

Prácticas

Tema 4 – Desarrollo de aplicaciones Web

Objetivos	<p>En esta práctica, se aborda la realización de tareas relacionadas con:</p> <ul style="list-style-type: none">• Conocer los aspectos relacionados con la detección de errores en tiempo de ejecución y la ejecución de código en modo depuración.• Habilitar y deshabilitar el seguimiento a nivel de página y de aplicación.• Conocer y emplear los distintos controles de validación de ASP.NET.• Mostrar mensajes descriptivos de validación de entrada de datos.• Crear e insertar controles de usuario en Web Forms.• Intercambiar datos con controles de usuario.• Crear y utilizar páginas maestras o <i>Master Pages</i> en las aplicaciones Web para combinar la presentación visual con páginas de contenido.
Requisitos previos	<p>Para realizar esta práctica, es necesario tener conocimientos sobre:</p> <ul style="list-style-type: none">• Manejo de herramientas de depuración en entornos integrados de desarrollo.• Utilización de controles de servidor en páginas de ASP.NET.• Programación lógica en páginas Web de servidor, empleando archivos de código subyacente.• Lenguaje de programación Microsoft Visual C#. <p>Para realizar esta práctica, es necesario haber realizado los ejercicios de la práctica correspondiente al Tema 3. Se recomienda continuar esta práctica sobre una copia de la aplicación Web al finalizar la práctica del Tema 3.</p>
Escenario	<p>Se continúa con el desarrollo de una aplicación Web para la gestión de ayudas a los empleados a través de prestaciones sociales en la compañía Acme. En las prácticas anteriores, 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 denominado <i>GesPresta</i>. Además, se añadió código lógico para recoger y presentar los datos enviados como respuesta los valores introducidos en cada Web Form.</p> <p>Esta práctica aborda la incorporación de diversas funcionalidades específicas, que se suelen incorporar al desarrollo de las aplicaciones Web de ASP.NET, y que facilitan y amplían las prestaciones y funcionalidad del propio desarrollo:</p> <ul style="list-style-type: none">• Depuración y corrección de errores.• Validación de entradas de usuario.• Creación de controles de usuario.• Utilización de páginas maestras o <i>Master Pages</i>.

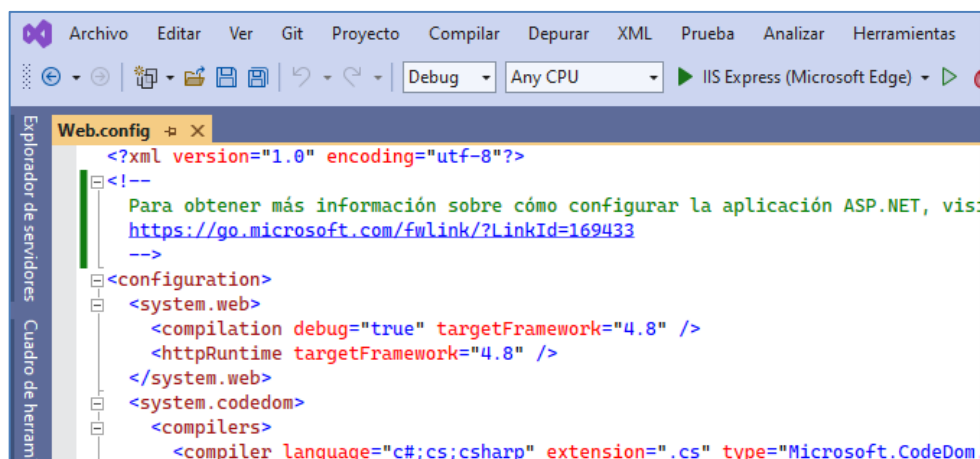
PARTE 1. DEPURACIÓN Y CORRECCIÓN DE ERRORES

Ejercicio 1

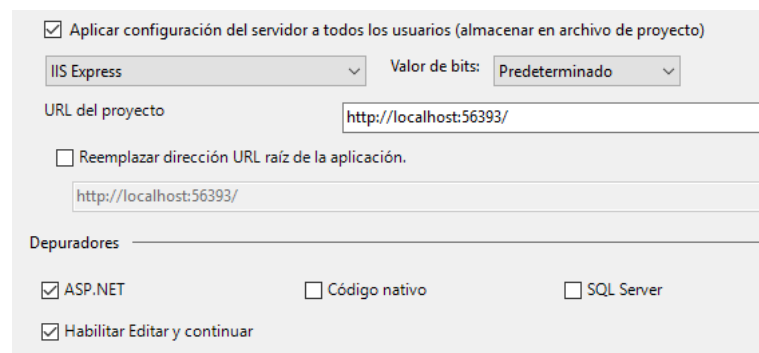
Depuración de errores de compilación y de errores en tiempo de ejecución

En este ejercicio se aborda el funcionamiento del depurador integrado en Visual Studio para mostrar los errores de compilación y los errores que se producen en tiempo de ejecución.

1. Abrir la Solución *Acme* correspondiente a una copia del resultado alcanzado en la práctica del tema anterior.
2. Comprobar si está activado el depurador para la aplicación Web de ASP.NET *GesPresta*. Para ello, realizar las siguientes acciones:
 - a. Usando el **Explorador de soluciones**, abrir el archivo de configuración *Web.Config* para comprobar si el valor del atributo *debug* del elemento `<compilation>` tiene el valor **true**.



- b. En el **Explorador de soluciones**, hacer clic con el botón secundario en el nombre del proyecto de aplicación Web *GesPresta* y seleccione la opción **Propiedades**. En la ventana de propiedades del proyecto, seleccionar la opción **Web** y comprobar que está activada la casilla de verificación **ASP.NET** en el apartado de **Depuradores**.



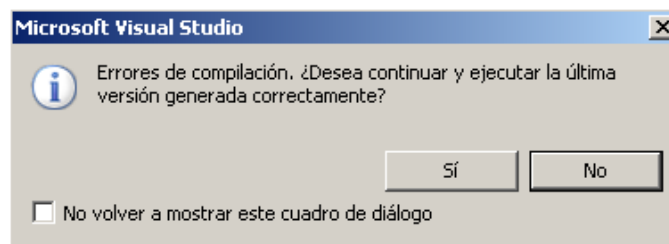
Para tener habilitada la depuración, deben ocurrir las dos circunstancias anteriores. Es decir, debe estar habilitada en el archivo *Web.config* y en la página de propiedades del proyecto. Cuando una aplicación Web está en fase de desarrollo conviene tener activada la depuración para que el depurador integrado en Visual Studio vaya informando a los desarrolladores de los errores que se puedan producir. Sin embargo, la depuración deberá desactivarse cuando la aplicación Web se encuentra en la fase de explotación por parte de los usuarios finales.

3. A continuación, se provocará un error de programación para conocer el comportamiento del depurador incorporado de Visual Studio cuando se produce un **error de compilación**. Para ello, abrir el archivo de código subyacente *Empleados.aspx.cs* del proyecto *GesPresta*.
4. Añadir la siguiente línea de código en el evento **Load** del objeto **Page**.

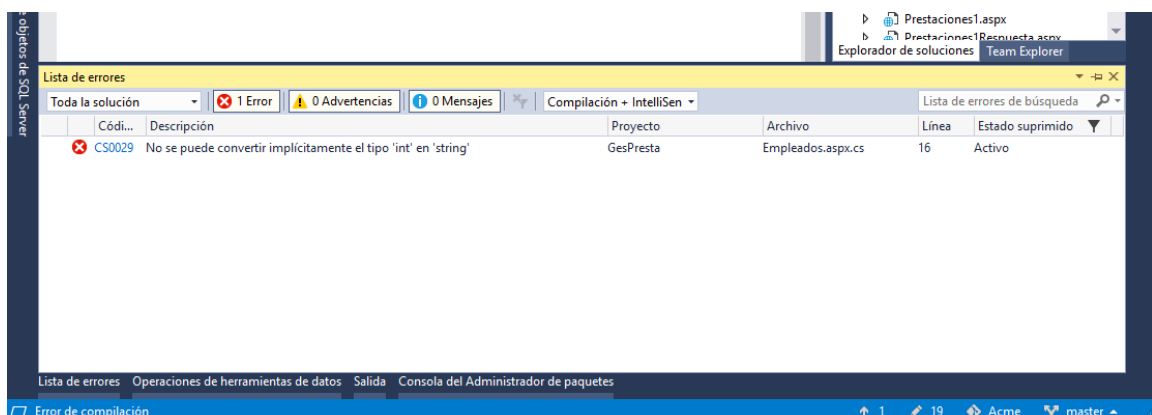
```
txtNifEmp.Text = 11111111; // Establece un valor por defecto para el campo
```

Evidentemente, en la línea de código anterior existe un error de programación, ya que la propiedad *Text* de un control solo admite valores de tipo *string*. Este error de programación producirá un error de compilación en la aplicación Web.

5. Al iniciar la depuración para ver el resultado en el explorador aparece una ventana similar a la siguiente.



6. En este caso, el depurador devuelve un error de compilación. Si se responde negativamente a la pregunta, entonces puede visualizarse la información correspondiente al error producido en la ventana **Lista de errores** de Visual Studio que mostrará un error similar al siguiente:



7. Para subsanar el error, se asignará de forma correcta el valor literal de una cadena:

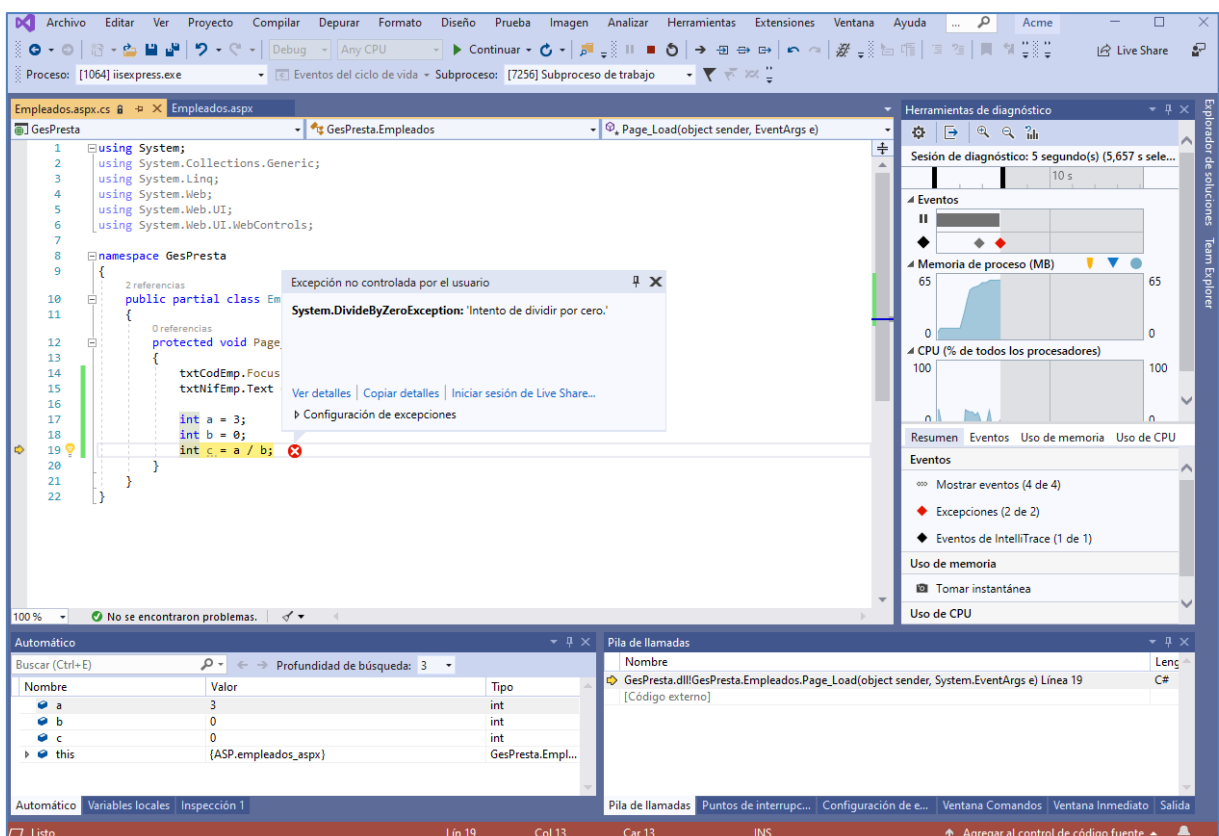
```
txtNifEmp.Text = "11111111"; // Establece un valor por defecto para el campo
```

8. Iniciar la depuración para comprobar que ya no se produce el error de compilación y que la aplicación Web funciona correctamente.
9. A continuación, se provocará un error de programación que producirá un **error en tiempo de ejecución**. Este tipo de errores producen que se detenga la ejecución de la aplicación Web de forma inmediata en el mismo momento en que se producen.
10. Añadir las siguientes líneas de código en el evento **Load** del objeto **Page**.

```
int a = 3;           // Produce un error en tiempo de ejecución
int b = 0;
int c = a / b;
```

Evidentemente, en la tercera línea del código anterior existe un error semántico, ya que el cálculo de la variable c supone una división por 0. Este error de programación producirá un error en tiempo de ejecución que detendrá la ejecución de la aplicación Web.

11. Iniciar la depuración para comprobar que se detiene la ejecución de la aplicación Web y que aparece una ventana similar al siguiente en el punto del código donde se produce el error.



12. Detener la depuración seleccionando la opción correspondiente en el menú **Depuración** o en la **Barra de herramientas** de Visual Studio.
13. A continuación, subsanar el error comentando las tres líneas de código anteriores.
14. Iniciar la depuración para comprobar que ya no se produce el error y que la aplicación Web funciona correctamente.

Ejercicio 2

Seguimiento a nivel de página

En este ejercicio se demuestra cómo habilitar y deshabilitar el seguimiento a nivel de página.

Habilitar el seguimiento en una página Web de ASP.NET

Para habilitar el seguimiento en una página Web concreta de la aplicación Web, realizar las siguientes acciones:

1. Abrir el Web Form *Empleados.aspx* del proyecto *GesPresta*. y cambiar a la vista **Código**.
2. Modificar la directiva **@ Page** agregando un atributo **Trace** y estableciendo su valor a *true*.

```
<%@Page Language="C#" AutoEventWireup="true" CodeBehind="Empleados.aspx.cs"
    Inherits="Gespresta.Empleados" Trace="true"%>
```

3. Iniciar la depuración para visualizar el resultado en el navegador. Podrá comprobarse que la información de seguimiento se muestra en la parte inferior de la propia página Web.

DATOS DE LOS EMPEADOS

Código Empleado

NIF

Apellidos y Nombre

Dirección

Ciudad

Teléfonos

Fecha de Nacimiento

Fecha de Ingreso

Sexo

☒ Hombre ☐ Mujer

Departamento

Investigación

Enviar

Detalles de la solicitud

Id. de sesión:	bc0swi45avingy55pv5b3set	Tipo de petición:	GET
Hora de la solicitud:	06/02/2007 16:05:53	Código de estado:	200
Codificación de la solicitud:	Unicode (UTF-8)	Codificación de la respuesta:	Unicode (UTF-8)

Información de seguimiento

Categoría	Mensaje	Desde los primeros	Desde los últimos
aspx.page	Begin PreInit		
aspx.page	End PreInit	0,00775373753530103	0,007754
aspx.page	Begin Init	0,00871668957588784	0,000963
aspx.page	End Init	0,0177886836778047	0,009072

De este modo, se ha habilitado la información de seguimiento solo en el Web Form *Empleados.aspx*.

Ejercicio 3

Mensajes de seguimiento

En este ejercicio se muestra cómo añadir mensajes de seguimiento personalizados a nivel de página.

Agregar mensajes de seguimiento personalizados

1. Abrir el archivo *Empleados.aspx.cs* del proyecto *GesPresta*.
2. Se va a añadir código lógico al procedimiento de evento *Page_Load*.
3. Se va a añadir una instrucción de seguimiento para que indique que se inicia el evento *Page_Load*. Una segunda instrucción de seguimiento, indicando que finaliza este mismo evento. Y finalmente, una instrucción de advertencia indicando que se establece un valor por defecto para el campo *txtNifEmp*. Para ello añadir las siguientes líneas de código resaltadas con negrita:

```
protected void Page_Load(object sender, EventArgs e)
{
    Trace.Write("Evento", "Entrando en Page_Load");

    txtCodEmp.Focus();           // Sitúa el foco en el elemento Código Empleado
    txtNifEmp.Text = "1111111X"; // Establece un valor por defecto para el campo

    Trace.Warn("Asignación", "Cambiado el valor de txtNifEmp a: " + txtNifEmp.Text);

    Trace.Write("Evento", "Saliendo de Page_Load");
}
```

4. Iniciar la depuración del Web Form *Empleados.aspx*. En la zona inferior de la visualización del Web Form se puede comprobar que se muestran los mensajes de información de seguimiento y de advertencia correspondientes a las instrucciones establecidas en el código.

aspx.page	End PreLoad	0,000326049034063929	0,000043
aspx.page	Begin Load	0,000355971809162022	0,000030
Evento	Entrando en Page_Load	0,0020479394196886	0,001692
Asignación	Cambiado el valor por defecto de txtNifEmp a: 1111111X	0,00233214705220803	0,000284
Evento	Saliendo de Page_Load	0,00237442331242346	0,000042
aspx.page	End Load	0,00241560742758603	0,000041

Ejercicio 4

Seguimiento a nivel de aplicación

En este ejercicio se estudia cómo habilitar y deshabilitar el seguimiento a nivel de aplicación.

Utilizar seguimiento a nivel de aplicación

1. En primer lugar, se deshabilitará el seguimiento a nivel de página que se ha establecido anteriormente en el Web Form *Empleados.aspx*. Para ello:
 - a. Acceder a vista de Código del Web Form.
 - b. Establecer la propiedad **Trace** al valor *false*.
 - c. Iniciar la depuración del Web Form *Empleados.aspx* y comprobar que no se muestra la información de seguimiento.
2. Desde el **Explorador de Soluciones**, abrir el archivo de configuración *Web.config* que contiene un conjunto de instrucciones declarativas para la configuración de la aplicación Web *GesPresta*. Al abrir el archivo *Web.config* se mostrará un contenido similar al siguiente:

```
<?xml version="1.0"?>
<!--
  Para obtener más información sobre cómo configurar la aplicación de ASP.NET, visite
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.8"/>
    <httpRuntime targetFramework="4.8"/>
  </system.web>
</configuration>
```

3. Añadir la siguiente línea a la sección `<system.web>` del archivo de configuración *Web.config*.

```
<trace enabled="true"/>
```

4. Iniciar la depuración del Web Form *Empleados.aspx* en el navegador. La información de seguimiento no se mostrará. Para visualizar la información de seguimiento en el navegador es necesario modificar la línea que se acaba de añadir de la siguiente forma:

```
<trace enabled="true" pageOutput="true"/>
```

5. Iniciar la depuración del Web Form *Empleados.aspx* en el navegador para comprobar que no aparece la información de seguimiento, porque está deshabilitada a nivel de página.
6. Iniciar la depuración del Web Form *Prestaciones.aspx* en el navegador para comprobar que sí aparece la información de seguimiento.
7. Para visualizar la información de seguimiento en *Empleados.aspx* es necesario abrir la vista Código del Web Form *Empleados.aspx* y eliminar el atributo de la directiva de página **Trace="false"**. Al iniciar la depuración de *Empleados.aspx* nuevamente, podrá comprobarse que una vez eliminada la directiva **Trace** aparece la información de seguimiento.

Ejercicio 5

Ejecución de código en modo depuración

En este ejercicio se mostrará cómo ejecutar código en el modo de depuración.

Utilizar la sentencia `Trace.IsEnabled`

1. Iniciar la depuración del Web Form *Empleados.aspx*. Puede comprobarse que el control *TextBox* denominado *txtNifEmp* muestra un valor predeterminado al cargarse la página. Ello es debido al código incluido anteriormente en el procedimiento de evento **Page_Load** del archivo de código subyacente *Empleados.aspx.cs*.
2. Abrir el archivo de código subyacente *Empleados.aspx.cs* para modificar el código del evento **Page_Load** en la siguiente forma:

```
protected void Page_Load(object sender, EventArgs e)
{
    Trace.Write("Evento", "Entrando en Page_Load");

    txtCodEmp.Focus();           // Sitúa el foco en el elemento Código Empleado

    if (Trace.IsEnabled)
    {
        txtNifEmp.Text = "1111111X"; // Establece un valor por defecto para el campo
        Trace.Warn("Asignación", "Cambiado el valor txtNifEmp a: " + txtNifEmp.Text);
    }

    Trace.Write("Evento", "Saliendo de Page_Load");
}
```

3. Iniciar la depuración del Web Form *Empleados.aspx* para comprobar que se muestra el valor predeterminado en el control *txtNifEmp*.
4. A continuación, para dejar de mostrar la información de seguimiento a nivel de aplicación, abrir el archivo *Web.config* para comentar la siguiente línea:

```
<!-- <trace enabled="true" pageOutput="true"/> -->
```

5. Iniciar la depuración de la aplicación Web para comprobar que se ha deshabilitado la información de seguimiento a nivel de aplicación y que, además, ya no se muestra el valor predeterminado en el control *TextBox* denominado *txtNifEmp*.

PARTE 2. VALIDACIÓN DE ENTRADAS DE USUARIO

Ejercicio 6 Validación de campo obligatorio

La compañía Acme considera que algunos de los datos que introducen los usuarios a través de Web Forms son básicos para el funcionamiento de la aplicación Web para la gestión de las prestaciones sociales. En este ejercicio se establece que ciertos campos de un Web Form sean obligatorios.

Campos obligatorios en un Web Form

A continuación, se realiza una validación de campo requerido sobre algunos controles de los Web Forms *Empleados.aspx* y *Prestaciones.aspx*, utilizando controles de validación de ASP.NET. Para ello:

1. Abrir el Web Form *Empleados.aspx*. Los campos considerados obligatorios para el Web Form *Empleados.aspx* son los siguientes: *Código empleado*, *NIF*, *Apellidos y nombre*, *Teléfonos*, *Fecha de nacimiento* y *Fecha de Ingreso*.
2. En la vista Diseño, desde la ficha **Validación** del **Cuadro de Herramientas**, arrastrar y soltar un control *RequiredFieldValidator* junto a cada uno de los campos obligatorios.
3. Cambiar el ID a cada uno de los controles *RequiredFieldValidator* incorporados del Web Form, de forma que comiencen por *rqd* y continúen con el ID del control al que validan cada uno. Por ejemplo: *rqdTxtCodEmp* para el control de validación situado junto a *txtCodEmp*.
4. A cada uno de los controles *RequiredFieldValidator* agregados al Web Form, establecer la propiedad *ControlToValidate* de forma que haga referencia al control asociado para su validación. Por ejemplo, el control *rqdtxtCodEmp* se asociará al control *txtCodEmp*.
5. Cambiar la propiedad *ErrorMessage* de cada uno de los controles *RequiredFieldValidator* para que muestre un mensaje descriptivo de validación en cada caso. Por ejemplo: "El Código Empleado es obligatorio" para el control *rqdtxtCodEmp*. Si se desea, también se puede modificar la propiedad *ForeColor* al valor *Red*.
6. Guardar el Web Form.
7. Los campos requeridos para el Web Form *Prestaciones.aspx* son: *Código Prestación*, *Importe Fijo* y *Porcentaje del Importe*. Repetir los pasos 2 al 6 con el Web Form *Prestaciones.aspx*.
8. Iniciar la depuración del Web Form *Empleados.aspx* para comprobar el funcionamiento de los controles de validación. Al hacer clic sobre el botón **Enviar** aparecerá el siguiente error:
UnobtrusiveValidationMode de WebForms requiere un ScriptResourceMapping para 'jquery'. Agregue un ScriptResourceMapping con nombre jquery (distinguir mayúsculas de minúsculas).

Para solucionarlo, es necesario desactivar la validación no intrusiva basada en jQuery en el cliente que está activada de forma predeterminada en ASP.NET a partir de .NET Framework 4.5, añadiendo el siguiente código de marcado en el archivo de configuración *Web.Config*:

```
<appSettings>
  <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
</appSettings>
```

9. Iniciar la depuración de los Web Forms *Empleados.aspx* y *Prestaciones.aspx* para comprobar que las validaciones de los campos obligatorios se realizan de forma correcta.

Ejercicio 7

Validación de rango

En este ejercicio se muestra cómo establecer la validez de un campo de un Web Form según su valor pertenezca o no a un rango de valores preestablecido.

Validación de rango en campos de un Web Form

La dirección de la compañía Acme ha decidido que el importe máximo de las prestaciones será de 500,00 € y que el porcentaje máximo de la subvención será de un 15,00 % del importe. Para establecer estas restricciones se introducirán unas validaciones de rango en los campos *Importe Fijo* y *Porcentaje del importe* del Web Form *Prestaciones.aspx*. Para ello:

1. Abrir el Web Form *Prestaciones.aspx*.
2. Desde la ficha **Validación** del **Cuadro de Herramientas**, añadir dos controles *RangeValidator* junto a los campos *Importe Fijo* y *Porcentaje del Importe*.
3. Cambiar el ID a cada uno de los *RangeValidator* de forma que comiencen por *rng* seguido del campo al que validan. Por ejemplo, *rngtxtImpPre*.
4. En cada uno de los *RangeValidator* establecer la propiedad *ControlToValidate* de forma que haga referencia a sus cuadros de texto asociados, en cada caso.
5. Establecer las propiedades *Minimum Value* y *Maximum Value* a los valores adecuados para cada control. Para el campo *Importe Fijo* entre 0,00 y 500,00. Y, para el *Porcentaje del Importe* entre 0,00 % y 15,00 %.
10. Cambiar la propiedad *ErrorMessage* de forma que muestre un mensaje descriptivo como “El valor introducido debe estar comprendido entre el 0,00 y el 15,00 %”. Si se desea, modificar la propiedad *ForeColor* al valor *Red*.
6. Establecer la propiedad *Type* a *Double*.
7. Guardar el Web Form.
8. Iniciar la depuración para cargar el Web Form *Prestaciones.aspx* y poder comprobar que las validaciones de rango establecidas se realizan de forma correcta.

Ejercicio 8

Validación de comparación

En este ejercicio se demuestra cómo realizar comparaciones entre campos del Web Form.

Validación de comparación entre campos de un Web Form

Se ha detectado que, en algunos casos, se ha podido introducir una fecha de ingreso menor que la fecha de nacimiento. Para corregir este problema se va a establecer una validación de comparación entre campos, de modo que no pueda suceder este error en futuras introducciones de datos en el Web Form de introducción de datos de empleados de la aplicación Web *GesPresta*. Para ello:

1. Abrir el Web Form *Empleados.aspx*.
2. Desde la ficha **Validación** del **Cuadro de Herramientas**, añadir un control *CompareValidator* junto al campo Fecha de Nacimiento.
3. Cambiar el ID de forma que comiencen por *cmp* seguido del nombre del control al que validan. Por ejemplo, *cmptxtFnaEmp*.
4. Establecer la propiedad *ControlToValidate* de forma que haga referencia al control asociado para la validación.
5. Establecer la propiedad *ControlToCompare* de forma que haga referencia al cuadro de texto de Fecha de Ingreso.
6. Cambiar la propiedad *ErrorMessage* de forma que muestre un mensaje descriptivo como el siguiente: “La Fecha de Ingreso del Empleado debe ser mayor que la Fecha de Nacimiento”. Si se desea, modificar la propiedad *ForeColor* al valor *Red*.
7. Establecer la propiedad *Type* a *Date*.
8. Establecer la propiedad *Operator* a *LessThan*.
9. Iniciar la depuración para cargar la página *Empleados.aspx* y comprobar que la validación de comparación de campos establecida se hace de forma correcta.

Ejercicio 9

Validación de expresiones regulares

En este ejercicio se practica cómo realizar validaciones personalizadas del contenido de los campos.

Validación de expresiones regulares en un Web Form

Algunos de los campos de los Web Forms pueden tener un formato de datos característico. Así, en el Web Form *Empleados.aspx* se podrían establecer los siguientes formatos específicos para los siguientes campos:

- Código empleado : una letra y 5 dígitos.
- NIF : 8 dígitos, un guion y una letra.
- Fecha de Nacimiento : Formato de fecha dd/mm/aaaa.
- Fecha de Ingreso : Formato de fecha dd/mm/aaaa.

Para representar las características de estos formatos particulares se utilizan las expresiones regulares, formando parte de los controles de validación correspondientes. Para ello:

1. Abrir el Web Form *Empleados.aspx*.
2. Desde la ficha **Validación** del **Cuadro de Herramientas**, añadir cuatro controles *RegularExpressionValidator* junto a los controles del Web Form sobre los que actuarán.
3. Cambiar el ID de forma que comiencen por *reg* seguido del nombre del control del campo al que validan. Por ejemplo, *regtxtCodEmp*.
4. Establecer las propiedades *ControlToValidate* de los controles *RegularExpressionValidator* añadidos al Web Form, de forma que hagan referencia a los controles asociados.
5. Cambiar la propiedad *ErrorMessage* de forma que muestre un mensaje descriptivo como "El formato de los datos a introducir debe ser: ", seguido del formato adecuado para cada caso. Si se desea, se puede modificar la propiedad *ForeColor* al valor *Green*.
6. En cada caso, empleando el formato de definición de expresiones regulares establecer la propiedad *ValidationExpression*, de la siguiente manera:
 - Código empleado : `\w\d{5}`
 - NIF : `\d{8}-\w`
 - Fecha de Nacimiento : `\d\d\/\d\d\/\d\d\d\d`
 - Fecha de Ingreso : `\d\d\/\d\d\/\d\d\d\d`
7. Iniciar la depuración para cargar el Web Form *Empleados.aspx* y comprobar que las validaciones de los campos establecidas se realizan de forma correcta.
8. Realizar el mismo proceso con el Web Form *Prestaciones.aspx*, de modo que el campo Código de Prestación presente un formato de datos consistente en: 3 dígitos, un guión, 3 dígitos, un guion y, 3 dígitos. Así, la expresión regular será:
 - Código de Prestación : `\d{3}-\d{3}-\d{3}`

Ejercicio 10

Resumen de validaciones

En este ejercicio se muestra cómo añadir un resumen de validaciones para mostrar todos los errores de validación de forma resumida al enviar el Web Form.

Incorporar un resumen de errores de validación en un Web Form

1. Abrir el Web Form *Empleados.aspx*.
2. Añadir un control *ValidationSummary* en la parte final del Web Form *Empleados.aspx*.
3. Iniciar la depuración del Web Form *Empleados.aspx* para comprobar que los mensajes de error de validación se muestran también en este control, al hacer clic sobre el botón **Enviar**.
4. Modificar el valor de la propiedad *Text* de todos los controles de validación existentes en el Web Form para incluir un valor *“*”*.
5. Iniciar nuevamente la depuración para comprobar que todos los mensajes de error de validación aparecen únicamente en el control *ValidationSummary*. Y que se muestra un carácter asterisco junto a los campos del Web Form que contienen algún error de validación.
6. Modificar la propiedad *DisplayMode* del control *ValidationSummary* al valor *List*.

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

ACME INNOVACIÓN, S. FIG.
Gestión de Prestaciones Sociales

DATOS DE LOS EMPLEADOS

Código Empleado *

NIF *

Apellidos y Nombre *

Dirección

Ciudad

Teléfonos *

Fecha de Nacimiento *

Fecha de Ingreso *

Sexo ☒ Hombre ☐ Mujer

Departamento

El Código del Empleado es obligatorio
El Nif del Empleado es obligatorio
El Nombre del Empleado es obligatorio
Al menos un Teléfono del Empleado es obligatorio
La Fecha de Nacimiento del Empleado es obligatorio
La Fecha de Ingreso del Empleado es obligatorio

7. Como se puede comprobar, cuando se producen errores de validación en un Web Form no es posible acceder a otro Web Form a través de las opciones de menú, hasta que se resuelvan estos errores. Para que sea posible acceder a otro Web Form, es necesario asignar el valor **False** a la propiedad **CausesValidation** en los controles *LinkButton* que definen los enlaces de las opciones del menú, y que están especificados en el control de usuario *Cabecera.ascx*.
8. Iniciar la depuración para comprobar el funcionamiento correcto de la aplicación Web.

PARTE 3. CREACIÓN DE CONTROLES DE USUARIO

Ejercicio 11 Creación de un control de usuario

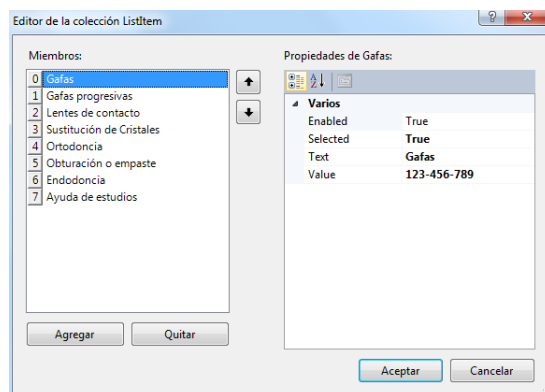
En este ejercicio se creará un control de usuario para seleccionar los datos de las prestaciones desde una lista de valores concretos.

Creación e intercambio de información con un Control de Usuario

Dado que es necesario seleccionar los datos de las prestaciones sociales que la compañía Acme ofrece a sus empleados desde una lista de valores, se va a crear un Control de usuario que encapsule este comportamiento para evitar errores e inconsistencias en la información y facilitar la operativa de la aplicación Web. Este ejercicio permite mostrar cómo crear un control de usuario y cómo realizar el intercambio de datos entre el control de usuario y el Web Form que lo incluye. Para ello:

1. Crear un nuevo Control de usuario denominado *prestacionesBuscar.ascx*. Este Control de usuario facilitará la búsqueda de prestaciones existentes mediante código y descripción. Para ello, realizar las siguientes acciones:
 - a. Hacer clic con el botón derecho en el proyecto *GesPresta*, seleccionar la opción **Agregar** y después la opción **Nuevo elemento...**
 - b. En la ventana **Agregar nuevo elemento** desplegar las opciones **Instalado**, **Visual C#** y **Web** el cuadro de la izquierda y, a continuación, seleccionar la opción **Web Forms**. En el cuadro central, seleccionar la plantilla de archivo **Control de usuario de Web Forms** y escribir *prestacionesBuscar.ascx* en el cuadro de texto **Nombre**.
 - c. Finalmente, hacer clic en **Agregar**.
2. Abrir el control de usuario *prestacionesBuscar.ascx*.
3. Agregar un control *ListBox* desde el **Cuadro de herramientas**, y hacer:
 - a. Modificar la propiedad ID del *ListBox* para asignarle el valor *lstPrestaciones*.
 - b. Editar los elementos del *ListBox* insertado en el Control de usuario accediendo a **las Tareas del ListBox**. Para acceder a las **Tareas del ListBox**, hacer clic sobre la flecha que aparece en la esquina superior derecha del control. A continuación, seleccionar la opción **Editar elementos...** para agregar los siguientes elementos:

Text	Value	Selected
Gafas	123-456-789	True
Gafas progresivas	123-557-209	
Lentes de contacto	123-787-001	
Sustitución de Cristales	123-674-650	
Ortodoncia	157-010-338	
Obturación o empaste	157-778-023	
Endodoncia	157-114-995	
Ayuda de estudios	012-783-146	



4. Abrir el archivo *prestacionesBuscar.ascx.cs* de código subyacente del control de usuario creado y añadir las líneas de código que aparecen a continuación. Este código facilita el intercambio de información entre el control de usuario y el Web Form a través del descriptor de acceso *get* de las propiedades Código y Descripción del control de usuario. Este código debe situarse contenido en la clase *prestacionesBuscar* pero fuera de cualquier procedimiento de evento, puesto que formará parte del código asociado a ningún evento.

```
public String Codigo
{
    get
    {
        return lstPrestaciones.SelectedValue;
    }
}

public String Descripcion
{
    get
    {
        return lstPrestaciones.SelectedItem.Text;
    }
}
```

5. Guardar *prestacionesBuscar.ascx* y *prestacionesBuscar.ascx.cs*
6. Para no modificar demasiado el trabajo realizado en prácticas anteriores, se recomienda realizar una copia del Web Form *Prestaciones.aspx* denominada *Prestaciones2.aspx*, tal como se realizó en prácticas anteriores.
7. Abrir el Web Form *Prestaciones2.aspx* en modo **Diseño** y, desde el **Explorador de soluciones**, arrastrar el control de usuario *prestacionesBuscar.ascx* sobre el Web Form hasta una posición adecuada. Si es necesario modificar la composición de la apariencia del Web Form para que ésta se acomode a la inserción del control de usuario.
8. Modificar la propiedad *Visible* de la instancia del control de usuario insertado en el Web Form *Prestaciones2.aspx* para asignarle el valor *False*. Es necesario conocer el valor de la propiedad *Id* de la instancia del control de usuario *prestacionesBuscar.ascx* insertado en el Web Form, probablemente, el *Id* asignado de manera predeterminada haya sido *prestacionesBuscar1*.
9. En el Web Form *Prestaciones2.aspx*, desde el **Cuadro de herramientas** añadir un control *Button* cuya propiedad *Id* tenga el valor *btnSeleccionar* debajo de la instancia del Control de usuario *prestacionesBuscar1* y ajustar sus propiedades:

Propiedad	Valor
Text	Seleccionar
CausesValidation	False
Visible	False

10. En el Web Form *Prestaciones2.aspx*, añadir un control *Button* cuyo *Id* sea *btnVerPrestaciones* a la derecha del campo *Código Prestación* y ajustar sus propiedades:

Propiedad	Valor
Text	Ver prestaciones
CausesValidation	False

11. Introducir el siguiente código asociado al evento Clic del botón *btnVerPrestaciones*.

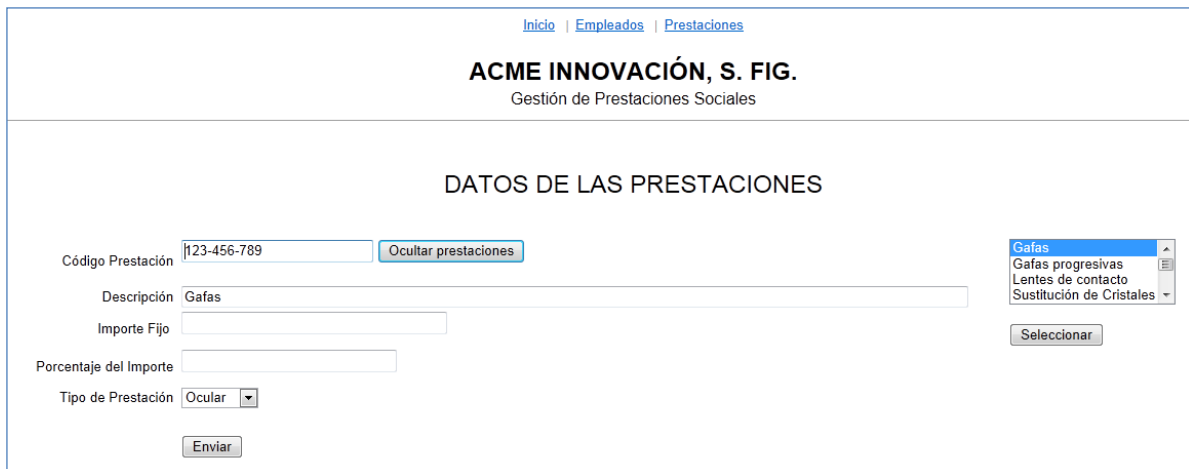
```
protected void btnVerPrestaciones_Click(object sender, EventArgs e)
{
    if (btnSeleccionar.Visible == false)
    {
        btnSeleccionar.Visible = true;
        prestacionesBuscar1.Visible = true;
        btnVerPrestaciones.Text = "Ocultar prestaciones";
    }
    else
    {
        btnSeleccionar.Visible = false;
        prestacionesBuscar1.Visible = false;
        btnVerPrestaciones.Text = "Ver prestaciones";
    }
}
```

Debe tenerse en cuenta que el código anterior es válido siempre que el *Id* de la instancia del Control de usuario insertado en el Web Form sea *prestacionesBuscar1*. Si el *Id* de la instancia del Control de usuario tiene un valor distinto deberá adecuarse el código.

12. Introducir el siguiente código asociado al evento *Clic* del botón *btnSeleccionar*. Este código captura el valor de las propiedades del código y la descripción de la prestación seleccionada en el *ListBox* del control de usuario. Estas propiedades establecen el mecanismo de intercambio de información entre el control de usuario y el Web Form.

```
protected void btnSeleccionar_Click(object sender, EventArgs e)
{
    txtCodPre.Text = prestacionesBuscar1.Codigo;
    txtDesPre.Text = prestacionesBuscar1.Descripcion;
}
```

13. Iniciar la depuración para comprobar que es posible mostrar y ocultar los controles que permiten seleccionar un código de prestación. Y que al hacer clic sobre el botón **Seleccionar**, los valores de la prestación seleccionada se transfieren a los cuadros de texto del Web Form.



Inicio | Empleados | Prestaciones

ACME INNOVACIÓN, S. FIG.
Gestión de Prestaciones Sociales

DATOS DE LAS PRESTACIONES

Código Prestación: 123-456-789 [Ocultar prestaciones]

Descripción: Gafas

Importe Fijo: []

Porcentaje del Importe: []

Tipo de Prestación: Ocular [v]

[Enviar]

Seleccionar

Gafas
Gafas progresivas
Lentes de contacto
Sustitución de Cristales

PARTE 4. UTILIZACIÓN DE PÁGINAS MAESTRAS (*MASTER PAGES*)

En los ejercicios siguientes de esta parte de la práctica, se creará nuevamente el Web Form de inicio de la aplicación Web, así como, los Web Forms de introducción de los datos de empleados y prestaciones. El comportamiento de los nuevos Web Forms a crear, desde el punto de vista funcional, será similar a los ya creados, sin embargo, se empleará una página maestra o *Master Page* para establecer una presentación visual común. Para evitar que las nuevas páginas Web se sobrescriban sobre las ya realizadas y se puedan producir pérdidas del trabajo realizado, se asignarán nombres diferentes a las nuevas páginas web de ASP.NET que se van a crear a continuación.

Ejercicio 12

Crear el Web Form: DefaultMP.aspx empleando una página maestra

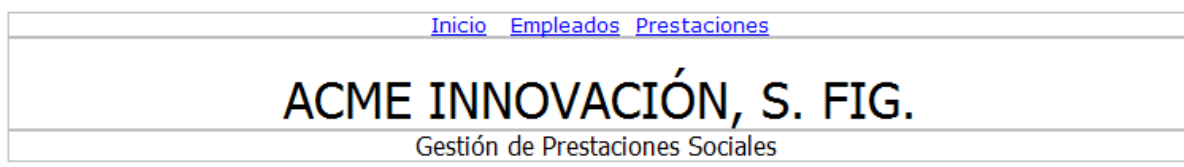
En primer lugar, se creará una Página Maestra o *Master Page* que facilitará la creación de una presentación visual común y coherente para todas las páginas de la aplicación Web. Las *Master Pages* de ASP.NET permiten definir el diseño visual y el comportamiento que se desea tengan todas las páginas, o un grupo de páginas, de una aplicación Web. Una vez creadas las páginas de contenido individual que incluirán el contenido específico que se desea mostrar en cada página, al ser solicitadas estas se combinan con la página maestra asociada para generar una interfaz que fusiona el diseño visual de la página maestra con el de la página de contenido. En la aplicación Web *GesPresta*, la página maestra contendrá la cabecera común que incluye los enlaces de acceso a las diferentes páginas de ASP.NET que conforman los procesos de la aplicación Web.

Crear y diseñar una Página Maestra o *Master Page*

Una página maestra es un archivo de ASP.NET con extensión de nombre .master que establece un diseño predefinido y que puede incluir texto estático, elementos HTML y controles de servidor. Una página maestra incluye todos los elementos de nivel superior de página, como las etiquetas html, head, body y form, así como la directiva @ Master que define sus características. Además, una página maestra también incluye uno o varios controles *ContentPlaceHolder* que definen las regiones que incluirán contenido reemplazable. Este contenido reemplazable se define en las páginas de contenido. Una página maestra también tiene asociado un archivo de código subyacente para poder incluir el código lógico que define su comportamiento y que afectará a todas las páginas de contenido con las que se combina. A continuación, se va a crear una página maestra para la aplicación Web *GesPresta*:

1. Para crear una Página Maestra en la aplicación Web *GesPresta*, hacer:
 - a. En el **Explorador de soluciones**, hacer clic con el botón derecho en el proyecto *GesPresta*, seleccionar la opción **Agregar** y después la opción **Nuevo elemento...**
 - b. En la ventana **Agregar nuevo elemento** desplegar las opciones **Instalado**, **Visual C#** y **Web** el cuadro de la izquierda y, a continuación, seleccionar la opción **Web Forms**. En el cuadro central, seleccionar la plantilla de archivo **Web Forms con página maestra** y escribir *MasterPage.master* en el cuadro de texto **Nombre**.
 - c. Finalmente, hacer clic en **Agregar**.

- Desde el **Explorador de soluciones**, abrir la página maestra *MasterPage.master*. En la vista Diseño o Código, agregar los controles necesarios desde el **Cuadro de Herramientas** hasta conseguir un aspecto similar al siguiente. Los controles que se van a agregar a la página maestra deben situarse siempre fuera del control **ContentPlaceHolder**.



El aspecto, el contenido y el comportamiento que se define en una página maestra se refiere a las regiones de código no reemplazable, esto es, a los textos estáticos, los elementos HTML y los controles de servidor que se sitúan fuera de los controles *ContentPlaceHolder* de la página maestra y que conforman la presentación común de las páginas de ASP.NET asociadas.

Crear un Web Form asociado a una página maestra

Para crear un nuevo Web Form denominado *MPDefault.aspx* asociado a la página maestra *MasterPage.master* y que actuará como página de inicio de la aplicación Web, hacer:

- En el **Explorador de soluciones**, hacer clic con el botón derecho en el proyecto *GesPresta*, seleccionar la opción **Agregar** y después la opción **Nuevo elemento...**
- En la ventana **Agregar nuevo elemento** desplegar las opciones **Instalado**, **Visual C#** y **Web** el cuadro de la izquierda y, a continuación, seleccionar la opción **Web Forms**. En el cuadro central, seleccionar la plantilla de archivo **Formulario Web con página maestra** y escribir *MPDefault.aspx* en el cuadro de texto **Nombre**.
- Hacer clic en **Agregar**.
- En la ventana **Agregar una página maestra**, seleccionar la página maestra creada anteriormente y que se denomina *MasterPage.master* en el cuadro de la derecha.
- Hacer clic en **Aceptar**.

Se habrá creado el Web Form *MPDefault.aspx* que actuará como página de contenido. La página maestra *MasterPage.master* se combinará con el Web Form al solicitar la página de contenido *MPDefault.aspx*. Como puede comprobarse en la vista **Código**, el Web Form *MPDefault.aspx* contiene una directiva **@ Page** que enlaza la página de contenido actual con la página maestra seleccionada.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
    CodeBehind="MPDefault.aspx.cs" Inherits="GesPresta.MPDefault" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

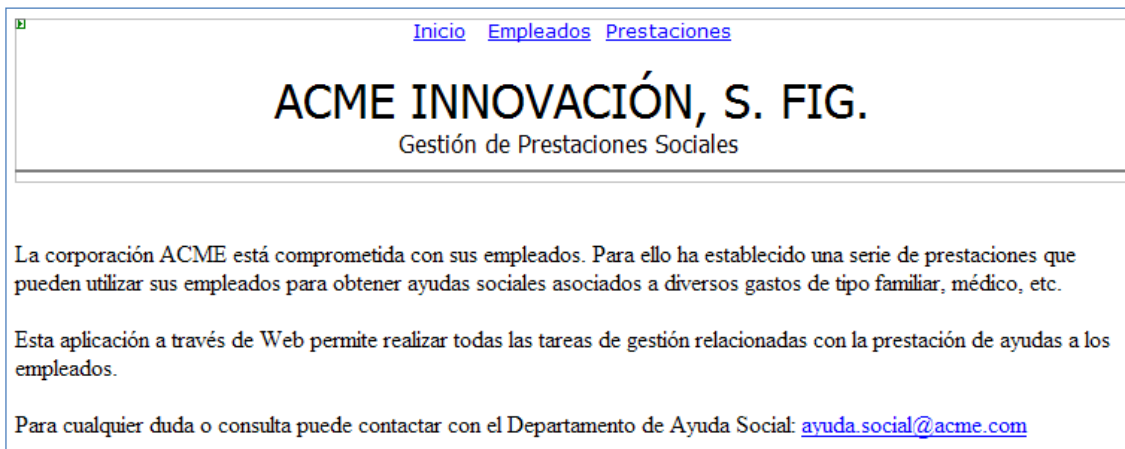
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

</asp:Content>
```

Además, puede apreciarse que la página *MPDefault.aspx* incluye controles *Content* asignados a los controles *ContentPlaceHolder* de la página maestra, mediante la propiedad *ContentPlaceHolderID*. Así, el contenido específico o individual del Web Form *MPDefault.aspx*, que actúa como página de contenido, debe ser definido dentro de los controles *Content*. Dentro de las etiquetas de marcado que establecen un control *Content* asignado a un control *ContentPlaceHolder* de una página maestra se incluirá el texto estático, los elementos HTML y los controles de servidor que definen el contenido específico de esa página. En una página de contenido, todo lo que esté fuera de los controles *Content* producirá un error. Cabe destacar que las páginas de contenido no incluyen los elementos de nivel superior de página, como las etiquetas *html*, *head*, *body* y *form*. Es la página maestra la que los incluye y la que se combinará con las páginas de contenido con las que se combine.

Colocar controles en un formulario Web Form (página de contenido)

1. En la vista de **Diseño** o de **Código** de Web Form *MPDefault.aspx*, agregar las etiquetas, la información y los controles necesarios desde el **Cuadro de herramientas**, hasta obtener un aspecto similar al de la siguiente ilustración. El contenido a agregar al Web Form debe situarse dentro del control **Content** cuyo *ContentPlaceHolderId* es **ContentPlaceHolder1** para poder conseguir que la combinación con la página maestra sea efectiva. Las características de estilo propias del Web Form pueden situarse en el *Content* cuyo *ContentPlaceHolderID* es **head**.



The screenshot shows a web form with a navigation bar at the top containing links: [Inicio](#), [Empleados](#), and [Prestaciones](#). Below the navigation bar is the company name **ACME INNOVACIÓN, S. FIG.** and the subtitle **Gestión de Prestaciones Sociales**. The main content area contains the following text:

La corporación ACME está comprometida con sus empleados. Para ello ha establecido una serie de prestaciones que pueden utilizar sus empleados para obtener ayudas sociales asociados a diversos gastos de tipo familiar, médico, etc.

Esta aplicación a través de Web permite realizar todas las tareas de gestión relacionadas con la prestación de ayudas a los empleados.

Para cualquier duda o consulta puede contactar con el Departamento de Ayuda Social: ayuda.social@acme.com

2. Iniciar la depuración para comprobar el funcionamiento del Web Form *MPDefault.aspx*.

Ejercicio 13

Crear el Web Form MPEmpleados.aspx empleando una página maestra

En este ejercicio, se creará un nuevo Web Form denominado *MPEmpleados.aspx* basado en una Página Maestra que permitirá introducir la información relativa a los empleados en la aplicación Web.

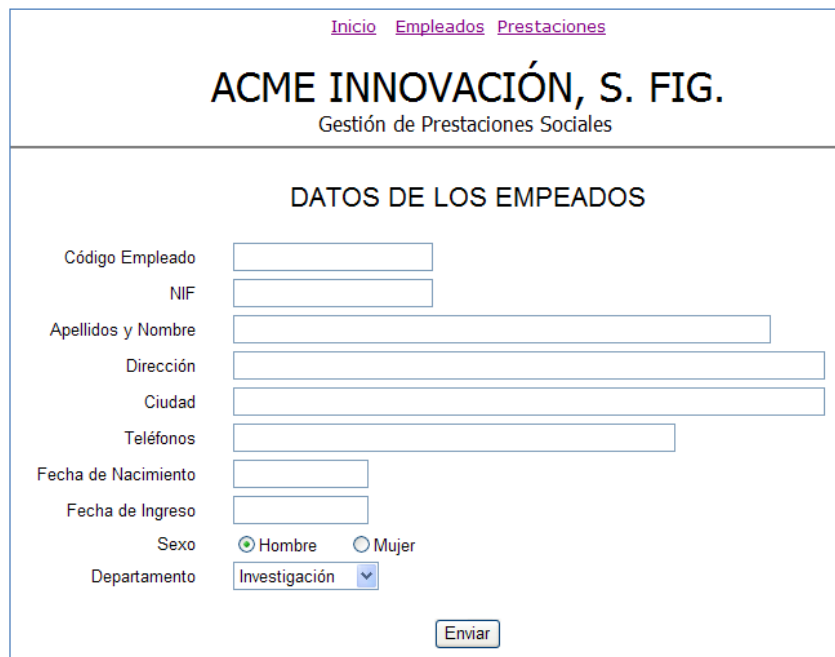
Crear un formulario Web Form

Para crear un Web Form denominado *MPEmpleados.aspx* basado en una Página Maestra, hacer:

1. En el **Explorador de soluciones**, hacer clic con el botón derecho en el proyecto *GesPresta*, seleccionar la opción **Agregar** y después la opción **Nuevo elemento...**
2. En la ventana **Agregar nuevo elemento** desplegar las opciones **Instalado**, **Visual C#** y **Web** el cuadro de la izquierda y, a continuación, seleccionar la opción **Web Forms**. En el cuadro central, seleccionar la plantilla de archivo **Formulario Web con página maestra** y escribir *MPEmpleados.aspx* en el cuadro de texto **Nombre** y hacer clic en **Agregar**.
3. En la ventana **Agregar una página maestra**, seleccionar la página maestra deseada.
4. Hacer clic en **Aceptar**.

Colocar controles en un formulario Web Form

1. En el Web Form *MPEmpleados.aspx*, agregar los controles necesarios hasta obtener un aspecto similar al de la siguiente ilustración. Los controles que se van a agregar deben situarse dentro del **Content** cuyo ContentPlaceHolderId es **ContentPlaceHolder1**. Las características de estilo del Web Form pueden situarse en el Content cuyo ContentPlaceHolderID es **head**.



2. Modificar las propiedades correspondientes de los controles insertados en el Web Form, tal como se indica en la siguiente tabla:

Control	ID	Text	Items	Value	Selected	Repeat Direction
TextBox	txtCodEmp					
TextBox	txtNifEmp					
TextBox	txtNomEmp					
TextBox	txtDirEmp					
TextBox	txtCiuEmp					
TextBox	txtTelEmp					
TextBox	txtFnaEmp					
TextBox	txtFinEmp					
RadioButtonList	rblSexEmp		Hombre Mujer	H M	True	Horizontal
DropDownList	ddlDepEmp		Investigación Desarrollo Innovación Administración	Investigación Desarrollo Innovación Administración	True	
Button	cmdEnviar	Enviar				

3. Iniciar la depuración de la página *MPEmpleados.aspx* para comprobar su funcionamiento.
4. Modificar la propiedad **PostBackUrl** de los controles **LinkButton** que aparecen en la página maestra *MasterPage.master* para poder acceder a los Web Forms *MPDefault.aspx* y *MPEmpleados.aspx* haciendo clic en los enlaces Inicio y Empleados, respectivamente.
5. Iniciar la depuración del Web Form para comprobar su funcionamiento.

Ejercicio 14

Crear el Web Form MPPrestaciones.aspx empleando una página maestra

En este ejercicio, se creará un nuevo Web Form denominado *MPPrestaciones.aspx* basado en una página maestra que permitirá introducir la información relativa a las prestaciones sociales.

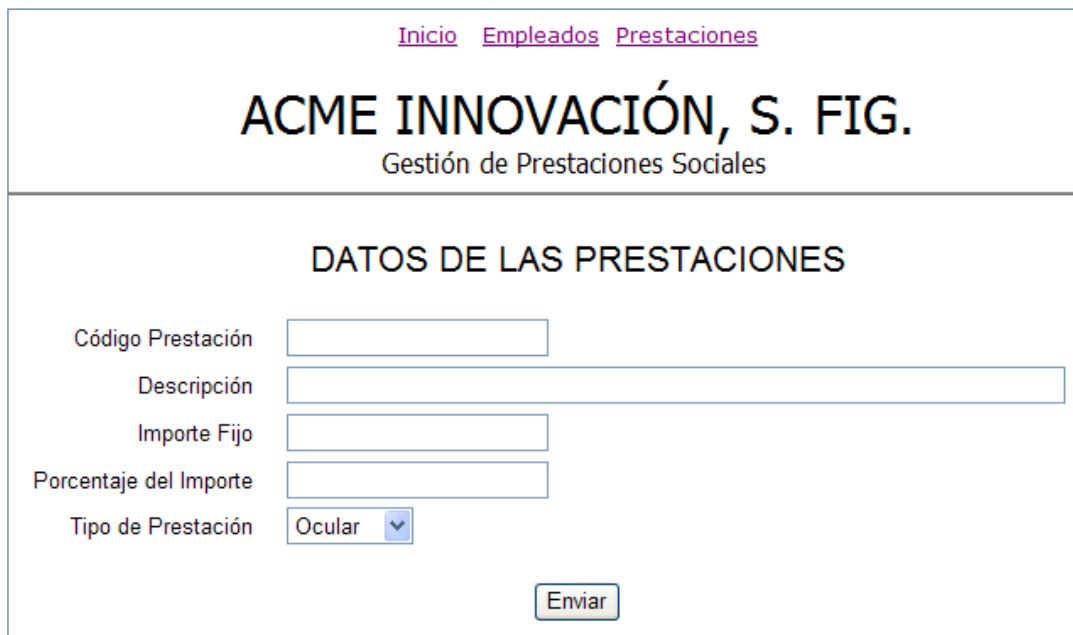
Crear un formulario Web Form

Para crear un Web Form denominado *MPPrestaciones.aspx* basado en una Página Maestra, hacer:

1. En el **Explorador de soluciones**, hacer clic con el botón derecho en el proyecto *GesPresta*, seleccionar la opción **Agregar** y después la opción **Nuevo elemento...**
2. En la ventana **Agregar nuevo elemento** desplegar las opciones **Instalado**, **Visual C#** y **Web** el cuadro de la izquierda y, a continuación, seleccionar la opción **Web Forms**. En el cuadro central, seleccionar la plantilla de archivo **Formulario Web con página maestra** y escribir *MPPrestaciones.aspx* en el cuadro de texto **Nombre** y hacer clic en **Agregar**.
3. En la ventana **Agregar una página maestra**, seleccionar la página maestra deseada.
4. Hacer clic en **Aceptar**.

Colocar controles en un formulario Web Form

1. En el Web Form *MPPrestaciones.aspx*, agregar los controles necesarios hasta obtener un aspecto similar al de la siguiente ilustración. Los controles que se van a agregar deben situarse dentro del **Content** cuyo ContentPlaceHolderId es **ContentPlaceHolder1**. Las características de estilo del Web Form pueden situarse en el Content cuyo ContentPlaceHolderID es **head**.



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

ACME INNOVACIÓN, S. FIG.

Gestión de Prestaciones Sociales

DATOS DE LAS PRESTACIONES

Código Prestación	<input type="text"/>
Descripción	<input type="text"/>
Importe Fijo	<input type="text"/>
Porcentaje del Importe	<input type="text"/>
Tipo de Prestación	Ocular <input type="button" value="v"/>

2. Modificar las propiedades correspondientes de los controles insertados, tal como se indica a continuación:

Control	ID	Text	Items	Value	Selected
TextBox	txtCodPre				
TextBox	txtDesPre				
TextBox	txtImpPre				
TextBox	txtPorPre				
DropDownList	ddlTipPre		Dentaria	Dentaria	
			Familiar	Familiar	
			Ocular	Ocular	True
			Otras	Otras	
Button	cmdEnviar	Enviar			

3. Modificar la propiedad **PostBackUrl** de los controles **LinkButton** que aparecen en la página maestra *MasterPage.master* para poder acceder al Web Form *MPPrestaciones.aspx* haciendo clic en el enlace correspondiente.
4. Iniciar la depuración del Web Form para comprobar su funcionamiento.