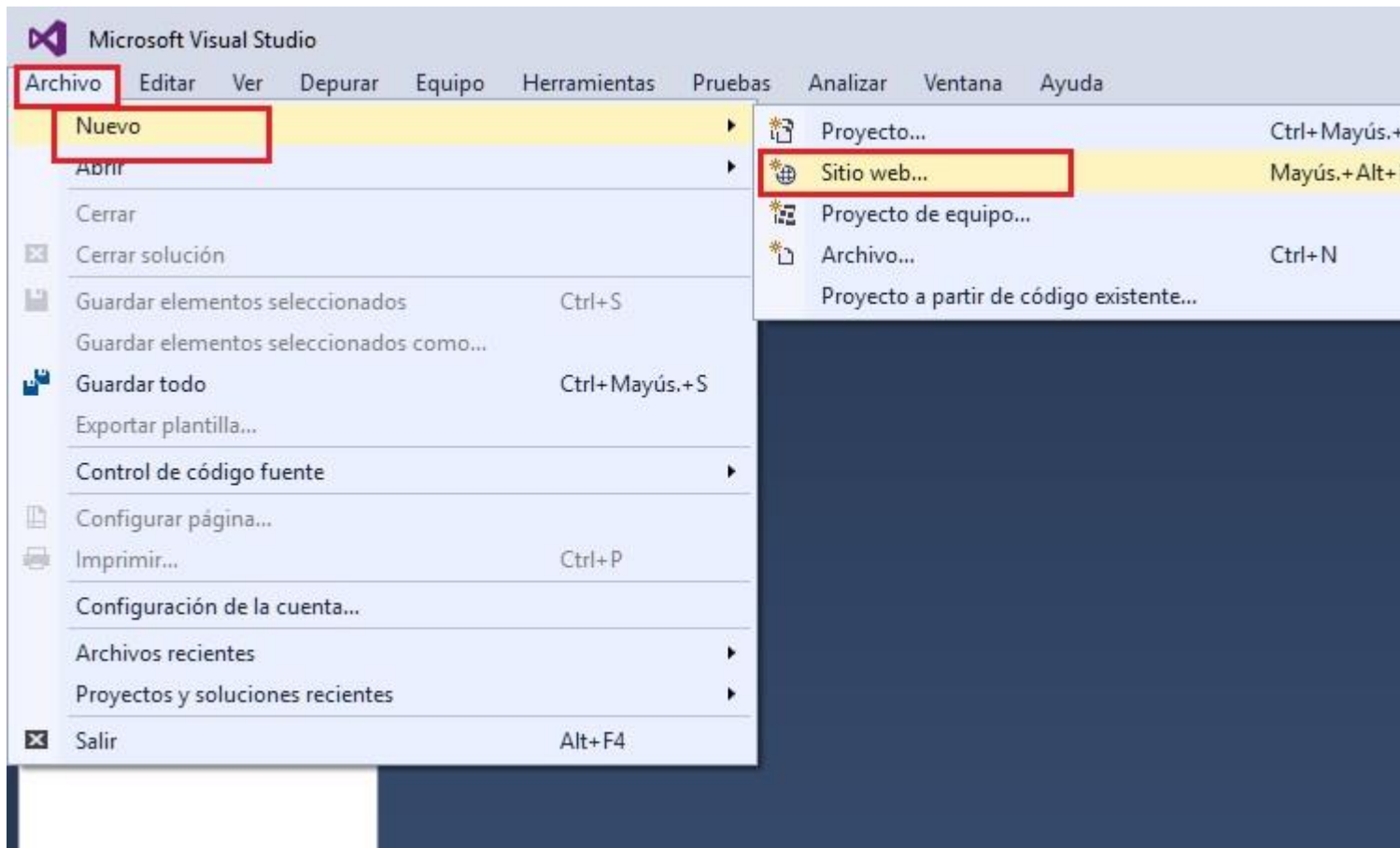


Captura del evento click de un objeto de la clase `Button`.

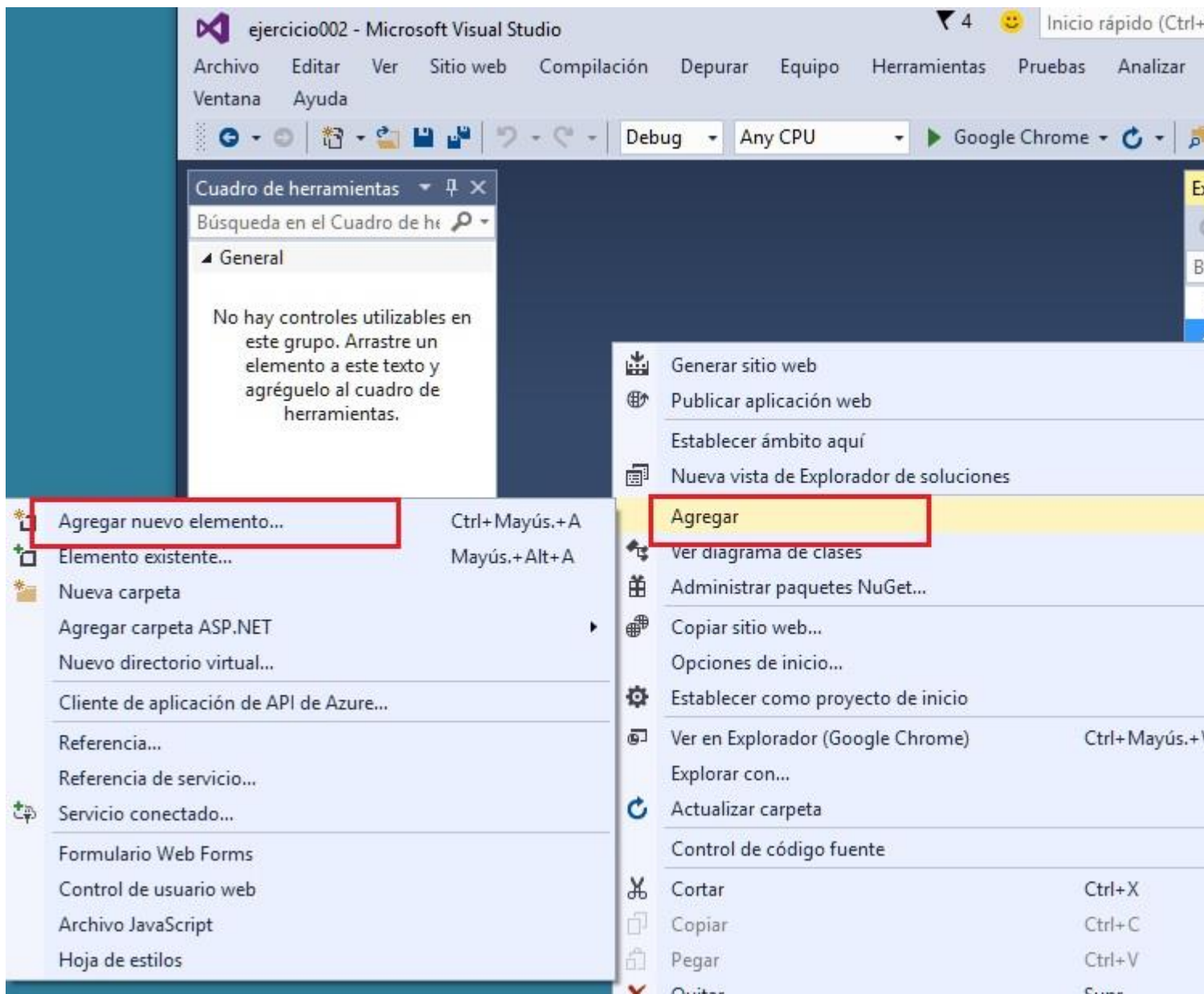
Para ver como funciona la clase `Button` cerraremos nuestro primer sitio web y procederemos a crearemos otros. Para cerrar el sitio actual desde el menú de opciones elegimos Archivo -> Cerrar Solución:



Ahora damos los mismos pasos que hicimos para crear el sitio ejercicio001 pero lo llamamos ejercicio002 (almacenarlo en la carpeta c:\aspnetya\ejercicio002):

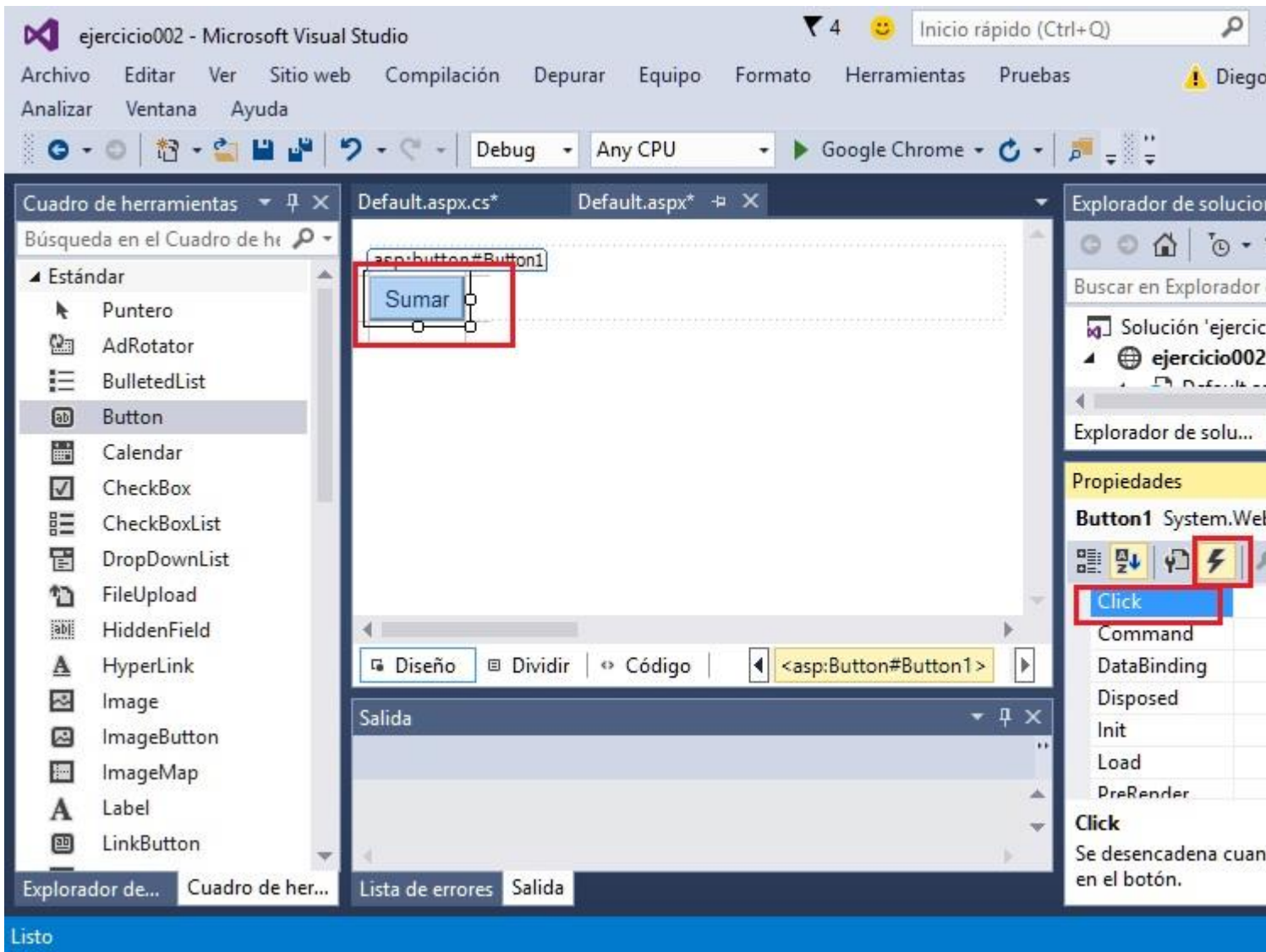


Creamos también el Web Form Default.aspx:



En nuestro Web Form disponemos un objeto de la clase Label y un objeto de la clase Button. La propiedad Text de la Label la inicializamos con el valor ?0? y la propiedad Text del objeto Button lo inicializamos con la cadena "Sumar". El objetivo es que cada vez que se presione el botón se actualice el contenido de la Label con el valor actual más uno.

La forma más sencilla de generar dicho evento es hacer doble clic sobre el objeto Button. Otra forma es seleccionar dicho objeto (Button) y en la ventana de propiedades seleccionamos el ícono del "rayo" y hacer doble clic sobre el evento Click:



Luego para el evento Button1_Click actualizamos el contenido de la propiedad Text de la Label con el valor actual más uno:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        int x = int.Parse(Label1.Text);
        x++;
        Label1.Text = x.ToString();
    }
}
```

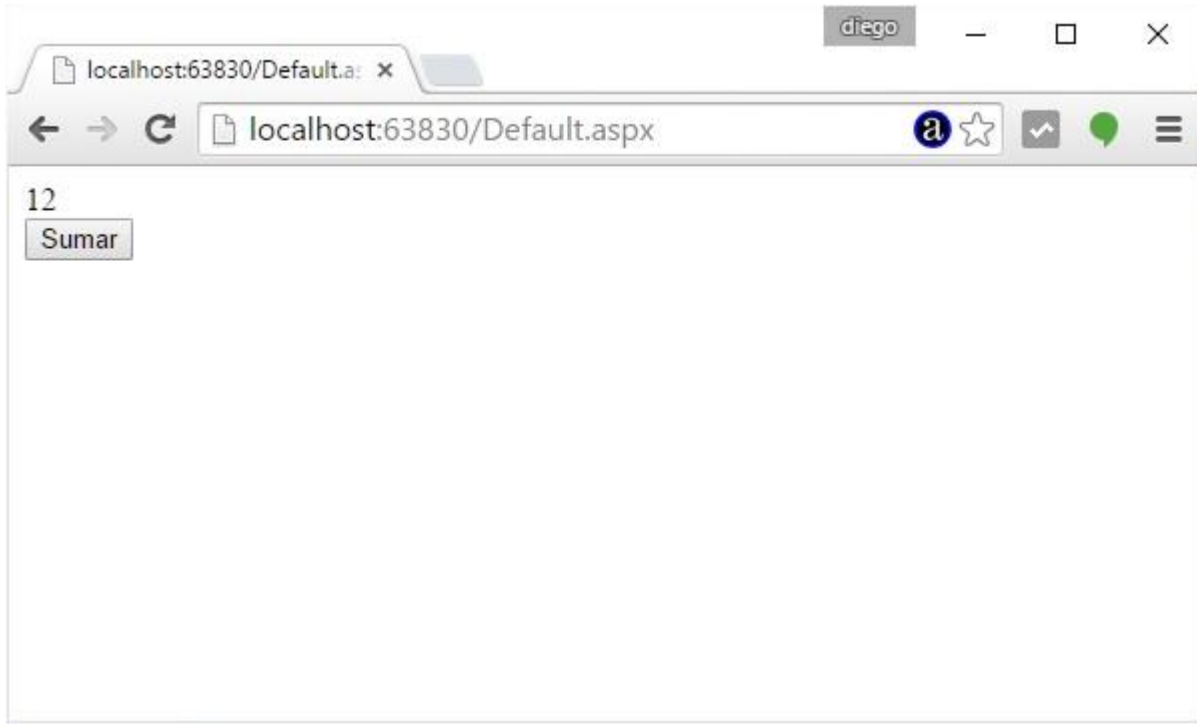
Debemos convertir a entero el contenido de la Label mediante el método estático Parse de la clase int:


```
int x = int.Parse(Label1.Text);
```

Incrementamos en uno la variable x y la volvemos a mostrar en la Label:

```
x++;  
Label1.Text = x.ToString();
```

Cuando ejecutamos la página cada vez que presionamos el botón se incrementa en uno el contenido de la Label:

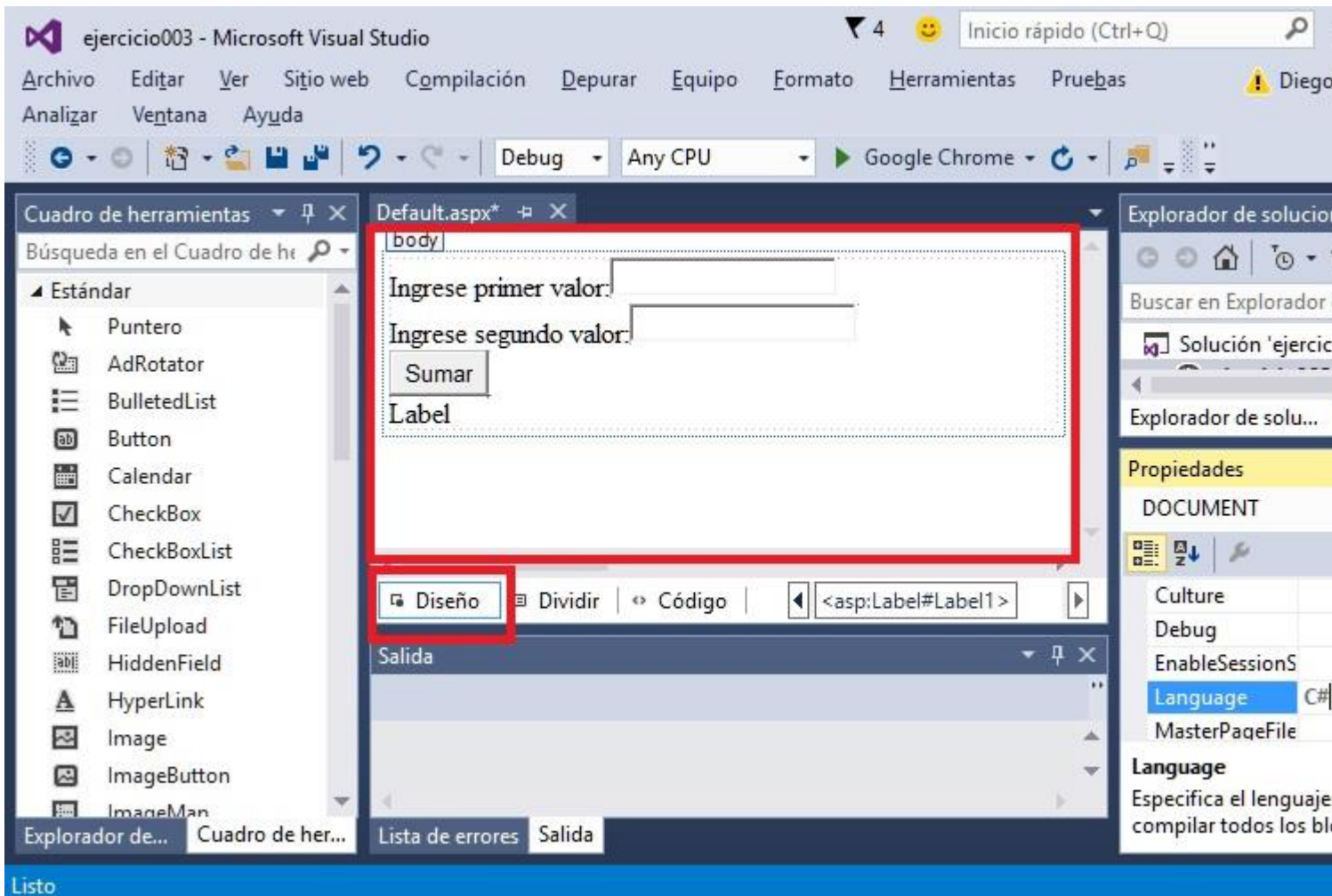


Controles Label, Button y TextBox.

Hasta ahora hemos utilizado los controles de tipo Label y Button, ahora utilizaremos el control TextBox. Crearemos una aplicación que nos permita ingresar dos números y luego en una Label muestre la suma de los mismos.

(Si tenemos un sitio web abierto podemos cerrarlo seleccionando la opción: Archivo->Cerrar solución y luego seguir los mismos pasos que vimos anteriormente para crear un nuevo sitio web ASP.NET)

Crearemos un sitio web llamado ejercicio003 y desarrollaremos la siguiente interfaz de usuario:



Los textos "Ingrese primer valor:" e "Ingrese segundo valor:" los escribimos directamente en cambio el Button, los dos TextBox y la Label las arrastramos desde el Cuadro de herramientas. Inicializamos la propiedad Text del objeto de tipo Button con la etiqueta "Sumar".

Luego codificamos el evento click del objeto de la clase Button (en este evento debemos extraer el contenido de los dos controles de tipo TextBox y proceder a convertirlos a tipo de dato entero y sumarlos):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

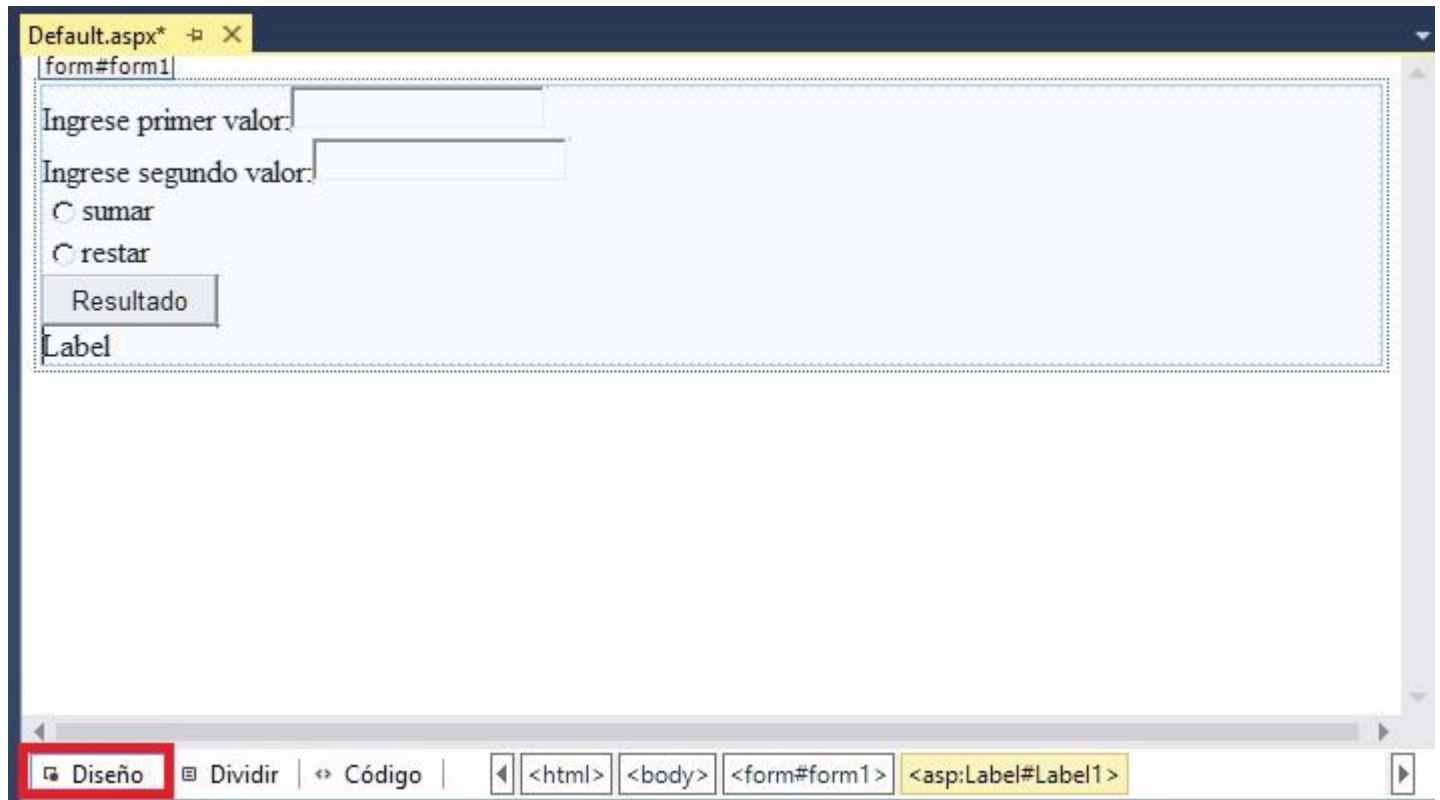
    protected void Button1_Click(object sender, EventArgs e)
    {
        int x1 = int.Parse(TextBox1.Text);
        int x2 = int.Parse(TextBox2.Text);
        int suma = x1 + x2;
        Label1.Text = "La suma de los dos valores es:" + suma;
    }
}
```

La clase int tiene un método estático llamado Parse y que tiene por objetivo recibir un String y retornar el valor del mismo convertido a entero.

Luego de sumar mostramos en la Label el resultado de la suma de los dos valores ingresados (al utilizar el operador + el string y el entero tenemos que se convierte suma automáticamente a string)

Control RadioButton.

Para probar el funcionamiento del control RadioButton crearemos un nuevo sitio web llamado ejercicio004. Crearemos una interfaz de usuario similar al problema anterior, con la salvedad que le agregaremos dos controles de tipo RadioButton para poder indicar si queremos sumar o restar los valores ingresados:



Como vemos agregamos dos controles de tipo RadioButton, inicializamos las propiedades Text con los textos ?Sumar? y ?Restar?. Luego para indicar que los controles RadioButton están en el mismo grupo debemos inicializar la propiedad GroupName con el mismo valor (con esto logramos que al seleccionar un RadioButton el otro se desmarca), si nos olvidamos inicializar la propiedad GroupName luego los dos controles de tipo RadioButton se podrán seleccionar en forma simultánea (podemos asignar la propiedad GroupName de ambos controles con el valor "operacion").

Si queremos que alguno de los dos RadioButton aparezca seleccionado por defecto debemos inicializar la propiedad Checked con el valor true.

La codificación del evento click del objeto Button1 es el siguiente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
```

```

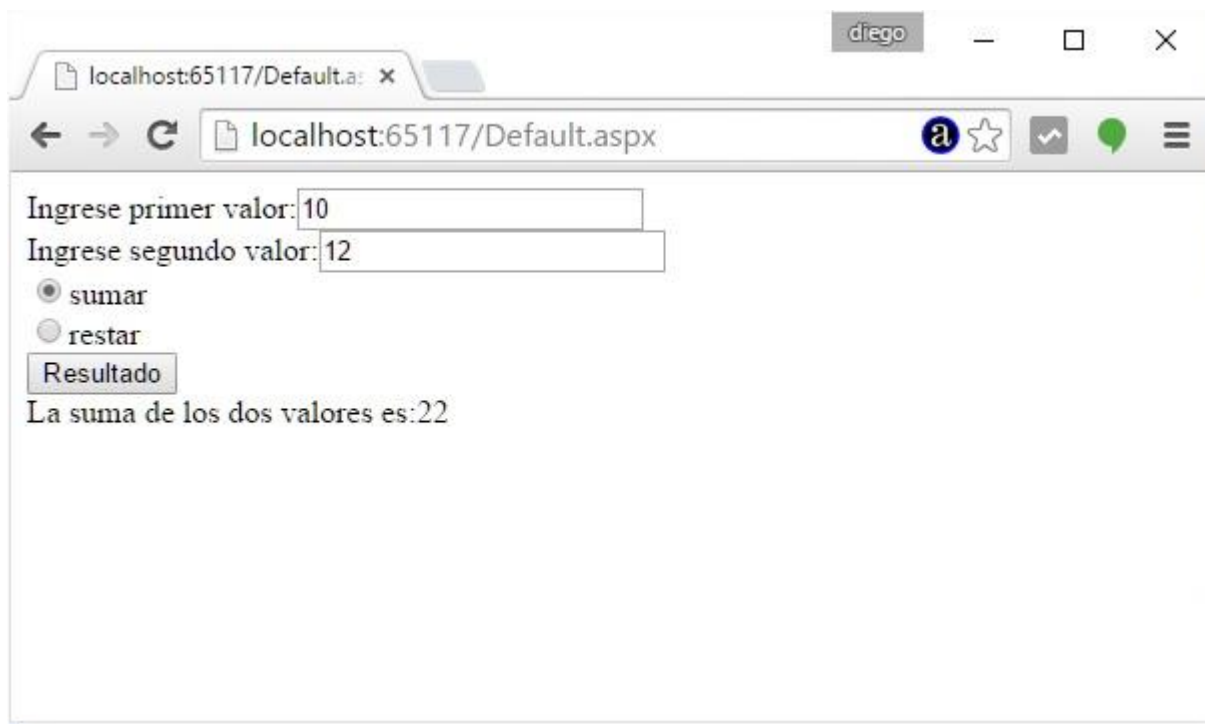
protected void Page_Load(object sender, EventArgs e)
{

}

protected void Button1_Click(object sender, EventArgs e)
{
    int x1 = int.Parse(TextBox1.Text);
    int x2 = int.Parse(TextBox2.Text);
    int resultado = x1 + x2;
    if (RadioButton1.Checked)
    {
        resultado = x1 + x2;
        Label1.Text = "La suma de los dos valores es:" + resultado;
    }
    else
    {
        if (RadioButton2.Checked)
        {
            resultado = x1 - x2;
            Label1.Text = "La diferencia de los dos valores es:" + resultado;
        }
    }
}
}

```

Cuando se presiona el botón se ejecuta el método Button1_Click donde primero extraemos el contenido de los dos controles TextBox y los convertimos a entero. Verificamos con if cual de los dos controles RadioButton se encuentra seleccionado. La propiedad Checked del RadioButton indica si está seleccionado el control o no.

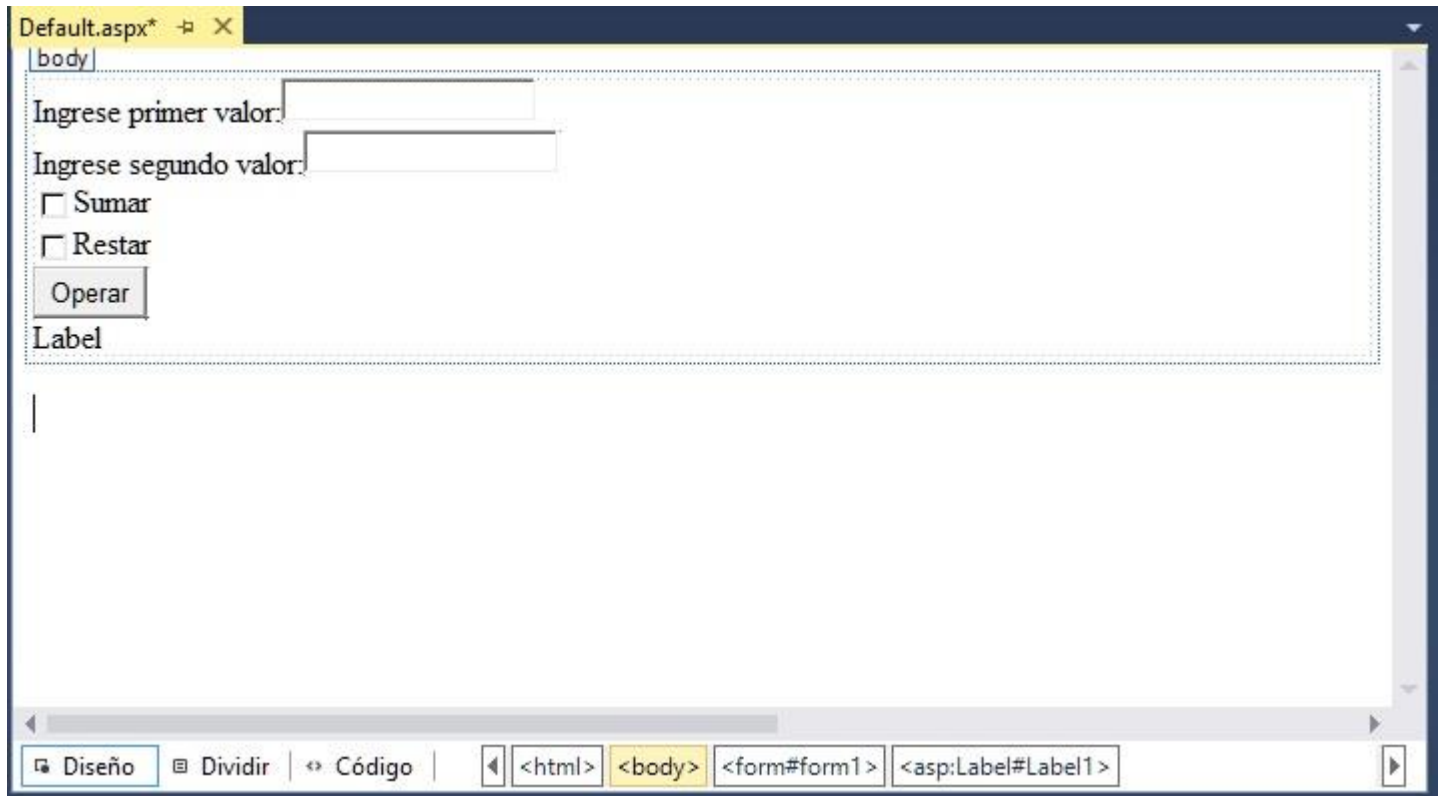


Control CheckBox.

Los controles CheckBox permiten que más de uno sea seleccionado. Similar a los controles RadioButton tiene dos estados (seleccionado o no seleccionado) y esto lo sabemos según el estado de la propiedad Checked.

Codificaremos un nuevo sitio web llamado ejercicio005 que permita cargar dos valores y luego calcule la suma y/o resta de los valores ingresados. Como podemos seleccionar ambas operaciones utilizaremos los

controles de tipo CheckBox.
La interfaz visual es la siguiente:



La codificación del evento click del botón es:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

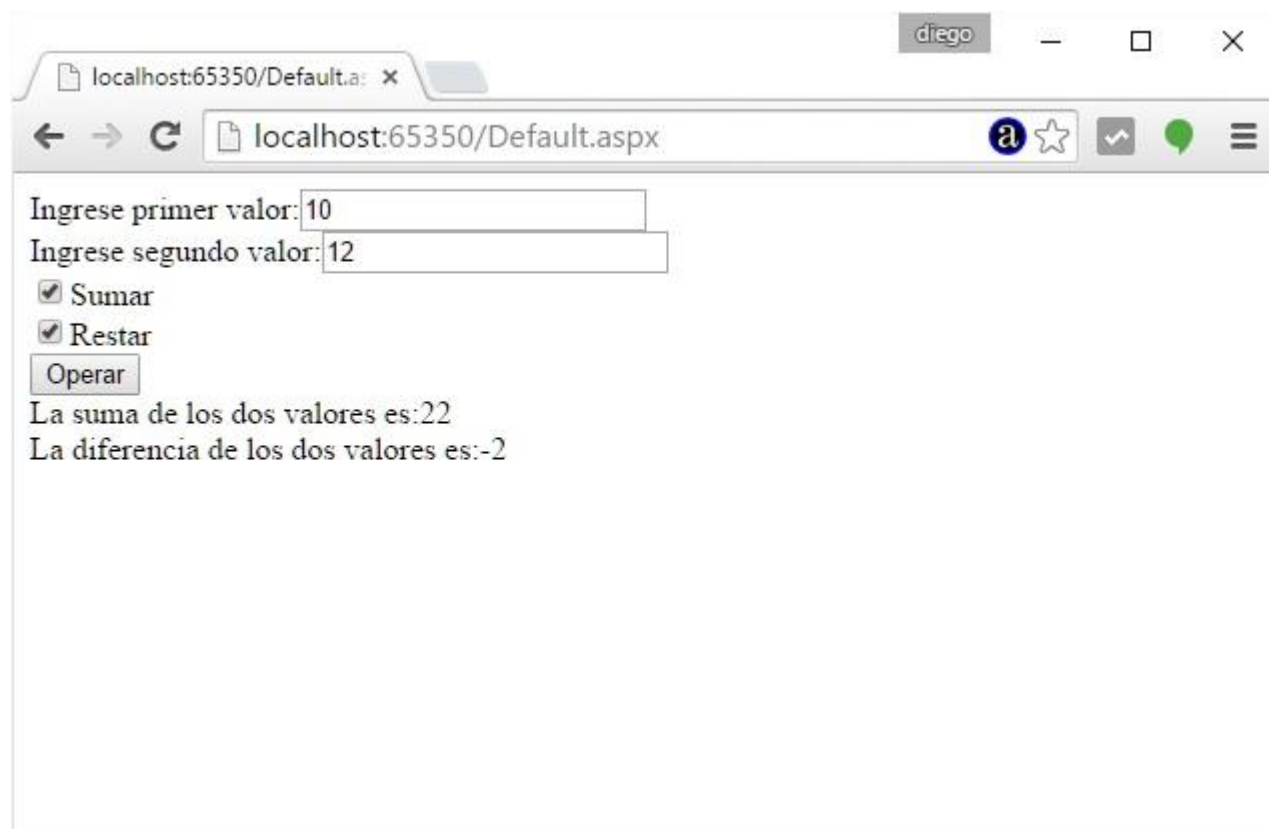
    protected void Button1_Click(object sender, EventArgs e)
    {
        int x1 = int.Parse(TextBox1.Text);
        int x2 = int.Parse(TextBox2.Text);
        Label1.Text = "";
        int resultado;
        if (CheckBox1.Checked)
        {
            resultado = x1 + x2;
            this.Label1.Text = "La suma de los dos valores es:" + resultado + "<br>";
        }
        if (CheckBox2.Checked)
        {
            resultado = x1 - x2;
            Label1.Text = Label1.Text + "La diferencia de los dos valores es:" +
resultado;
        }
    }
}
```



```
}  
}
```

Disponemos dos if a la misma altura ya que ambos CheckBox pueden estar seleccionados. Previo a los if borramos el contenido de la Label en el caso que tenga el resultado de operaciones anteriores. Luego en el primer if verificamos si el primer CheckBox está seleccionado y procedemos a inicializar la propiedad Text de la Label con el resultado de la suma de los dos valores ingresados, seguidamente verificamos con un segundo if si el siguiente CheckBox está seleccionado, en caso afirmativo agregamos al contenido actual de la Label el resultado de la diferencia de los valores ingresados (Como vemos podemos añadir elementos HTML a la propiedad Text de una Label, luego estas serán interpretadas por el navegador)

En el navegador tenemos el siguiente resultado:



Control ListBox.

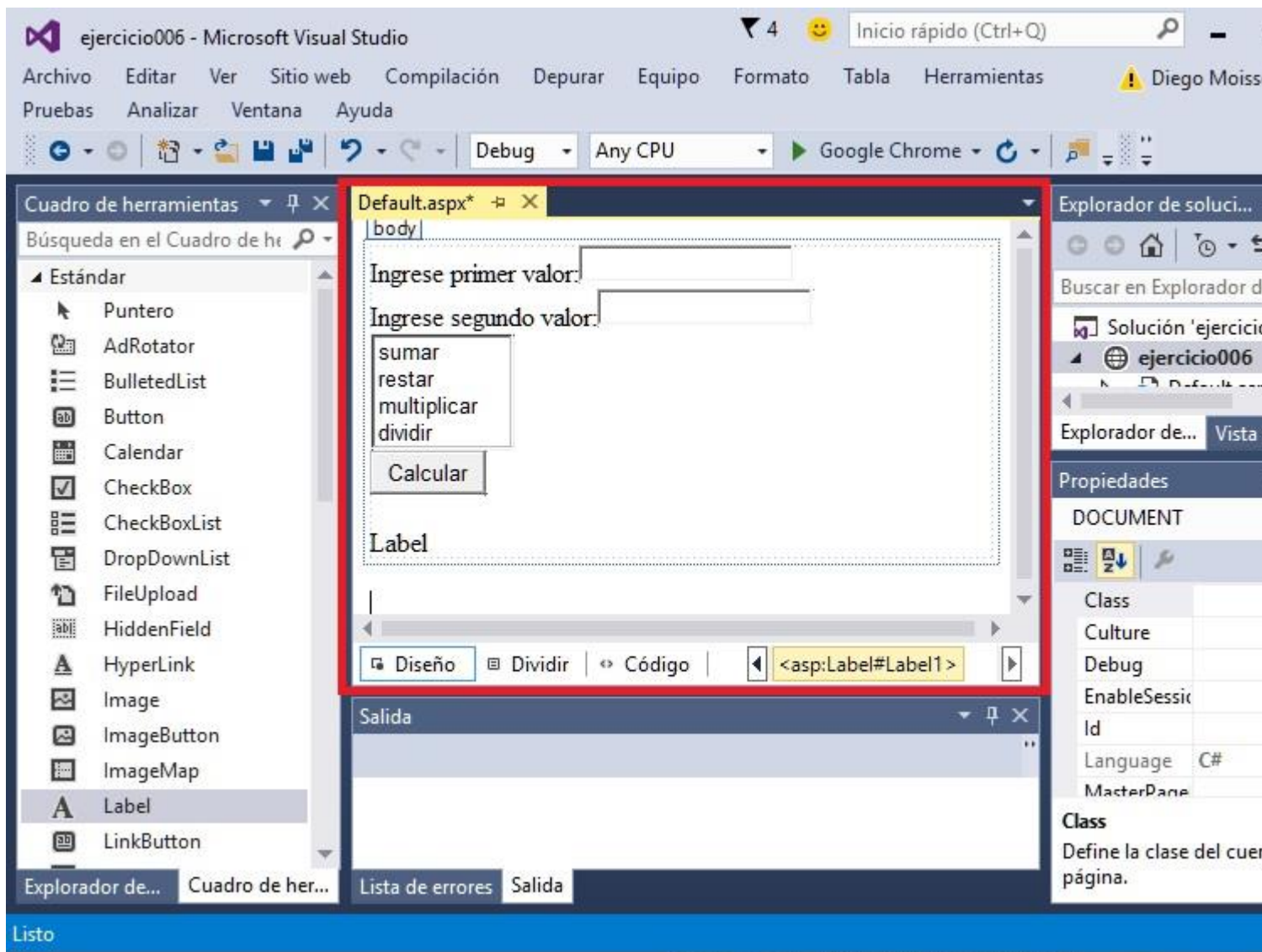
El control ListBox permite crear una lista de valores.

La propiedad Item permite definir los miembros de la lista (cada item define las propiedades Text (valor a mostrar), Value (valor a retornar en caso de estar seleccionado), Selected (con un valor lógico))

Otra propiedad muy importante del control ListBox es SelectionMode, esta admite dos valores: Single o Multiple.

Crearemos un sitio web (ejercicio006) que permita cargar dos valores y mediante un control ListBox poder seleccionar si queremos sumar, restar, multiplicar o dividir dichos valores (como podemos seleccionar varias operaciones en forma simultánea configuraremos la propiedad SelectionMode del ListBox con el valor Multiple) cargamos en la propiedad Items las cuatro operaciones posibles.

Luego la interfaz visual a crear es la siguiente (insertamos también una Label luego del botón ?Calcular?, con el objetivo de mostrar los resultados):



Cuando se presiona el botón calcular verificamos cual de las opciones está seleccionada y procedemos a calcular y mostrar los resultados.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        int x1 = int.Parse(TextBox1.Text);
        int x2 = int.Parse(TextBox2.Text);
        Label1.Text = "";
        if (ListBox1.Items[0].Selected)
        {
            int suma = x1 + x2;
            Label1.Text = "La suma es:" + suma + "<br>";
        }
    }
}
```

```

    }
    if (ListBox1.Items[1].Selected)
    {
        int resta = x1 - x2;
        Label1.Text += "La diferencia:" + resta + "<br>";
    }
    if (ListBox1.Items[2].Selected)
    {
        int multi = x1 * x2;
        Label1.Text += "El producto:" + multi + "<br>";
    }
    if (ListBox1.Items[3].Selected)
    {
        int divi = x1 / x2;
        Label1.Text += "La division:" + divi + "<br>";
    }
}
}

```

Como podemos ver primero vaciamos el contenido de la Label1 y procedemos mediante cuatro if a verificar cuales de los elementos del ListBox se encuentran seleccionados:

```
if (ListBox1.Items[0].Selected)
```

Si por ejemplo el primer elemento del ListBox se encuentra seleccionado procedemos a sumar los dos valores almacenados en los TextBox y los agregamos a la Label:

```

    int suma = x1 + x2;
    Label1.Text = "La suma es:" + suma + "<br>";

```

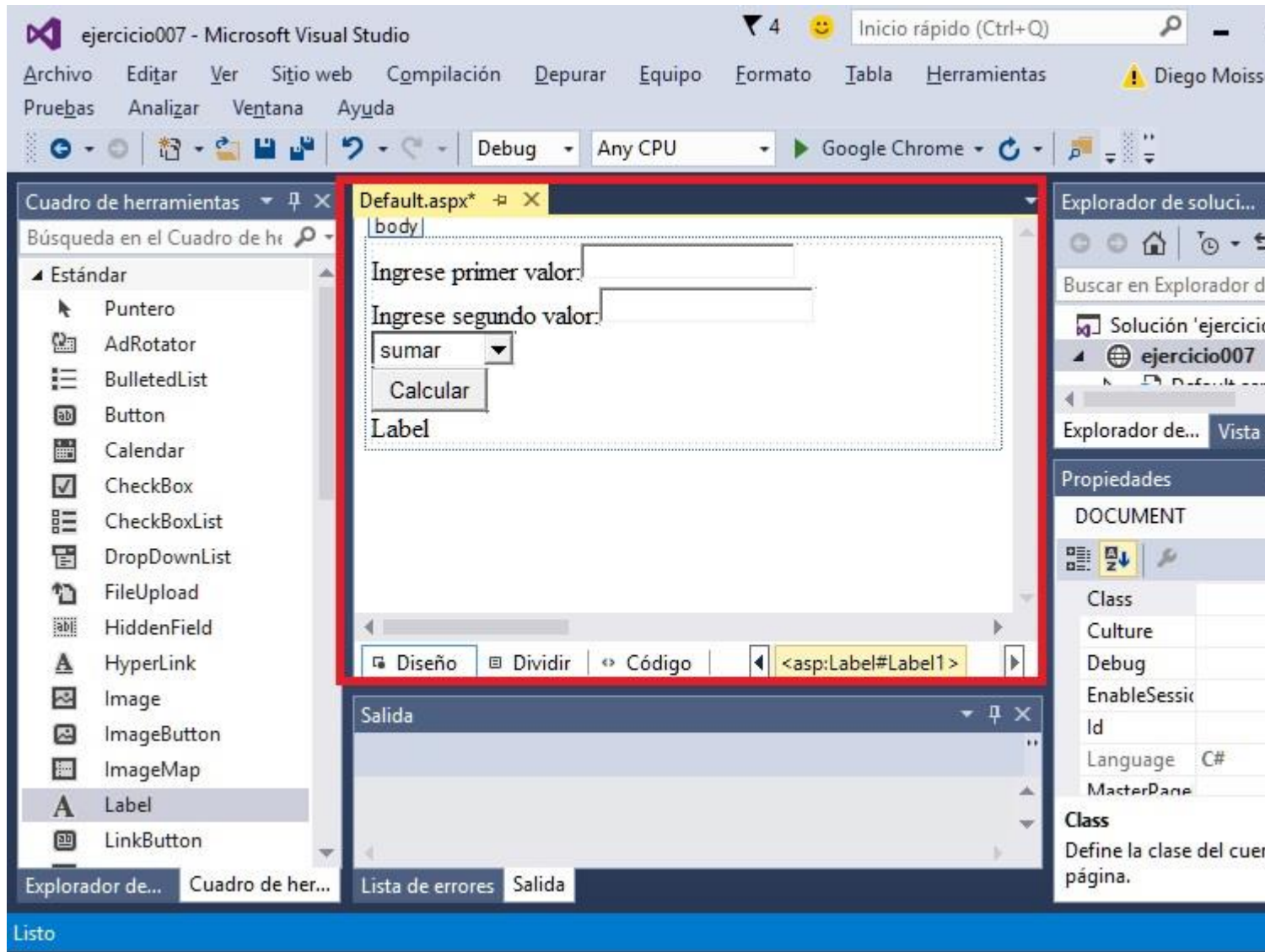
En el navegador tenemos como resultado:



Control DropDownList.

El control DropDownList permite crear una lista de valores y luego seleccionar solo uno de ellos, esta es la diferencia fundamental con el control ListBox.

Para probar el DropDownList implementaremos el problema propuesto con el control ListBox, ahora la interfaz es la siguiente (crear un sitio web llamado ejercicio007):



Cargamos las cuatro operaciones básicas en el control DropDownList y para el evento clic del botón tenemos que codificar:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        int x1 = int.Parse(TextBox1.Text);
        int x2 = int.Parse(TextBox2.Text);
        Label1.Text = "";
    }
}
```

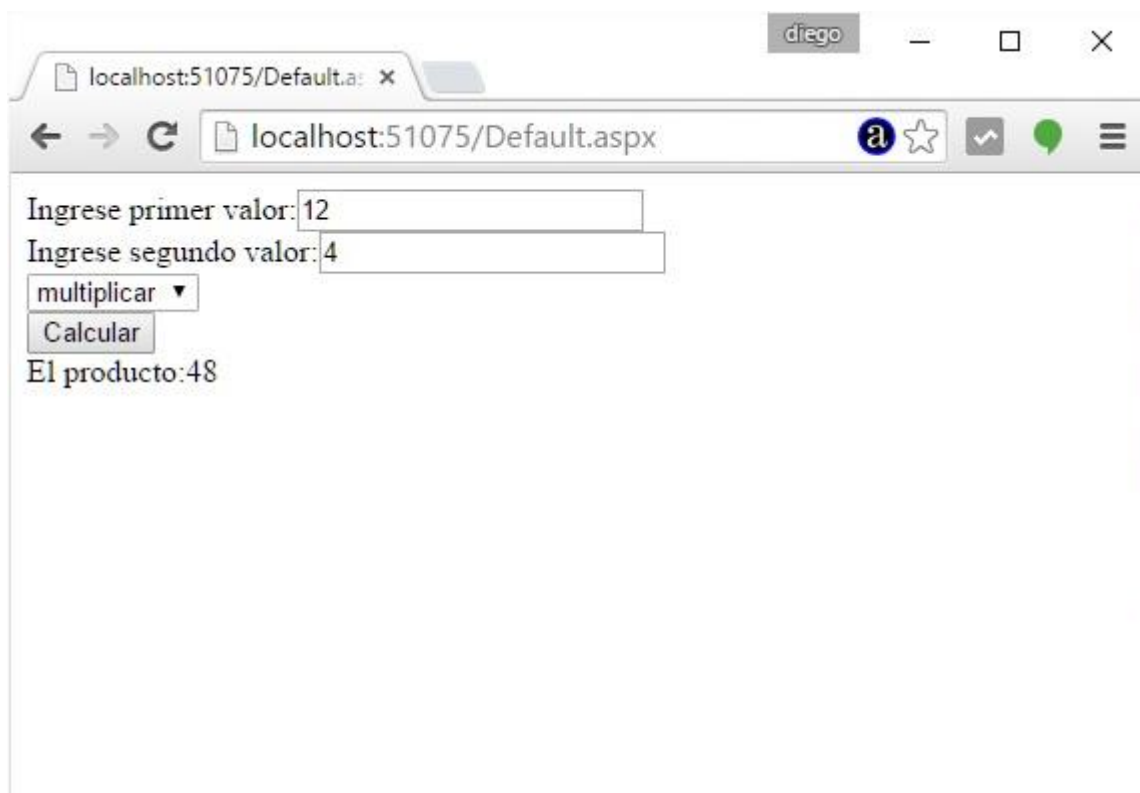
```

if (DropDownList1.Items[0].Selected)
{
    int suma = x1 + x2;
    Label1.Text = "La suma es:" + suma + "<br>";
}
else
    if (DropDownList1.Items[1].Selected)
    {
        int resta = x1 - x2;
        Label1.Text = "La diferencia:" + resta + "<br>";
    }
    else
        if (DropDownList1.Items[2].Selected)
        {
            int multi = x1 * x2;
            Label1.Text = "El producto:" + multi + "<br>";
        }
        else
            if (DropDownList1.Items[3].Selected)
            {
                int divi = x1 / x2;
                Label1.Text = "La division:" + divi + "<br>";
            }
}
}

```

Como solo un elemento del control DropDownList puede estar seleccionado disponemos una serie de if/else if para verificar cual de ellos es el seleccionado. Cuando identificamos el item seleccionado procedemos a efectuar el cálculo correspondiente y mostrarlo en la Label.

En el navegador tenemos como resultado:



Ejercicios propuestos

1 ? Confeccionar una página que solicite el ingreso del nombre y apellido de una persona (cada elemento en un TextBox), luego al presionar un botón mostrar en una label si alguno de los datos no se cargó.

2 ? Confeccionar una página que muestre un examen múltiple choice (disponer 4 preguntas y tres respuestas posibles por pregunta) utilizar controles de tipo RadioButton para la selección de la respuesta correcta. Mostrar la cantidad de respuestas correctas luego que se presiona un botón.

3 ? Disponer un conjunto de RadioButton agrupados. Mostrar en las leyendas de cada RadioButton distintos buscadores (Google, Bing, Yahoo etc.)

Cuando se presione un botón redireccionar a dicho servidor (para redireccionar debemos utilizar la siguiente sintaxis

```
Response.Redirect("http://www.google.com.ar");
```

4 ? Confeccionar el siguiente formulario para registrarse en un sitio web (utilizar controles de la pestaña estandar)

En una Label mostrar los datos cargados en cada control (disponer la Label al final del formulario) Hacer por lo menos 5 validaciones y mostrar mensajes de errores en una Label.

Nombre de usuario:	<input type="text"/>
Clave:	<input type="password"/>
Repita la clave:	<input type="password"/>
Correo electrónico:	<input type="text"/>
Apellido:	<input type="text"/>
Nombre:	<input type="text"/>
País de origen:	<input type="text" value="Argentina"/>
Provincia:	<input type="text"/>
Código Postal:	<input type="text"/>
Sexo:	<input type="radio"/> Hombre <input type="radio"/> Mujer
Fecha de nacimiento (dd/mm/aaaa)	<input type="text"/>
<div>Comentarios</div> <div><input type="text"/></div>	
Acepto los términos y condiciones.	<input type="checkbox"/> Acepto
	<input type="button" value="Aceptar"/>