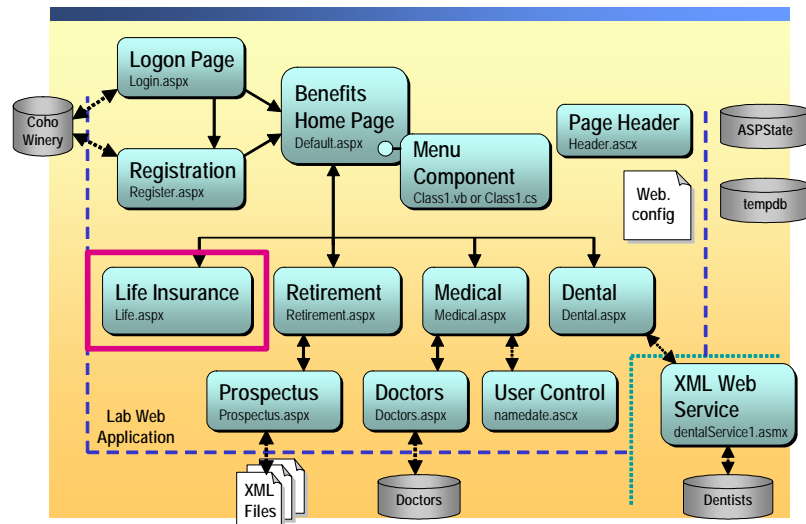


Práctica A: Validar la entrada de datos del usuario



Objetivos

En este laboratorio, aprenderemos a:

Utilizar controles de validación para validar valores de cuadros de texto en un formulario Web Form con Microsoft® ASP.NET.

Utilizar un control de validación **ValidationSummary** para resumir errores de validación en un formulario Web Form.

Nota Este laboratorio se centra en los conceptos de este módulo. Por ello, podría no cumplir las recomendaciones de seguridad de Microsoft.

Requisitos previos

Para trabajar en este laboratorio, es necesario tener:

Conocimientos sobre cómo agregar controles Web a un formulario Web Form ASP.NET.

Conocimientos sobre cómo escribir una función Microsoft Visual Basic® Scripting Edition (VBScript) en el lado del cliente.

Conocimientos sobre cómo crear un procedimiento de evento para un control Web.

Escenario

Coho Winery ofrece varios beneficios a sus empleados. En los laboratorios de este curso, crearemos un sitio Web que permita a los empleados seleccionar y configurar los beneficios que elijan.

La aplicación Web Benefits tiene un formulario Web Form denominado life.aspx. Esta página se utiliza para el beneficio de seguro de vida, y solicita Nombre, Fecha de nacimiento e Importe de cobertura al usuario. En este laboratorio, agregaremos controles de validación para validar las entradas en estos campos de texto. También agregaremos un control de validación resumen para resumir las entradas no válidas en la página.

Tiempo estimado para completar el laboratorio:
15 minutos

Ejercicio 0

Configuración del laboratorio

Para realizar este laboratorio, es necesario haber creado el proyecto de la aplicación Web *Benefits* y el proyecto de la biblioteca de clases *BenefitsList*. Estos proyectos pueden haberse creado utilizando Visual Basic .NET o Microsoft Visual C#™ .NET.

Si no se han creado estos proyectos, realizar los siguientes pasos.

Crear la solución LabApplication

Importante Realizar este procedimiento únicamente si no se ha creado un archivo de solución LabApplication.

Utilizando Visual Studio .NET, crear una nueva solución en blanco denominada **LabApplication**:

En el menú **Archivo**, seleccionar **Nuevo**, y hacer clic en **Solución en blanco**.

En el cuadro de diálogo **Nuevo proyecto**, escribir **LabApplication** en el cuadro de texto **Nombre** y hacer clic en **Aceptar**.

Crear el proyecto Benefits

Importante Realizar este procedimiento únicamente si no se ha creado anteriormente un proyecto Benefits, o si se ha eliminado.

Crear un nuevo proyecto de aplicación Web ASP.NET, denominado **BenefitsVB** o **BenefitsCS**, en la solución LabApplication:

En el menú **Archivo**, seleccionar **Nuevo**, y hacer clic en **Proyecto**.

En el cuadro de diálogo **Nuevo proyecto**, en la lista **Tipos de proyecto**, hacer clic en **Proyectos de Visual Basic** o **Proyectos de Visual C#**.

En la lista **Plantillas**, hacer clic en **Aplicación Web ASP.NET**, establecer la **Ubicación** en **http://localhost/BenefitsVB** para un proyecto Visual Basic .NET o en **http://localhost/BenefitsCS** para un proyecto Visual C#.

Hacer clic en **Agregar a Solución**, y hacer clic en **Aceptar**.

Precaución Cuando se agregan proyectos a la solución, es importante el uso de mayúsculas en el nombre del proyecto. Probablemente, utilizaremos algunos formularios Web Forms ya generados en éste y en otros laboratorios, por ello, debemos verificar que hemos escrito con mayúsculas el nombre del proyecto Benefits.

Actualizar el proyecto Benefits

1. En Visual Studio .NET, abrir el archivo de solución LabApplication.
2. En el Explorador de soluciones, hacer clic con el botón derecho en **BenefitsVB** o **BenefitsCS**, seleccionar **Agregar**, y hacer clic en **Agregar elemento existente**.

Para el proyecto Visual Basic .NET

Ir a la carpeta VB\Starter\BenefitsVB

Para el proyecto Visual C#

Ir a la carpeta CS\Starter\BenefitsCS

Estas carpetas pueden encontrarse dentro del fichero labs06.zip

1. En el cuadro **Archivos de tipo** del cuadro de diálogo **Agregar elemento existente – Benefits**, hacer clic en **Todos los archivos (*.*)**.
2. Seleccionar todos los archivos de la carpeta y hacer clic en **Abrir**.
3. Hacer clic en **Sí** a la pregunta de si sobrescribir o recargar los archivos.

Crear la biblioteca de clases BenefitsList

Importante Realizar estos pasos únicamente si no se ha creado anteriormente un proyecto BenefitsList, o si se ha eliminado.

1. Crear un proyecto de biblioteca de clases.

Para el proyecto Visual Basic .NET

Crear un nuevo proyecto de biblioteca de clases Visual Basic .NET con el nombre **BenefitsListVB** y añadirlo a la solución LabApplication:

En el menú **Archivo**, seleccionar **Nuevo**, y hacer clic en **Proyecto**.

En el cuadro de diálogo **Nuevo proyecto**, en la lista **Tipos de proyecto**, hacer clic en **Proyectos de Visual Basic**.

En la lista **Plantillas**, hacer clic en **Biblioteca de clases**, establecer el **Nombre** de **BenefitsListVB**.

Hacer clic en **Agregar a solución**, y clic en **Aceptar**.

Para el proyecto Visual C#

Crear un nuevo proyecto de biblioteca de clases Visual C# .NET con el nombre **BenefitsListCS** y añadirlo a la solución LabApplication:

- a. En el menú **Archivo**, seleccionar **Nuevo**, y hacer clic en **Proyecto**.
- b. En el cuadro de diálogo **Nuevo proyecto**, en la lista **Tipos de proyecto**, hacer clic en **Proyectos de Visual C#**.
- c. En la lista **Plantillas**, hacer clic en **Biblioteca de clases**, establecer el **Nombre** de **BenefitsListCS**.
- d. Hacer clic en **Agregar a solución**, y clic en **Aceptar**.

Importante Verificar que se han escrito correctamente las mayúsculas del proyecto BenefitsList.

Actualizar el proyecto BenefitsList

1. En Visual Studio .NET, abrir el archivo de solución LabApplication.
2. En el Explorador de soluciones, hacer clic con el botón derecho en **BenefitsListVB** o **BenefitsListCS**, seleccionar **Agregar** y hacer clic en **Agregar elemento existente**.
3. Ir al proyecto BenefitsList.

Para el proyecto Visual Basic .NET

Ir a la carpeta VB\Starter\BenefitsListVB.

Para el proyecto Visual C#

Ir a la carpeta CS\Starter\BenefitsListCS.

Estas carpetas pueden encontrarse dentro del fichero labs06.zip

4. En el cuadro **Archivos de tipo** del cuadro de diálogo **Agregar elemento existente – BenefitsList**, hacer clic en **Todos los archivos (*.*)**.
5. Seleccionar todos los archivos de la carpeta y hacer clic en **Abrir**.
6. Hacer clic en **Sí** a la pregunta de si sobrescribir o recargar los archivos.

Crear una referencia al componente BenefitsList del proyecto Benefits

1. En el proyecto **Benefits** de la solución LabApplication, completar los siguientes pasos para agregar una referencia al componente **BenefitsList** que acabamos de crear:
 - a. Hacer clic con el botón derecho en el proyecto **BenefitsVB** o **BenefitsCS** en el Explorador de soluciones y hacer clic en **Agregar referencia**.
 - b. En el cuadro de diálogo **Agregar referencia**, en la pestaña **Proyectos**, hacer doble clic en el proyecto **BenefitsListVB** o **BenefitsListCS**.
 - c. En la lista **Componentes seleccionados**, seleccionar el componente **BenefitsListVB** o **BenefitsListCS**, y hacer clic en **Aceptar**.

El componente se agrega a la carpeta Referencias en el Explorador de soluciones.

Ejercicio 1

Uso de los controles RequiredFieldValidator

En este ejercicio, agregaremos controles **RequiredFieldValidator** para los cuadros de texto **Name**, **Birth Date** y **Coverage** del formulario Web Form ASP.NET life.aspx.

Agregar controles RequiredFieldValidator

1. Abrir la página life.aspx en el proyecto Benefits de la solución LabApplication.
2. Mediante una operación de arrastrar y soltar, ubicar tres controles **RequiredFieldValidator** en el formulario Web Form de forma que el formulario tenga un aspecto similar a la siguiente ilustración.

Diagrama de un formulario web con tres campos de texto y sus respectivos validadores RequiredFieldValidator. El formulario está sobre una cuadrícula de puntos. Los campos son: 'Name:', 'Birth Date:', y 'Coverage:'. Cada campo tiene un cuadro de texto adyacente. A la derecha de cada cuadro de texto hay un control RequiredFieldValidator, representado por un icono de un cuadrado con una 'P' y el texto 'RequiredFieldValidator' en rojo.

3. Establecer las propiedades de los tres controles **RequiredFieldValidator** utilizando los valores que se muestran en la siguiente tabla.

ID	Mensaje de error	Texto	ControlToValidate
vldName	El Nombre no puede dejarse en blanco	*	txtName
vldBirth	La Fecha de nacimiento no puede dejarse en blanco	*	txtBirth
vldCoverage	La cobertura no puede dejarse en blanco	*	txtCoverage

Cuando cambiamos la propiedad **Text** de los controles, cambiamos el texto que aparece en la vista Diseño. La página debería tener ahora un aspecto similar a la siguiente ilustración.

Diagrama de un formulario web con tres campos de texto y sus respectivos validadores RequiredFieldValidator. El formulario está sobre una cuadrícula de puntos. Los campos son: 'Name:', 'Birth Date:', y 'Coverage:'. Cada campo tiene un cuadro de texto adyacente. A la derecha de cada cuadro de texto hay un control RequiredFieldValidator, representado por un icono de un cuadrado con una 'P' y un asterisco rojo (*).

4. Generar y examinar la página life.aspx.
5. En el explorador, dejar los campos **Name**, **Birth Date** y **Coverage** en blanco y hacer clic en **Guardar**.

¿Qué ocurre y por qué?

6. En el explorador, introducir los valores para los campos **Name**, **Birth Date** y **Coverage** y hacer clic en **Guardar**.

¿Qué ocurre y por qué?

Ejercicio 2

Uso del control ValidationSummary

En este ejercicio, utilizaremos un control **ValidationSummary** para hacer un resumen de los errores de validación de la página life.aspx.

Agregar un control ValidationSummary a la página life.aspx

1. Mediante una operación de arrastrar y soltar, ubicar un control **ValidationSummary** en la parte inferior del formulario Web Form life.aspx y establecer su propiedad **ID** en **vldSummary**.
2. Establecer la propiedad **HeaderText** del control **vldSummary** en el valor *These errors were found*

El aspecto de la página debería ser similar al de la siguiente ilustración.

The screenshot shows a web form on a grid background. It contains three text input fields labeled 'Name:', 'Birth Date:', and 'Coverage:'. Each field has a red asterisk to its right. Below these fields are two checkboxes: 'Short-term disability' and 'Long-term disability'. At the bottom left is a 'Save' button. To the right of the button, the text 'These errors were found:' is displayed in red. Below this text are two red bullet points: 'Error message 1.' and 'Error message 2.'

3. Generar y examinar la página life.aspx.
4. En el explorador, dejar en blanco los campos **Name**, **Birth Date** y **Coverage** y hacer clic en **Guardar**.

Además del asterisco rojo que aparece junto a las entradas no válidas, deberíamos ver un resumen de errores en la parte inferior de la página.

5. ¿De dónde proviene el resumen de mensajes de error?

Probar si la página es válida

1. Agregar un control **Label** en la parte inferior del formulario Web Form, establecer la propiedad **ID** del control en **lblMessage** y borrar el valor predeterminado de su propiedad **Text**.

El aspecto de la página debería ser similar al de la siguiente ilustración.

Name: [] *

Birth Date: [] *

Coverage: [] *

☐ Short-term disability ☐ Long-term disability

Save [] These errors were found.

[lblMessage] • Error message 1.
• Error message 2.

2. Abrir el procedimiento de evento **Click** para el botón **Guardar**.
3. Modificar el código del procedimiento de evento para probar si los controles de la página son válidos y, si lo son, escribir un mensaje en el control **lblMessage** indicando que la página es válida.

El código debería ser como el siguiente:

Visual Basic .NET

```
If Page.IsValid Then  
    lblMessage.Text = "La página es válida"  
End If
```

C#

```
if (Page.IsValid)  
{  
    lblMessage.Text = "La página es válida";  
}
```

4. Generar y examinar la página life.aspx.
5. En el explorador, introducir valores para los campos **Name**, **Birth Date** y **Coverage** y hacer clic en **Guardar**.

Verificar que aparece el mensaje **La página es válida**.

Ejercicio 3

Uso del control CompareValidator

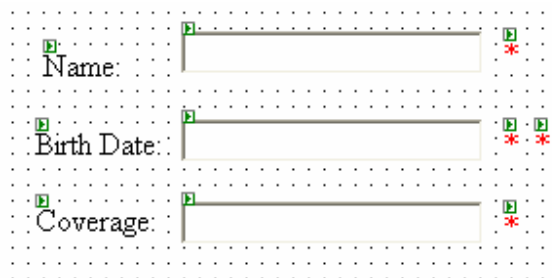
En este ejercicio, utilizaremos un control **CompareValidator** para validar la entrada del usuario en el campo de texto **Fecha de nacimiento**.

Agregar un control CompareValidator

1. Mediante una operación de arrastrar y soltar, ubicar un control **CompareValidator** en el formulario Web Form life.aspx, junto al control **vldBirth RequiredFieldValidator**.
2. Establecer las propiedades del control **CompareValidator** como se muestra en la siguiente tabla.

Propiedad	Valor
ErrorMessage	El valor de Fecha de nacimiento no es válido
Text	*
ControlToValidate	txtBirth
Operator	DataTypeCheck
Type	Date
ID	vldBirthType

El aspecto de la página debería ser similar al de la siguiente ilustración.



The screenshot shows a web form with three text input fields. The first field is labeled 'Name:' and has a small green icon to its left and a red asterisk icon to its right. The second field is labeled 'Birth Date:' and has a small green icon to its left and two red asterisk icons to its right. The third field is labeled 'Coverage:' and has a small green icon to its left and a red asterisk icon to its right. The form is displayed on a grid background.

3. Generar y examinar la página life.aspx.
4. Introducir una fecha no válida, como **14/3/02** (el mes 14 es incorrecto), en el campo **Birth Date** y hacer clic en **Guardar**.
Debería aparecer el asterisco rojo junto al campo **Birth Date**.
5. Introducir una fecha válida, como **12/3/02**, en el campo **Birth Date** y hacer clic en **Guardar**.
El asterisco rojo junto al campo **Birth Date** debería desaparecer.

Ejercicio 4

Uso del control `RegularExpressionValidator`

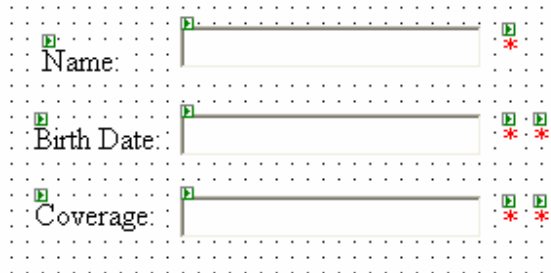
En este ejercicio, utilizaremos un control **`RegularExpressionValidator`** para validar la entrada del usuario en el campo de texto **Coverage**.

Agregar un control `RegularExpressionValidator`

1. Mediante una operación de arrastrar y soltar, colocar un control **`RegularExpressionValidator`** en el formulario Web Form `life.aspx`, junto al control **`vldCoverage RequiredFieldValidator`**.
2. Establecer las propiedades del control como muestra la siguiente tabla.

Propiedad	Valor
ErrorMessage	La Cobertura debe tener un valor de tipo moneda
Text	*
ControlToValidate	<code>txtCoverage</code>
ValidationExpression	<code>\d + (\. \d { 2 }) ?</code>
ID	<code>vldCoverageType</code>

El aspecto de la página debería ser similar al de la siguiente ilustración.



The screenshot shows a web form with three text input fields. The first field is labeled 'Name:', the second 'Birth Date:', and the third 'Coverage:'. The 'Coverage' field has a red asterisk next to it, indicating a validation error. The form is set against a dotted grid background.

3. ¿Cuáles son las entradas válidas para el campo **Coverage**?

4. Generar y examinar la página `life.aspx`.
5. Introducir un valor incorrecto, como **333.3**, en el campo **Coverage** y hacer clic en **Guardar**.

Debería aparecer el asterisco rojo junto al campo **Coverage**.

6. Introducir un valor correcto, como **3.33**, en el campo **Coverage** y hacer clic en **Guardar**.

El asterisco rojo junto al campo **Coverage** debería desaparecer.