

Objetivo

Al finalizar el desarrollo de este ejercicio, el estudiante:

- Estará en capacidad de implementar controles para el manejo de errores a fin de mejorar la comunicación de la aplicación con el usuario

Metodología

A continuación se presenta el enunciado del ejercicio y una serie de pasos para su desarrollo. Recuerde que debe haber leído la parte conceptual del tema, antes de realizar esta práctica en clase.

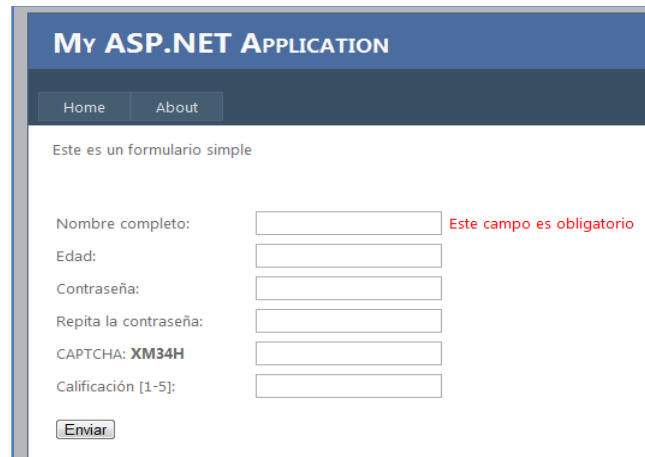
Mostrar mensajes de error durante la ejecución de un formulario

1. En un proyecto de aplicación Web adicione una tabla de 3 columnas y ubique en las dos primeras los controles necesarios como muestra en la siguiente figura. Nombre los controles respectivamente al siguiente estándar: tb_nombre, tb_edad, tb_contraseña, tb_contraseña2, tb_captcha, tb_calificacion, b_enviar.



Durante este ejercicio se adicionarán controles de validación a los campos.

2. En la tercera columna al lado de tb_nombre, adicione un control **RequiredFieldValidator** al formulario y configure las siguientes propiedades:
 - a. Id: rfv_nombre
 - b. ErrorMessage: Este campo es obligatorio
 - c. ForeColor: red
 - d. ControlToValidate: tb_nombre
3. Ejecute la aplicación y pruebe el control dejando el campo vacío y dado click en el botón "Enviar". Su aplicación debe lucir como la siguiente imagen:



4. Adicione al resto del formulario los demás controles de validación en la siguiente correspondencia:
 - a. `tb_edad` : `RangeValidator`, que valide entre 1 y 100 suponiendo que no se registrarán personas mayores a 100 años
 - b. `tb_contraseña2`: `CompareValidator`, donde compare `tb_contraseña` con `tb_contraseña2`
 - c. `tb_calificacion`: `RegularExpressionValidator`, con la expresión regular `[1-5]`
5. Pruebe su aplicación ingresando valores no válidos y debe tener una salida de errores como esta:

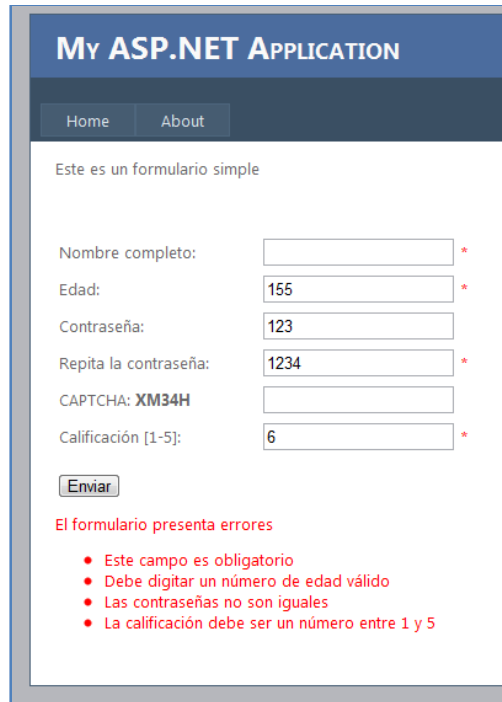


Explore las otras propiedades de los componentes y revise su comportamiento.

Mostrar los mensajes de error en un “resumen”

1. En el mismo ejercicio adicione un **ValidationSummary** y configure las siguientes propiedades:
 - a. `Id`: `vs_resumenValidacion`
 - b. `ForeColor`: `red`
 - c. `HeaderText`: El formulario presenta errores

2. Modifique los controles anteriores modificando el valor de la propiedad Text por un asterisco “*”
3. Ejecute la aplicación y pruebe el control dejando el campo vacío y dado click en el botón “Enviar”. Su aplicación debe lucir como la siguiente imagen:



The screenshot shows a web application titled "My ASP.NET APPLICATION" with a navigation bar containing "Home" and "About" links. The main content area displays a simple form with the following fields and values:

Nombre completo:	<input type="text"/>	*
Edad:	155	*
Contraseña:	123	
Repita la contraseña:	1234	*
CAPTCHA: XM34H	<input type="text"/>	
Calificación [1-5]:	6	*

Below the form is an "Enviar" button. A red error message is displayed below the button:

El formulario presenta errores

- Este campo es obligatorio
- Debe digitar un número de edad válido
- Las contraseñas no son iguales
- La calificación debe ser un número entre 1 y 5

Investigación

Investigue el comportamiento del control: CustomValidator y aplíquelo sobre el control tb_captcha. Tenga en cuenta que este control es especial, pues permite hacer validaciones personalizadas lo que significa que usted debe crear el método de la validación y enlazarlo al control.