

Prácticas

Tema 3 – Programación basada en código de servidor embebido

Objetivos	<p>En esta práctica, se aborda la realización de tareas relacionadas con:</p> <ul style="list-style-type: none"> • Crear un procedimiento de evento Page_Load para un Web Form de ASP.NET • Crear procedimientos de evento para controles de servidor Web en un Web Form de ASP.NET
Requisitos previos	<p>Para realizar esta práctica, es necesario tener conocimientos sobre:</p> <ul style="list-style-type: none"> • Programación lógica, estructuras de datos, funciones, objetos y eventos. • Utilización de código procedural para la representación de la lógica del negocio de la aplicación Web. • Lenguaje de programación Microsoft Visual C#. <p>Para realizar esta práctica, es necesario haber realizado los ejercicios de la práctica correspondiente al Tema 2</p>
Escenario	<p>La empresa ACME, S.FIG. dedicada a investigación, desarrollo e innovación industrial ofrece varios beneficios a sus empleados a través de diversas prestaciones sociales.</p> <p>En la práctica anterior, se crearon los formularios <i>Empleados.aspx</i> y <i>Prestaciones.aspx</i> sobre una solución de Visual Studio .NET denominada <i>Acme</i> y un proyecto de aplicación Web para la aplicación de sitio Web <i>GesPresta</i>.</p> <p>En la práctica correspondiente a este tema, se agregará código lógico de procesamiento en el servidor para recoger y visualizar como respuesta los valores introducidos en cada formulario y enviados al servidor Web. Para ello, se editará el código lógico en los procedimientos de evento adecuados sobre los archivos de código subyacente correspondientes a las páginas de ASP.NET.</p>

Ejercicio 1

Crear un procedimiento de evento Page_Load

En este ejercicio, editaremos código asociado al evento **Page_Load**, cuyo significado se podría establecer como “al cargar la página”, sobre las páginas *Empleados.aspx* y *Prestaciones.aspx*

Situar el foco sobre el primer control del formulario

1. Utilizando Visual Studio .NET, abrir la solución *Acme* correspondiente a una copia del resultado alcanzado al finalizar la práctica del tema anterior. Conviene realizar los ejercicios de esta práctica sobre una copia del resultado de la práctica anterior para, de esta forma, guardar el estado del trabajo realizado sobre la solución *Acme* y el sitio Web *GesPresta* al finalizar la práctica anterior. Esto mismo debe hacerse al comenzar cada práctica.
2. Abrir la página *Empleados.aspx* del proyecto *GesPresta*.
3. En la vista Diseño, hacer doble clic en el fondo del Web Form *Empleados.aspx* para crear un procedimiento de evento asociado al evento **Page_Load**. Como consecuencia, se abre el archivo de código subyacente *Empleados.aspx.cs* y se agrega una plantilla para el procedimiento de evento **Page_Load**, que contiene el siguiente código de Visual C#:

```
protected void Page_Load(object sender, EventArgs e)
{
}
```

4. Para situar el foco al cargar la página, en el primer elemento del formulario *Empleados.aspx*, sobre el cuadro de texto CodEmp, el código C# a añadir debería ser similar al siguiente:

```
protected void Page_Load(object sender, EventArgs e)
{
    txtCodEmp.Focus();    // Sitúa el foco en el elemento Código Empleado
}
```

5. Realizar la misma operación con la página *Prestaciones.aspx*, añadiendo el código correspondiente sobre el archivo de código subyacente *Prestaciones.aspx.cs*

```
protected void Page_Load(object sender, EventArgs e)
{
    txtCodPre.Focus();    // Sitúa el foco en el elemento Código Prestación
}
```

6. Iniciar la depuración para comprobar los resultados.

Ejercicio 2

Crear un procedimiento de evento Click

En este ejercicio, editaremos código asociado al evento **Click** del botón de envío para recoger y mostrar los valores de los controles incluidos en el formulario de la página *Empleados.aspx*.

Mostrar los valores de los controles de un Web Form: Empleados1.aspx

- Para no modificar demasiado el trabajo realizado hasta ahora, vamos a obtener una copia de la página *Empleados.aspx* denominada *Empleados1.aspx*. Para ello:
 - En el **Explorador de Soluciones**, seleccionar el Web Form *Empleados.aspx*.
 - Hacer clic sobre el botón derecho y seleccionar la opción **Copiar**.
 - En el **Explorador de Soluciones**, seleccionar el proyecto *GesPresta*.
 - Hacer clic sobre el botón derecho y seleccionar la opción **Pegar**. Como se puede comprobar, se ha añadido al sitio Web el Web Form *Copia de Empleados.aspx*.
 - Finalmente, cambiar el nombre de *Copia de Empleados.aspx* a *Empleados1.aspx*, seleccionándolo en el **Explorador de Soluciones** y haciendo clic sobre el botón derecho para seleccionar la opción **Cambiar Nombre**.
- Abrir la página *Empleados1.aspx*.
- En la vista diseño, añadir desde el **Cuadro de Herramientas** un nuevo control de tipo **Label** al final de la página. Modificar sus propiedades de la siguiente forma:

Control	ID	Text	BackColor	Width	Visible
Label	lblValores		#66FFFF	60%	False

- A continuación, editar el procedimiento asociado al evento **Click** de control **Button** de envío del formulario, denominado *cmdEnviar*, para recoger los valores de los controles del Web Form y mostrarlos sobre la etiqueta **lblValores**. El código C# podría ser similar al siguiente:

```
protected void cmdEnviar_Click(object sender, EventArgs e)
{
    lblValores.Visible = true;
    lblValores.Text = "VALORES DEL FORMULARIO" +
        "<br/> Código Empleado: " + txtCodEmp.Text +
        "<br/> NIF: " + txtNifEmp.Text +
        "<br/> Apellidos y Nombre: " + txtNomEmp.Text +
        "<br/> Dirección: " + txtDirEmp.Text +
        "<br/> Ciudad: " + txtCiuEmp.Text +
        "<br/> Teléfonos: " + txtTelEmp.Text +
        "<br/> Fecha de Nacimiento: " + txtFnaEmp.Text +
        "<br/> Fecha de Incorporación: " + txtFinEmp.Text +
        "<br/> Sexo: " + rblSexEmp.SelectedItem.Value +
        "<br/> Departamento: " + ddlDepEmp.Text;
}
```

Para editar el procedimiento de evento **Click** del control *cmdEnviar*, hacer:

- En la vista Diseño del Web Form, hacer clic sobre el control para seleccionarlo.
- En la ventana de Propiedades, seleccionar el icono **Eventos** cuyo símbolo es un rayo.
- Hacer doble clic sobre el nombre del evento que se desea editar.

Otra forma de acceder a los procedimientos de evento más habitualmente utilizados de cada control es hacer doble clic sobre el control en la vista Diseño.

5. Iniciar la depuración para comprobar los resultados. Tras introducir los datos y pulsar el botón **Enviar**, el resultado debería ser similar al que muestra la siguiente ilustración:

DATOS DE LOS EMPEADOS

Código Empleado	<input type="text" value="A-02872"/>
NIF	<input type="text" value="21345678-M"/>
Apellidos y Nombre	<input type="text" value="Gilabert Moreno, Juan J."/>
Dirección	<input type="text" value="C/ La Vega, 23 - 1º"/>
Ciudad	<input type="text" value="03005 Alicante (Alicante)"/>
Teléfonos	<input type="text" value="966109900"/>
Fecha de Nacimiento	<input type="text" value="12/03/1967"/>
Fecha de Ingreso	<input type="text" value="01/01/2001"/>
Sexo	<input type="radio"/> Hombre <input checked="" type="radio"/> Mujer
Departamento	<input type="text" value="Administración"/>

VALORES DEL FORMULARIO

Código Empleado: A-02872
Nif: 21345678-M
Apellidos y Nombre: Gilabert Moreno, Juan J.
Dirección: C/ La Vega, 23 - 1º
Ciudad: 03005 Alicante (Alicante)
Teléfonos: 966109900
Fecha de Nacimiento: 12/03/1967
Fecha de Incorporación: 01/01/2001
Sexo: M
Departamento: Administración

Al comprobar funcionamiento del Web Form *Empleados1.aspx*, podrá comprobarse que los valores que incluyen los controles de servidor que conforman el formulario, se recogen en el mismo Web Form tras el envío de los mismos desde el cliente hacia el servidor. Este modo de operación, denominado **Post-Back**, que está establecido de manera predeterminada cuando se trabaja con ASP.NET, facilita el trabajo de desarrollo en un entorno de desarrollo visual, orientado a objetos y controlado por eventos sobre interfaz Web.

Además, puede observarse que en la vista **Código** del Web Form *Empleados1.aspx* no existe ninguna referencia hacia ningún destino de procesamiento de los datos contenidos en los campos del formulario, ni en la etiqueta `<form>`, ni en la etiqueta `<asp:Button>`. Esto es porque el destino del procesamiento de los valores de los controles de servidor que forman el formulario lo realiza el mismo Web Form de manera predeterminada.

Ejercicio 3

Pasar valores de un formulario a otro Web Form de ASP.NET

Si bien el modo de trabajo predeterminado y recomendado es el que utiliza la opción **Post-Back** de ASP.NET cuando se trabaja con Web Forms, tal como se ha utilizado en el ejercicio anterior. En algún caso, puede ser necesario enviar y recoger los valores de los controles de un formulario en páginas de ASP.NET distintas. En este ejercicio, se editará código asociado al evento **Click** del botón de envío para recoger y mostrar los valores de los controles del Web Form *Prestaciones.aspx* en otra página.

Mostrar los valores de un formulario Web Form en otro Web Form: Prestaciones1.aspx

1. Como en el ejercicio anterior, conviene realizar una copia del Web Form *Prestaciones.aspx*, denominada *Prestaciones1.aspx*, para realizar el trabajo sobre ella y así, no modificar demasiado el trabajo realizado hasta ahora.
2. Abrir la página *Prestaciones1.aspx*.
3. En la **vista Código** de *Prestaciones1.aspx*, añadir el atributo **PostBackUrl** en la etiqueta correspondiente al botón **cmdEnviar**, para establecer como página de destino del procesamiento el Web Form *Prestaciones1Respuesta.aspx*, tal como se muestra a continuación:

```
<asp:Button ID="cmdEnviar" runat="server" Text="Enviar"
    PostBackUrl="~/prestaciones1Respuesta.aspx" />
```

Para introducir el carácter ~, pulsar las teclas **<AltGr> + <4>**.

4. Crear un nuevo Web Form denominado *Prestaciones1Respuesta.aspx*. También sería posible completar esta tarea y la anterior, creando primero Web Form *Prestaciones1Respuesta.aspx* y después, accediendo a la propiedad **PostBackUrl** del botón de comando **cmdEnviar** para seleccionar el Web Form *Prestaciones1Respuesta.aspx*.
5. En el Web Form *Prestaciones1Respuesta.aspx*, añadir cuatro controles de tipo **Label** para mostrar, empleando cuatro formas distintas, los valores enviados desde el formulario del Web Form *Prestaciones1.aspx*. Modificar sus propiedades de la siguiente forma:

Control	ID	Text	BackColor	Width
Label	lblValores1		#C0FFFF	70%
Label	lblValores2		#C0FFC0	70%
Label	lblValores3		#FFC0C0	70%
Label	lblValores4		#FFFC00	70%

6. A continuación, se editará el código lógico correspondiente al procedimiento de evento **Load** del objeto **Page** del Web Form página *prestaciones1Respuesta.aspx*, de modo que se recojan y muestren los valores de los controles de servidor enviados desde el cliente a través del Web Form *Prestaciones1.aspx*. Sobre cada uno de los cuatro controles **Label** definidos en el Web Form se mostrarán los valores recibidos en el servidor Web, empleado para ello cuatro formas diferentes de recogerlos. El código lógico de Visual C# a incluir en el procedimiento de evento **Load** del objeto **Page** se muestra a continuación:

```
protected void Page_Load(object sender, EventArgs e)
{
    // Método 1: Acceso directo por ID a los valores de la colección Request.Form
    string cadena = "";
    cadena = cadena + "Código: " + Request.Form["txtCodPre"] + "<br/>";
    cadena = cadena + "Descripción: " + Request.Form["txtDesPre"] + "<br/>";
    cadena = cadena + "Importe: " + Request.Form["txtImpPre"] + "<br/>";
    cadena = cadena + "Porcentaje: " + Request.Form["txtPorPre"] + "<br/>";
    cadena = cadena + "Tipo de Prestación: " + Request.Form["ddlTipPre"] + "<br/>";
    lblValores1.Text = cadena;

    // Método 2: Recorrido dinámico de la Colección Request.Form
    System.Collections.Specialized.NameValueCollection valor1 = Request.Form;
    cadena = "";
    foreach (string campo in Request.Form)
    {
        if (campo.Substring(0, 2) != "__") // No considera campos que empiezan por __
        {
            cadena = cadena + campo + ": " + valor1[campo] + "<br/>";
        }
    }
    lblValores2.Text = cadena;

    // Método 3: Recorrido dinámico de la Colección Request.Form (Índice numérico)
    System.Collections.Specialized.NameValueCollection valor2 = Request.Form;
    cadena = "";
    int i = 0;
    foreach (string campo in Request.Form)
    {
        if (campo.Substring(0, 2) != "__") // No considera campos que empiezan por __
        {
            cadena = cadena + campo + ": " + valor2[i] + "<br/>";
        }
        i++;
    }
    lblValores3.Text = cadena;

    // Método 4: Compartir valores de controles entre Web Forms en el mismo sitio Web
    cadena = "";
    if (Page.PreviousPage != null)
    {
        cadena = cadena + ((TextBox)Page.PreviousPage.FindControl("txtCodPre")).Text + "<br/>";
        cadena = cadena + ((TextBox)Page.PreviousPage.FindControl("txtDesPre")).Text + "<br/>";
        cadena = cadena + ((TextBox)Page.PreviousPage.FindControl("txtImpPre")).Text + "<br/>";
        cadena = cadena + ((TextBox)Page.PreviousPage.FindControl("txtPorPre")).Text + "<br/>";
        cadena = cadena + ((DropDownList)Page.PreviousPage.FindControl("ddlTipPre")).Text;
        cadena = cadena + "<br/>";
    }
    else
    {
        cadena = "No existe el Web Form origen";
    }
    lblValores4.Text = cadena;
}
```

Como puede apreciarse en el código anterior, los métodos 2 y 3 realizan un recorrido sobre la colección *Form* del objeto *Request* que contiene toda la información del encabezado HTTP que acompaña la solicitud desde el cliente. Los nombre de las variables de encabezado HTTP que contienen valores de los campos del formulario comienzan siempre por dos caracteres de subrayado, “__”. Es posible visualizar todas las variables de encabezado, y sus valores, enviadas desde el cliente comentando en el código la estructura de selección *if* en el código.

7. Iniciar la depuración del Web Form *Prestaciones1.aspx*, introducir los datos en los controles del formulario y pulsar el botón **Enviar**.

[Inicio](#) [Empleados](#) [Prestaciones](#)

ACME INNOVACIÓN, S. FIG.

Gestión de Prestaciones Sociales

DATOS DE LAS PRESTACIONES - 1

Código Prestación	<input type="text" value="LENTDE"/>
Descripción	<input type="text" value="Lentillas desechables"/>
Importe Fijo	<input type="text" value="60,00 €"/>
Porcentaje del Importe	<input type="text"/>
Tipo de Prestación	<input type="text" value="Ocular"/>

El resultado en la página ASP.NET *Prestaciones1Respuesta.aspx*, será similar al siguiente:

VALORES RECIBIDOS DEL FORMULARIO prestaciones1.aspx

Código: LENTDE
Descripción: Lentillas desechables
Importe: 60,00 €
Porcentaje:
Tipo de Prestación: Ocular

txtCodPre: LENTDE
txtDesPre: Lentillas desechables
txtImpPre: 60,00 €
txtPorPre:
ddlTipPre: Ocular
cmdEnviar: Enviar

txtCodPre: LENTDE
txtDesPre: Lentillas desechables
txtImpPre: 60,00 €
txtPorPre:
ddlTipPre: Ocular
cmdEnviar: Enviar

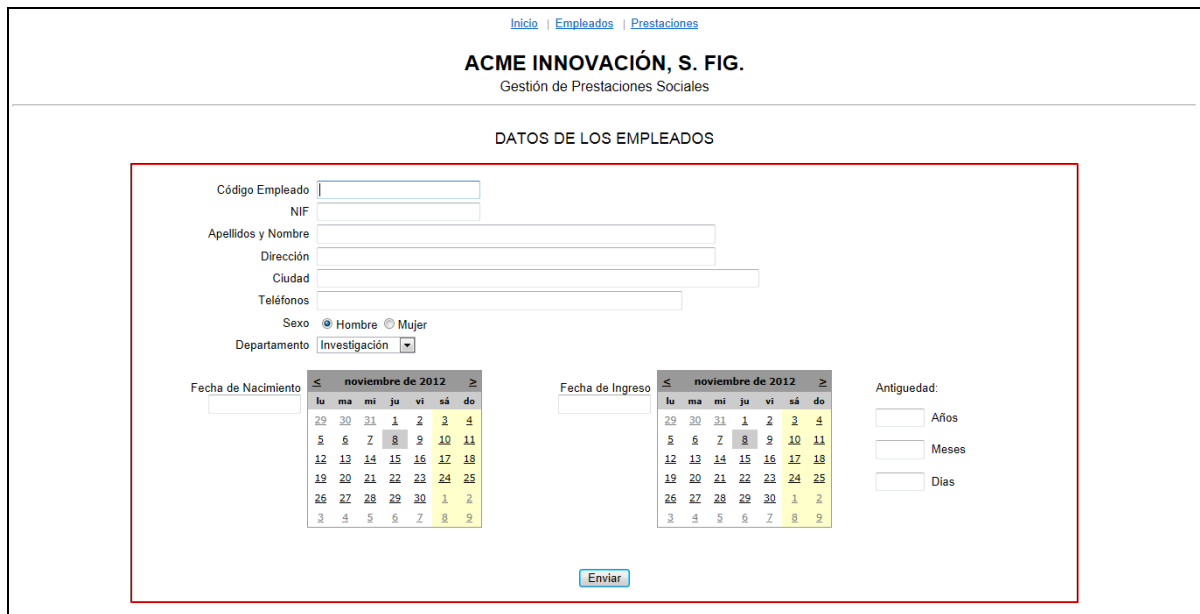
LENTDE
Lentillas desechables
60,00 €

Ocular

Ejercicio 4 (opcional)

Supuesto práctico de programación

Se propone el siguiente supuesto práctico de ampliación de conocimientos. Sobre la base del Web Form *Empleados1.aspx*, crear un nuevo Web Form denominado *EmpleadosCalendar.aspx* que incorpore dos controles Calendar para representar las fechas de nacimiento del empleado y de ingreso en la compañía. La apariencia del Web Form puede ser similar a la siguiente.



A continuación, incluir código en los eventos **SelectionChanged** de los dos controles Calendar para obtener los siguientes controles de error sobre los datos introducidos por el usuario:

- Mostrar un error mediante un control *Label* denominado *lblError1*, cuando la fecha de ingreso en la compañía sea menor que la fecha de nacimiento del empleado.
- Mostrar un error mediante un control *Label* denominado *lblError2*, cuando la fecha de ingreso en la compañía sea mayor que la fecha actual.

Para obtener la fecha actual del sistema y asignarla a una variable de tipo *DateTime* emplear el siguiente código:

```
DateTime dtHoy = System.DateTime.Now;
```

En caso que se produzca alguno de los dos errores anteriores, deberá deshabilitarse el control *Button* denominado *btnEnviar* que envía los datos del Web Form.

Además, al introducir una fecha de ingreso válida se calculará la antigüedad del empleado en años, meses y días para presentarla sobre los controles *TextBox* denominados *txtAnyos*, *txtMeses* y *txtDías*, respectivamente, que aparecen en la parte inferior derecha del Web Form considerando la apariencia de la ilustración anterior.