

Creación, lectura y escritura de archivos de texto.

Conceptos.

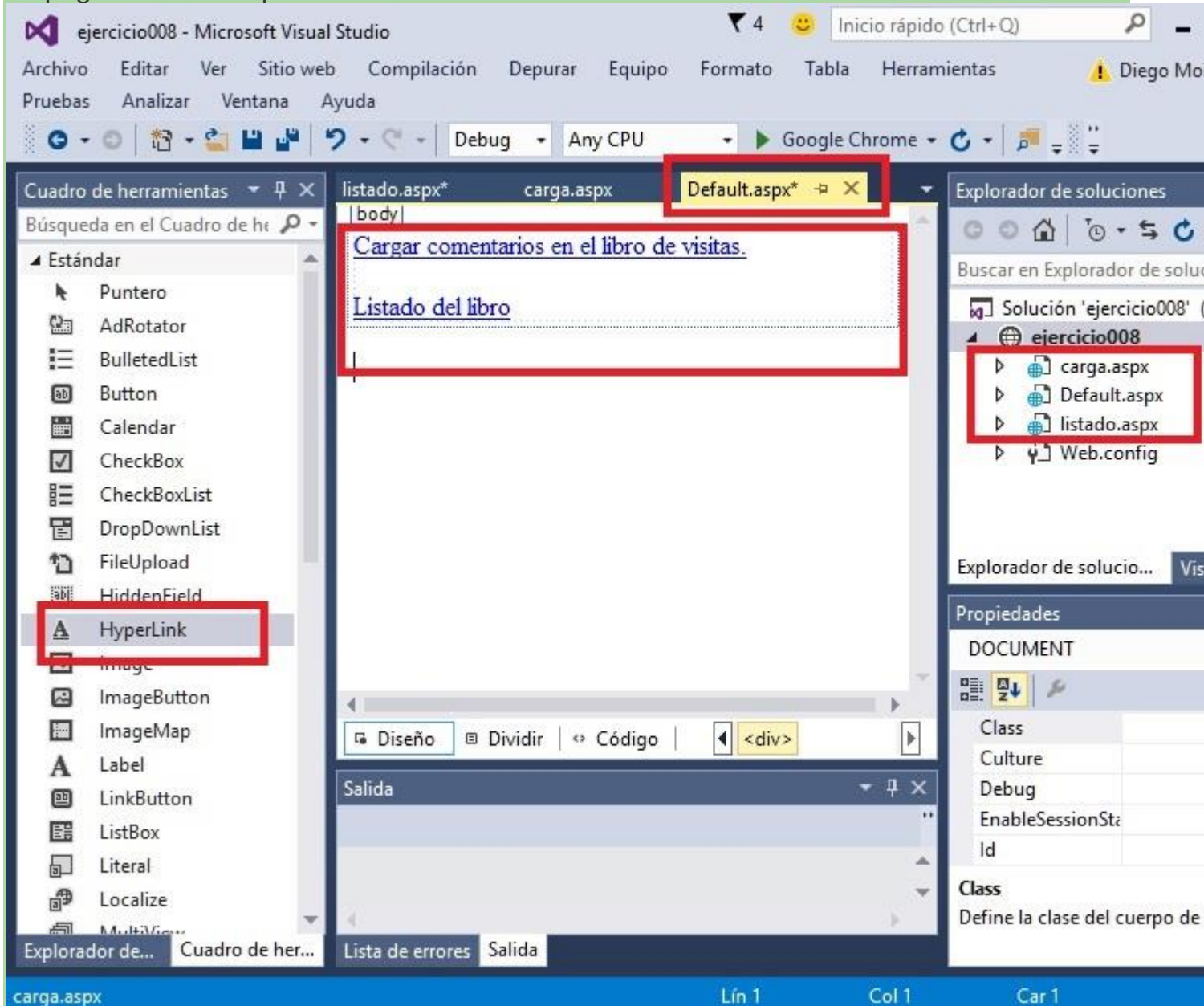
En muchas situaciones es necesario almacenar información en el servidor, tenemos dos alternativas, si la cantidad de datos a almacenar es pequeña podemos utilizar un archivo de texto para ello (en la próxima clase veremos como almacenar en una base de datos) Un archivo de texto no requiere grandes recursos del servidor.

Creación y carga del archivo de texto.

Confeccionaremos un libro de visitas de un sitio web.

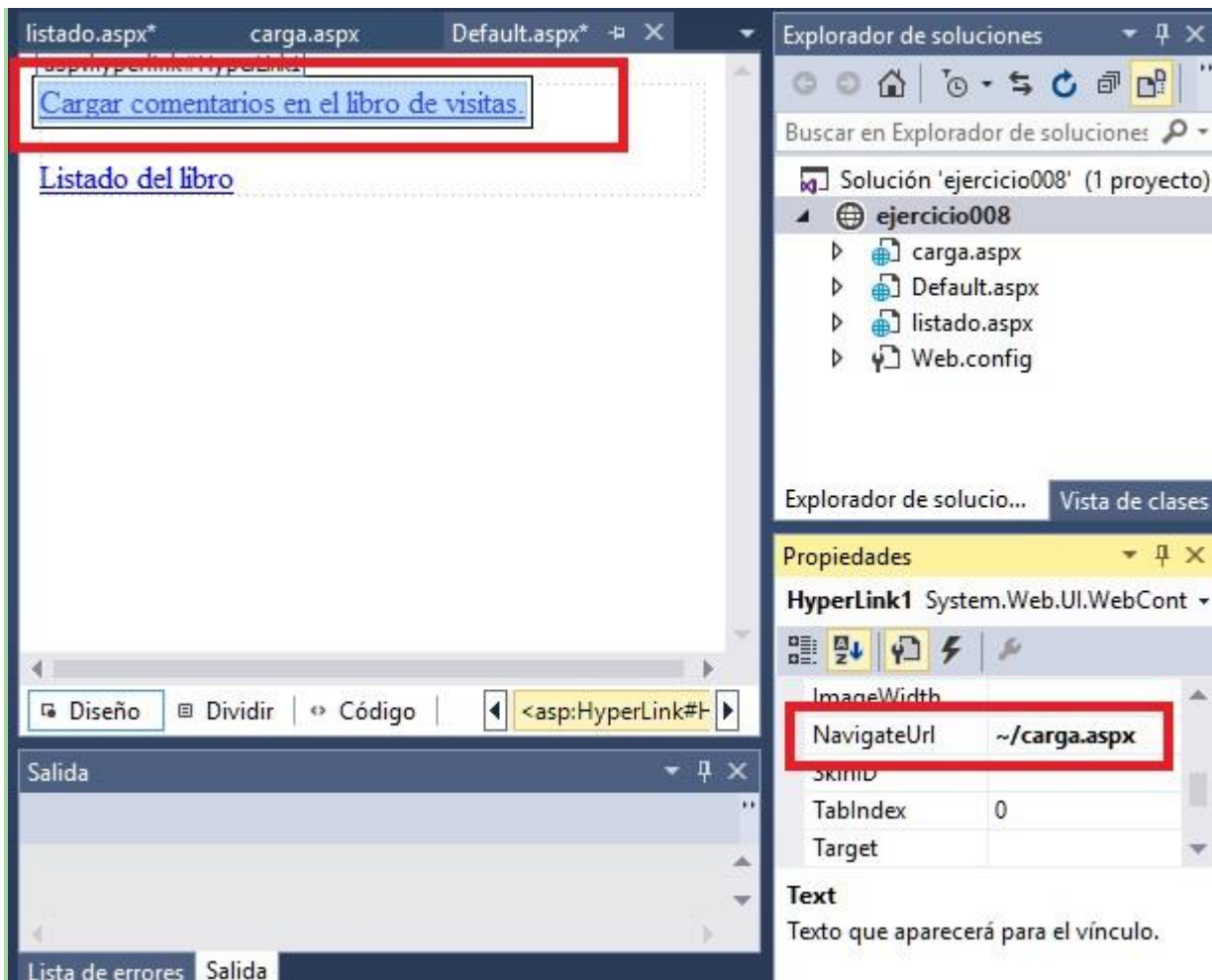
Crear un sitio web llamado ejercicio008 y luego agregar tres web form (Default.aspx, carga.aspx y listado.aspx), en la página principal (Default.aspx) dispondremos de dos hipervínculos (HyperLink) que sacamos del Cuadro de herramientas, el primero irá al formulario de carga y el segundo al listado del contenido del archivo.

La página Default.aspx:



Para los controles HyperLink debemos inicializar las propiedades Text con el mensaje a mostrar en el hipervínculo y mediante la propiedad NavigateUrl indicamos a que página se llama al ser presionado:

Creación, lectura y escritura de archivos de texto.



El primer control HyperLink nos lleva al formulario web que permite cargar el nombre del visitante, su país y los comentarios.

El formulario web requiere tres objetos de la clase TextBox, al tercero donde se ingresan los comentarios debemos inicializar la propiedad TextMode con el valor MultiLine. Disponemos un control de tipo HyperLink para poder retornar a la página principal (Default.aspx)

Creación, lectura y escritura de archivos de texto.

The screenshot shows a web browser window with three tabs: 'listado.aspx', 'carga.aspx' (which is active), and 'Default.aspx'. The 'carga.aspx' page contains a form with the following elements:

- A label 'Nombre:' followed by a text input field.
- A label 'Pais:' followed by a text input field.
- A label 'Comentarios' followed by a large text area with a vertical scrollbar.
- A 'Confirmar' button.
- A 'Label' control, likely for displaying a message.
- A blue link labeled 'Retornar'.

At the bottom of the browser window, the 'Diseño' (Design) view is selected. The breadcrumb trail shows the hierarchy: '<form#form1>' > '<div>' > '<asp:TextBox#TextBox3>'. The 'asp:TextBox#TextBox3' element is currently selected.

Cuando se presiona el botón confirmar procedemos a almacenar los datos del formulario en el archivo de texto, si existe los agregamos al final, en caso que no exista se crea el archivo.

Mostramos finalmente en una label que se almacenaron los datos.

El código completo para registrar los datos es:

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

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

    }

    protected void Button1_Click(object sender,
    EventArgs e)
    {
```

Creación, lectura y escritura de archivos de texto.

```
        StreamWriter arch = new
StreamWriter(Server.MapPath(".") + "/visitas.txt",
true);
        arch.WriteLine("Nombre:" + TextBox1.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("Pais:" + TextBox2.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("Comentarios<br>");
        arch.WriteLine(TextBox3.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("<hr>");
        arch.Close();
        Label1.Text = "Datos Registrados";
    }
}
```

Primero importamos el espacio de nombres donde está declarada la clase StreamWriter:
`using System.IO;`

Cuando se presiona el botón confirmar creamos un objeto de la clase StreamWriter y le pasamos en el constructor el path y nombre del archivo a abrir o crear según sea el caso. Para obtener el directorio actual utilizamos el método MapPath del objeto Server:

```
        StreamWriter arch = new StreamWriter(Server.MapPath(".") + "/visitas.txt",
true);
```

El valor true que le pasamos en el constructor significa que si el archivo no existe en el servidor en la carpeta especificada se procede a su creación y en el caso que si existe se procede a su apertura y posicionado del puntero de archivo al final del mismo.

El método WriteLine de la clase StreamWriter permite almacenar una línea en el archivo de texto y el posterior ingreso en forma automática del salto de línea:

```
        arch.WriteLine("Nombre:" + TextBox1.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("Pais:" + TextBox2.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("Comentarios<br>");
        arch.WriteLine(TextBox3.Text);
        arch.WriteLine("<br>");
        arch.WriteLine("<hr>");
```

Cuando finalizamos de trabajar con el archivo procedemos a cerrarlo:

```
        arch.Close();
```

Lectura del archivo de texto.

Creamos una página donde mostraremos todos los datos almacenados en el archivo de texto visitas.txt.

El archivo de texto almacena los datos de los visitantes y las elementos HTML básicas para hacer los saltos de línea y líneas separadoras entre comentarios, solo nos queda leer el archivo e ir almacenándolo en la Label para que lo muestre.

Podemos disponer un control de tipo HyperLink para retornar a la página principal.

Creación, lectura y escritura de archivos de texto.



El código necesario para leer y cargar los datos en la Label1 es:

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

public partial class listado : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        StreamReader arch = new
StreamReader(Server.MapPath(".") + "/visitas.txt");
        Label1.Text = arch.ReadToEnd();
        arch.Close();
    }
}
```

Importamos el espacio de nombres donde está declarada la clase StreamReader:

```
using System.IO;
```

El algoritmo de lectura lo implementamos en el método Page_Load que se ejecuta primero al cargarse la página.

Procedemos a realizar la apertura del archivo, indicando el camino donde se encuentra:

```
StreamReader arch = new StreamReader(Server.MapPath(".") + "/visitas.txt");
```

El método ReadToEnd() de la clase StreamReader retorna el contenido de todo el archivo de texto:

```
Label1.Text = arch.ReadToEnd();
```

Finalmente procedemos a cerrar el archivo:

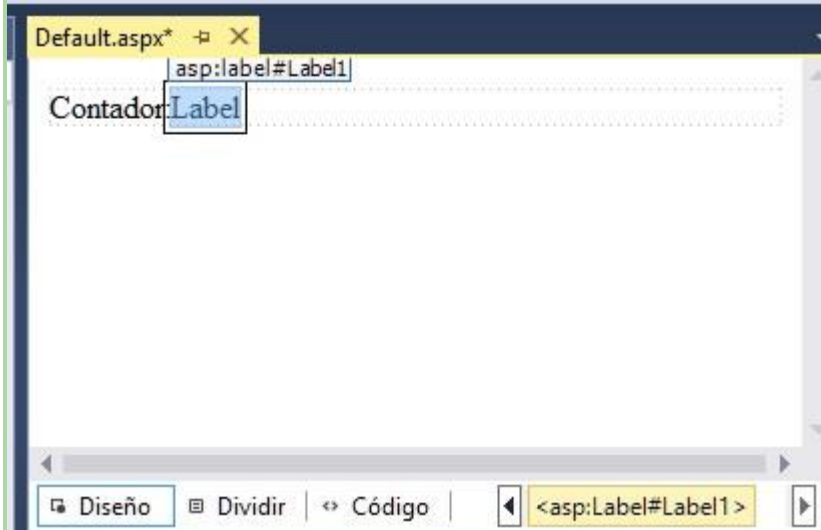
```
arch.Close();
```

Creación, lectura y escritura de archivos de texto.

Contador de páginas vistas.

Confeccionaremos ahora un simple contador de páginas utilizando un archivo de texto de una sola línea. Cada vez que un navegador solicite la página mostraremos el contador. Crear un nuevo sitio web llamado ejercicio009 y crear un único web form llamado Default.aspx.

Escribir el texto "Contador:" directamente en la página y a su lado disponer una Label:



En la Label mostraremos el valor del contador que se almacena en el archivo de texto. En el evento Page_Load hacemos todo el algoritmo:

```
using System;
using System.Collections.Generic;
using System.IO;
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)
    {
        if (File.Exists(Server.MapPath(".") +
"/contador.txt"))
        {
            StreamReader arch1 = new
StreamReader(Server.MapPath(".") + "/contador.txt");
            string valor = arch1.ReadToEnd();
            int contador = int.Parse(valor);
            contador++;
            arch1.Close();
            StreamWriter arch2 = new
StreamWriter(Server.MapPath(".") + "/contador.txt");
            arch2.WriteLine(contador.ToString());
            arch2.Close();
            Label1.Text = contador.ToString();
        }
    }
}
```


Creación, lectura y escritura de archivos de texto.

```
    }
    else
    {
        StreamWriter arch = new
StreamWriter(Server.MapPath(".") + "/contador.txt");
        arch.WriteLine("1");
        arch.Close();
        Label1.Text = "1";
    }
}
```

Mediante un if verificamos si existe el archivo que almacena el contador (la clase File tiene un método estático llamado Exists que retorna true si existe el archivo en la carpeta indicada):

```
if (File.Exists(Server.MapPath(".") + "/contador.txt"))
```

En caso que el archivo existe procedemos a abrirlo con el objetivo de leer su contenido:

```
StreamReader arch1 = new StreamReader(this.Server.MapPath(".") +
"/contador.txt");
string valor = arch1.ReadToEnd();
int contador = int.Parse(valor);
contador++;
arch1.Close();
```

Luego procedemos a crearlo nuevamente y almacenar el valor que acabamos de leer, previo a incrementarlo en uno (mostramos en la Label el valor del contador actual):

```
StreamWriter arch2 = new StreamWriter(Server.MapPath(".") +
"/contador.txt");
arch2.WriteLine(contador.ToString());
arch2.Close();
Label1.Text = contador.ToString();
```

Si el archivo no existe procedemos a crearlo almacenando el valor 1:

```
StreamWriter arch = new StreamWriter(Server.MapPath(".") +
"/contador.txt");
arch.WriteLine("1");
arch.Close();
this.Label1.Text = "1";
```

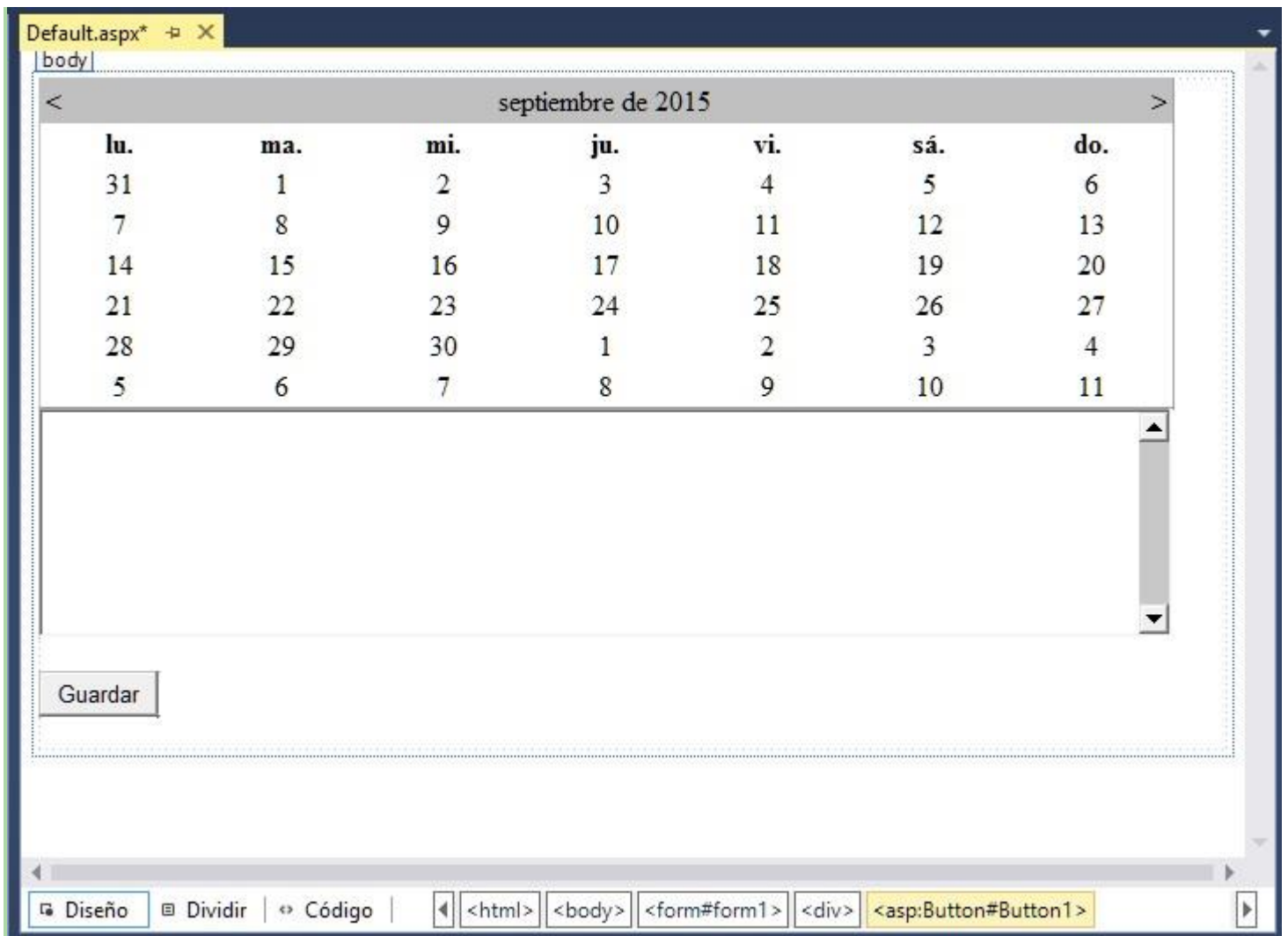
Agenda de actividades.

Implementaremos un tercer problema utilizando como estructura de datos un archivo de texto.

Crearemos un sitio web llamado ejercicio010 (sitio web ASP.Net vacío) y agregaremos un Web Form (Default.aspx) donde implementaremos una agenda.

Crear una interfaz visual similar a esta para administrar la agenda de actividades:

Creación, lectura y escritura de archivos de texto.



Disponemos del "Cuadro de herramientas" un objeto de la clase "Calendar", uno de tipo TextBox e iniciamos su propiedad TextMode con el valor MultiLine y finalmente un Button. Si queremos modificar la piel del objeto "Calendar" presionamos el botón derecho del mouse sobre el objeto y elegimos la opción "Formato Automático..." Debemos seleccionar el objeto Calendar y definir el evento "Selection_Changed" que se dispara cuando el operador selecciona una fecha:

Creación, lectura y escritura de archivos de texto.

The screenshot shows the Visual Studio IDE. The main window displays a calendar control titled "septiembre de 2015". The calendar is a table with days of the week as headers and dates as rows. A red rectangle highlights the calendar control. Below the calendar is a "Guardar" button. The right sidebar shows the "Explorador de soluciones" (Solution Explorer) with a project named "ejercicio010" containing "Default.aspx" and "Web.config". Below that is the "Propiedades" (Properties) window for "Calendar1". The "SelectionChanged" property is highlighted with a red rectangle. The bottom status bar shows the current view is "Diseño" (Design).

L	M	X	J	V	S	D
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

El código fuente para el evento "Selection_Changed", "Page_Load" y "Button1_Click" es:

```
using System;
using System.Collections.Generic;
using System.IO;
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)
    {
        string fecha =
            Calendar1.TodaysDate.Day.ToString() +
            Calendar1.TodaysDate.Month.ToString() +
            Calendar1.TodaysDate.Year.ToString();
        if (File.Exists(Server.MapPath(".") + "/" +
            fecha))
        {

```

Creación, lectura y escritura de archivos de texto.

```
        StreamReader arch = new
StreamReader(Server.MapPath(".") + "/" + fecha);
        TextBox1.Text = arch.ReadToEnd();
        arch.Close();
    }

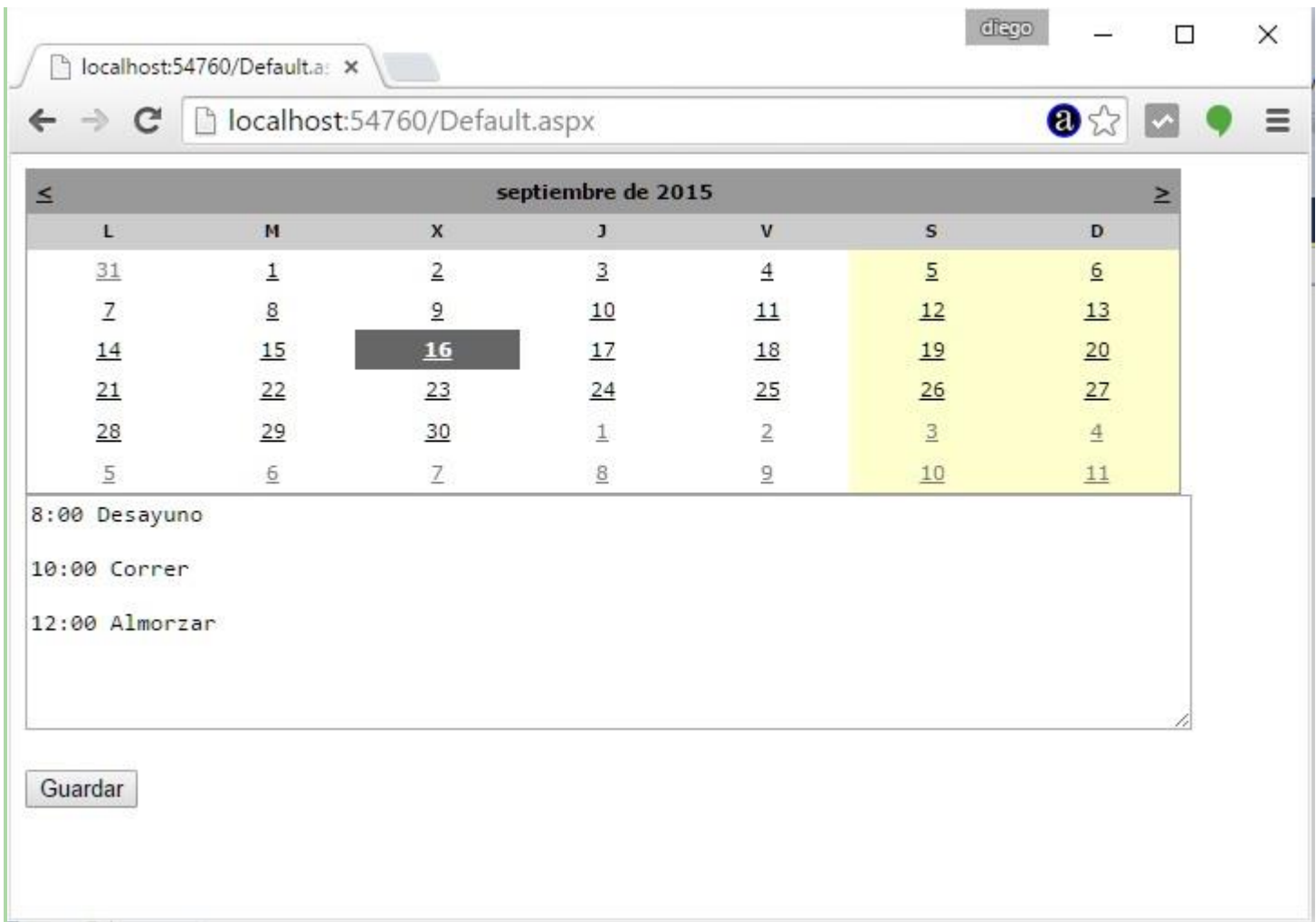
}

protected void Button1_Click(object sender,
EventArgs e)
{
    string fecha =
Calendar1.SelectedDate.Day.ToString() +
Calendar1.SelectedDate.Month.ToString() +
Calendar1.SelectedDate.Year.ToString();
    StreamWriter arch = new
StreamWriter(Server.MapPath(".") + "/" + fecha, true);
    arch.WriteLine(TextBox1.Text);
    arch.Close();
}

protected void Calendar1_SelectionChanged(object
sender, EventArgs e)
{
    string fecha =
Calendar1.SelectedDate.Day.ToString() +
        Calendar1.SelectedDate.Month.ToString() +
        Calendar1.SelectedDate.Year.ToString();
    if (File.Exists(Server.MapPath(".") + "/" +
fecha))
    {
        StreamReader arch = new
StreamReader(Server.MapPath(".") + "/" + fecha);
        TextBox1.Text = arch.ReadToEnd();
        arch.Close();
    }
    else
        TextBox1.Text = "";
}
}
```

Si lo ejecutamos podemos cargar actividades para el día seleccionado:

Creación, lectura y escritura de archivos de texto.



localhost:54760/Default.aspx

septiembre de 2015

L	M	X	J	V	S	D
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

8:00 Desayuno
10:00 Correr
12:00 Almorzar

Guardar

Cuando se presiona el botón "Guardar" se ejecuta:

```
protected void Button1_Click(object sender, EventArgs e)
{
    string fecha = Calendar1.SelectedDate.Day.ToString() +
                  Calendar1.SelectedDate.Month.ToString() +
                  Calendar1.SelectedDate.Year.ToString();
    StreamWriter arch = new StreamWriter(Server.MapPath(".") + "/" + fecha, true);
    arch.WriteLine(TextBox1.Text);
    arch.Close();
}
```

Primero generamos una fecha con la fecha seleccionada en el objeto "Calendar". Creamos un archivo con nombre igual a la fecha seleccionada y finalmente grabamos todo el contenido del objeto TextBox.

Para el evento "SelectionChanged" del control "Calendar" recuperamos la fecha seleccionada y verificamos si existe un archivo con dicha fecha, en caso afirmativo procedemos a leer el archivo y mostrarlo en el TextBox:

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    string fecha = Calendar1.SelectedDate.Day.ToString() +
                  Calendar1.SelectedDate.Month.ToString() +
                  Calendar1.SelectedDate.Year.ToString();
    if (File.Exists(Server.MapPath(".") + "/" + fecha))
    {
        StreamReader arch = new StreamReader(Server.MapPath(".") + "/" + fecha);
        TextBox1.Text = arch.ReadToEnd();
        arch.Close();
    }
    else

```

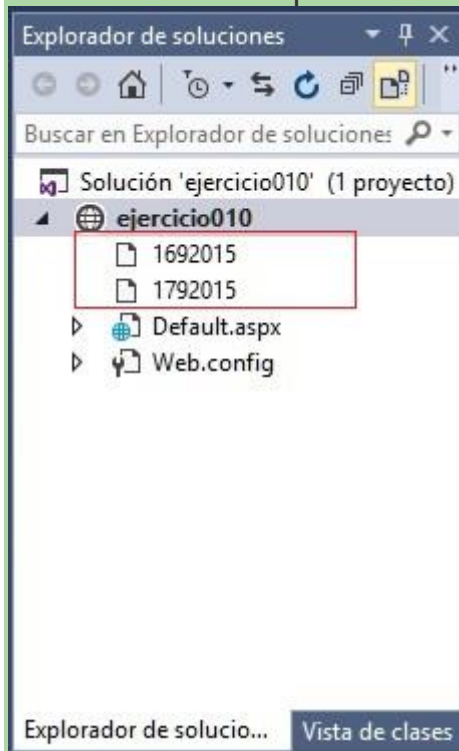
Creación, lectura y escritura de archivos de texto.

```
        TextBox1.Text = "";  
    }
```

Finalmente en el evento "Page_Load" recuperamos la fecha actual y si hay un archivo con dicho nombre lo mostramos:

```
protected void Page_Load(object sender, EventArgs e)  
{  
    string fecha = Calendar1.TodaysDate.Day.ToString() +  
        Calendar1.TodaysDate.Month.ToString() +  
        Calendar1.TodaysDate.Year.ToString();  
    if (File.Exists(Server.MapPath(".") + "/" + fecha))  
    {  
        StreamReader arch = new StreamReader(Server.MapPath(".") + "/" + fecha);  
        TextBox1.Text = arch.ReadToEnd();  
        arch.Close();  
    }  
}
```

Si presionamos el botón derecho dentro del "Explorador de soluciones" donde dice "ejercicio010" podemos seleccionar la opción "Actualizar carpeta" y comprobar que se han creado un archivo por cada fecha donde guardamos actividades:



Ejercicio propuesto

1 ? Confeccionar un sitio que permita ingresar los datos del curriculum de una persona y almacenarlos en un archivo de texto.