

Prácticas

Tema 3 – Programación basada en código lógico de servidor

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 a nivel de página, como por ejemplo el evento Page_Load, para un Web Form de ASP.NET.• Crear procedimientos de evento para los controles de servidor 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 o procedural: 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 Visual C#. <p>Para realizar esta práctica, es necesario haber realizado los ejercicios de la práctica correspondiente al Tema 2. Conviene desarrollar los ejercicios de esta práctica sobre una copia correspondiente al estado de la aplicación Web al finalizar la práctica correspondiente al Tema 2.</p>
Escenario	<p>La empresa ACME, S.FIG., que se dedica a la investigación, el desarrollo y la innovación industrial, ofrece diversos beneficios de ayuda social a sus empleados a través de la gestión prestaciones sociales.</p> <p>En la práctica anterior, se crearon los Web Forms <i>Empleados.aspx</i> y <i>Prestaciones.aspx</i> sobre una Solución de Visual Studio denominada <i>Acme</i> y un proyecto de Aplicación Web basada en ASP.NET denominada <i>GesPresta</i>.</p> <p>En la práctica correspondiente a este tema, se agregará código lógico de procesamiento en el servidor para recoger los valores introducidos en cada formulario que son enviados al servidor Web y para visualizarlos en el navegador Web como respuesta. Para ello, se editará código lógico en los procedimientos de evento adecuados sobre los archivos de código subyacente correspondientes a los Web Forms ya creados.</p> <p>El último de ejercicio de esta práctica corresponde con un supuesto práctico de aplicación de la programación basada en código lógico de procesamiento en el servidor. La resolución correcta de este ejercicio puede ayudar a comprender las particularidades concretas que incorpora este tipo de programación.</p>

Ejercicio 1

Crear un procedimiento de evento Page_Load

En este ejercicio, se va a añadir código lógico al evento **Page_Load**, cuyo significado se podría definir como “al cargar la página”, sobre los Web Forms *Empleados.aspx* y *Prestaciones.aspx*

Es muy importante, 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 la Aplicación Web *GesPresta* al finalizar la práctica anterior. Esto mismo deberá hacerse al comenzar cada nueva práctica. Además, esto facilitará la entrega de las prácticas.

Situar el foco sobre el primer control del formulario

1. Utilizando Microsoft Visual Studio, abrir la solución *Acme* correspondiente a una copia del resultado alcanzado al finalizar la práctica del tema anterior.
2. Abrir la página *Empleados.aspx* del proyecto *GesPresta*.
3. Abrir el archivo de código subyacente *Empleados.aspx.cs*. Puede apreciarse que, de manera predeterminada, el archivo de código subyacente incorpora el prototipo del procedimiento de evento **Page_Load** que estará especificado de manera similar a la siguiente:

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

El procedimiento de evento **Page_Load** puede incluir el código lógico asociado al evento Load, o “al cargar”, del objeto Page, que representa la página de ASP.NET. Es decir, en este procedimiento de evento se puede incluir el código que se ejecutará al cargar el Web Form.

4. Para situar el foco en el cuadro de texto *txtCodEmp* al cargar el Web Form *Empleados.aspx*, añadir el siguiente código lógico escrito en lenguaje de programación C# en el procedimiento de evento **Page_Load** del archivo de código subyacente *Empleados.aspx.cs*:

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

5. Realizar una operación similar para situar el foco sobre el primer cuadro de texto al cargar el Web Form *Prestaciones.aspx*. Para ello, añadir el código lógico correspondiente en el procedimiento de evento **Page_Load** del 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 obtenidos en ambos Web Forms.

Ejercicio 2

Crear un procedimiento de evento Click

En este ejercicio, se agregará código lógico asociado al evento **Click** del botón de envío para poder obtener y mostrar los valores de los controles incluidos en el formulario de empleados.

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 habrá añadido al proyecto el Web Form *Empleados - Copia.aspx*.
 - Cambiar el nombre del Web Form *Empleados - Copia.aspx* a *Empleados1.aspx*, seleccionándolo en el **Explorador de Soluciones** y haciendo clic sobre el botón derecho para seleccionar la opción **Cambiar Nombre**.
 - Cuando se realiza una copia de un Web Form en el proyecto es necesario cambiar el nombre de la clase de código del archivo de código subyacente, sino se producen errores de compilación porque el nombre de la clase ya existe en otro Web Form. Para ello, abrir el archivo *Empleados1.aspx* y modificar el valor del parámetro **Inherits** de la directiva **@Page** a *GesPresta.Empleados1*. Y, abrir el archivo *Empleados1.aspx.cs* y cambiar el nombre de la clase de código a *Empleados1*.
- Iniciar la depuración del Web Form *Empleados1.aspx* para comprobar que no hay errores.
- En la vista **Diseño** o **Código** del Web Form *Empleados1.aspx*, añadir desde la paleta **Estándar** del **Cuadro de Herramientas** un control de tipo **Label** al final de la página. Y, modificar sus propiedades de la siguiente forma:

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

- A continuación, se va a editar el procedimiento de evento asociado al evento **Click** del control de tipo **Button** denominado *cmdEnviar* en el Web Form *Empleados1.aspx*. Este control permite realizar el envío y procesar el formulario en el servidor Web. El código lógico que se va a incluir permitirá recoger los valores de los controles del Web Form y mostrarlos sobre la etiqueta *lblValores*. Para editar el procedimiento de evento **Click** del control *cmdEnviar*, hacer:
 - En la vista **Diseño** del Web Form, seleccionar el control de tipo **Button** cuya propiedad **ID** es *cmdEnviar* haciendo clic sobre él.
 - En la ventana **Propiedades**, seleccionar el icono **Eventos** cuyo símbolo es un rayo.
 - En la ventana **Propiedades**, buscar el nombre del evento que se desea editar, en este caso se trata del evento **Click**, y hacer doble clic sobre el nombre del evento.Otra forma de acceder al procedimiento de evento más utilizado o habitual de cada tipo de control es hacer doble clic sobre el control en la vista **Diseño** del Web Form.
- Una vez que se haya accedido al evento **Click** del control de tipo **Button** denominado *cmdEnviar*, introducir el siguiente código lógico escrito en C#:

```
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;
}
```

6. 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 observarse que los valores introducidos en los controles se recogen en el mismo Web Form, tras el envío desde el cliente hacia el servidor Web. Además, puede comprobarse que no existe ninguna referencia hacia ningún destino de procesamiento en el código, ni en la etiqueta `<form>`, ni en la etiqueta `<asp:Button>`. Esto es porque, de manera predeterminada, el destino del procesamiento es el mismo Web Form. Y, además, dado que se utilizan controles que son procesados en el servidor Web, entonces los controles de servidor están disponibles desde el código lógico de procesamiento en el servidor, lo que facilita el desarrollo.

Ejercicio 3

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

Cuando se trabaja con Web Forms de ASP.NET, el modo predeterminado y recomendado de recoger los valores de los controles del formulario es el que se ha utilizado en el ejercicio anterior. Como se ha podido comprobar en el ejercicio anterior, se accede a los valores de los controles de servidor a través de las propiedades correspondientes, por ejemplo, a través de la propiedad **Text** de los controles de tipo **TextBox**. Los valores de los controles de servidor son accesibles desde el código lógico de servidor porque se mantiene el estado de la página, es decir, porque se puede acceder a los valores de los controles de la página de ASP.NET en cada acción de ida y vuelta (*Post-Back*). Ello es debido a que, de manera predeterminada, el destino del envío de un Web Form de ASP.NET es el mismo Web Form y también es debido a que se emplean controles de servidor. Sin embargo, en algún caso muy concreto podría ocurrir que se desearan recoger los valores de los controles de un formulario en un Web Form distinto. En este ejercicio, se define 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 de ASP.NET.

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

1. Como se hizo en el ejercicio anterior, hacer una copia del Web Form *Prestaciones.aspx*, denominada *Prestaciones1.aspx*, para realizar el trabajo sobre ella y así, no modificar 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 del 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 la combinación de 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 un control de tipo **Label** para mostrar los valores enviados desde el formulario del Web Form *Prestaciones1.aspx* y modificar sus propiedades de la siguiente forma:

Control	ID	Text	BackColor	Width
Label	lblValores		#C0FFFF	70%

6. A continuación, se editará el código lógico correspondiente al procedimiento de evento **Load** del objeto **Page** del Web Form *Prestaciones1Respuesta.aspx*, para recoger y mostrar los valores de los controles enviados desde el Web Form *Prestaciones1.aspx*. El código lógico de C# a incluir en el procedimiento de evento **Load** del objeto **Page** se muestra a continuación:

```
protected void Page_Load(object sender, EventArgs e)
{
    // Acceso directo por ID a los valores de la colección Form del objeto Request
    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/>";
    lblValores.Text = cadena;
    lblValores.Visible = true;
}
```

Como puede apreciarse en el código anterior, en este caso se utiliza la colección *Form* del objeto *Request* de ASP.NET para recoger los valores de los controles enviados desde el Web Form *Prestaciones1.aspx*. Los valores recogidos se muestran sobre el control **Label** *lblValores*.

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="button" value="Ocular"/>

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

VALORES RECIBIDOS DESDE EL FORMULARIO PRESTACIONES1.ASPX

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

Ejercicio 4

Supuesto práctico sobre código lógico de servidor

Partiendo del Web Form *Empleados1.aspx*, se propone realizar una copia para crear un nuevo Web Form denominado *EmpleadosCalendar.aspx* que incorpore dos controles de tipo **Calendar** para representar la fecha de nacimiento y la fecha de ingreso del empleado en la compañía Acme. La apariencia del nuevo Web Form *EmpleadosCalendar.aspx* puede ser similar a la siguiente.

Inicio | Empleados | Prestaciones

ACME INNOVACIÓN, S. FIG.
Gestión de Prestaciones Sociales

DATOS DE LOS EMPLEADOS

Código Empleado

NIF

Apellidos y Nombre

Dirección

Ciudad

Teléfonos

Sexo ☒ Hombre ☐ Mujer

Departamento

Fecha de Nacimiento

Fecha de Ingreso

Antigüedad:

Años

Meses

Días

A continuación, incluir código lógico en los eventos **SelectionChanged** de los dos controles de tipo **Calendar** para gestionar los errores de validación 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.
- Mostrar un error mediante un control *Label* denominado *lblError3*, cuando la fecha de nacimiento sea mayor que la fecha actual.

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

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

Para obtener la fecha seleccionada en un control de tipo **Calendar** se utiliza la propiedad **SelectedDate**.

Las fechas que se seleccionen en los controles de tipo **Calendar** deberán visualizarse, respectivamente, en los controles de tipo **TextBox** correspondientes a la fecha de nacimiento y la fecha de incorporación.

Además de la gestión de errores, se desea que, al introducir una fecha de ingreso válida, se calcule y visualice la antigüedad del empleado en años, meses y días sobre los controles *TextBox* que aparecen en la parte inferior derecha del Web Form, considerando la apariencia de la ilustración anterior.

Puede existir diversas formas de calcular la antigüedad del empleado en la empresa ACME en días, meses y años, aunque una de las formas más sencillas de obtener ese cálculo podría ser la que se muestra en la siguiente porción de código de ejemplo:

```
DateTime dtHoy = System.DateTime.Now;
TimeSpan diferencia = dtHoy - cFinEmp.SelectedDate;
DateTime fechamin = new DateTime(1, 1, 1);

txtAños.Text = ((fechamin + diferencia).Year - 1).ToString();
txtMeses.Text = ((fechamin + diferencia).Month - 1).ToString();
txtDias.Text = ((fechamin + diferencia).Day).ToString();
```

Para resolver el ejercicio se recomienda crear funciones que resuelvan la gestión y visualización de los errores, así como el cálculo de la antigüedad, de manera que podrán establecerse las llamadas a estas funciones que sean necesarias, desde los procedimientos de evento, para completar el ejercicio.

Parte opcional del ejercicio

Opcionalmente, también pueden utilizarse los dos cuadros de texto de tipo **TextBox** disponibles en la interfaz Web para poder introducir o editar la fecha de nacimiento y la fecha de ingreso mediante texto, además de los controles de tipo **Calendar** ya existentes en el Web Form.

Para resolver esta parte opcional deberá utilizarse el evento **Change** de los cuadros de texto, así como asignar la propiedad **AutoPostBack** de estos dos cuadros de texto al valor *true*, para que se pueda producir el envío de datos y el correspondiente procesamiento en el Servidor Web (*Post-Back*), después de finalizar la edición del valor de la fecha de nacimiento o de la fecha de ingreso en el cuadro de texto correspondiente. Estos dos cuadros de texto se emplearán, por tanto, tanto para visualizar, como para introducir o editar, respectivamente, la fecha de nacimiento y la fecha de incorporación.

El modo de asignar y hacer visible la fecha introducida o editada en los cuadros de texto sobre el control de tipo **Calendar** correspondiente, se realiza empleando las propiedades **SelectedDate** y **VisibleDate** del control Calendar, por ejemplo, de la siguiente forma:

```
Calendar1.SelectedDate = Convert.ToDateTime(txtFnaEmp.Text);
Calendar1.VisibleDate = Convert.ToDateTime(txtFnaEmp.Text);
```

La propiedad **SelectedDate** de un control de tipo **Calendar** permite obtener o establecer la fecha. Y la propiedad **VisibleDate** hace visible la fecha asignada a esta propiedad en el control de tipo **Calendar**.