

## Para Validar controles de entrada de formularios

El Framework de Formularios Web incluye un conjunto de controles de servidor de validación que proporcionan una forma sencilla pero poderosa de comprobar los formularios de entrada en busca de errores y, si es necesario, mostrar mensajes al usuario.

Podemos "ligar" más de un control de validación a un control de entrada. Por ejemplo, podríamos especificar tanto que un campo es obligatorio y que debe contener un rango específico de valores. Los controles de validación trabajan un limitado subconjunto de controles de servidor HTML y Web. Para cada control, una propiedad específica contiene el valor que se validará. La siguiente tabla muestra los controles de entrada que pueden ser validados.

Control	Propiedad de Validación
HtmlInputText	Value
HtmlTextArea	
HtmlSelect	
HtmlInputFile	
TextBox	Text
ListBox	SelectedItem.Value
DropDownList	
RadioButtonList	
FileUpload	FileName

## Tipos de Controles de Validación

El formulario de validación más sencillo es un campo obligatorio. Si el usuario introduce un valor en el campo, es válido. Si todos los campos de la página son válidos, la página es válida.

1. **RequiredFieldValidator:** Asegura que el usuario no se deja un campo en blanco.
2. **CompareValidator:** Compara los datos que introduce el usuario con una constante o el valor de una propiedad de otro control mediante un operador de comparación (menor que, igual que, mayor que, etc.).
3. **RangeValidator:** Comprueba que la entrada del usuario se encuentra entre un límite superior y otro inferior. Podemos comprobar los rangos con parejas de números, caracteres alfabéticos o fechas. Los límites se pueden expresar como constantes.
4. **RegularExpressionValidator:** Comprueba que la entrada sigue un patrón definido como una expresión regular. Este tipo de validación nos permite comprobar secuencias

predicibles de caracteres, tales como números de seguridad social, dirección de e-mail, números de teléfono, códigos postales, etc.

5. **CustomValidator:** Comprueba la entrada de usuario mediante lógica de validación que hemos programado nosotros. Este tipo de validación nos permite comprobar valores obtenidos en tiempo de validación.

1. **RequiredFieldValidator:**

- **ControlToValidate:** Establece el control de entrada que se va a validar.
- **ErrorMessage:** Establece el texto del mensaje de error que se muestra en un control.
- **CssClass:** Agregar estilos bootstrap.

```
<asp:RequiredFieldValidator ID="rfvNombre" runat="server" ErrorMessage="Proporcione el nombre"
ControlToValidate="txtNombre" CssClass="text-danger"></asp:RequiredFieldValidator><br/>
```

2. **CompareValidator:**

- **ControlToCompare:** Establece el control de entrada para compararlo con el control de entrada que se está validando.

3. **RangeValidator:**

- **MaximumValue:** Establece el valor máximo del intervalo de validación.
- **MinimumValue:** Establece el valor mínimo del intervalo de validación.
- **Type:** Tipo de datos que especifica cómo interpretar los valores que se van a comparar.

```
<asp:RangeValidator ID="rvFechaNacimiento" runat="server" ErrorMessage="La fecha debe estar entre 01-01-1980
y 31-12-1990" ControlToValidate="txtFechaNacimiento" CssClass="text-danger" MaximumValue="01-01-1990"
MinimumValue="31-12-1980" Type="Date"></asp:RangeValidator>
```

4. **RegularExpressionValidator:**

- **ValidationExpression:** Establece la expresión regular que determina el patrón utilizado para validar un campo.
- **MatchTimeout:** Establece el intervalo de tiempo máximo para ejecutar una única operación de coincidencia antes de que se agote el tiempo de espera de la operación.

```
<asp:RegularExpressionValidator ID="revCurp" runat="server" ErrorMessage="No cumple con el formato"
ControlToValidate="txtCURP" CssClass="text-danger" ValidationExpression="^[A-Z]{1}[AEIOU]{1}[A-Z]{2}
</asp:RegularExpressionValidator><br/>
```

5. **CustomValidator:**

- **ClientValidationFunction:** Establece el nombre de la función de script de cliente personalizada que se usa para la validación.
- **ValidateEmptyText:** establece un valor booleano que indica si se debe validar el texto vacío.

```
<asp:CustomValidator ID="cvCURPNacimiento" runat="server" ErrorMessage="Fecha del CURP no coincide con la de
Nacimiento"
ControlToValidate="txtCURP" CssClass="text-danger" OnServerValidate="cvCURPNacimiento_ServerValidate">
</asp:CustomValidator>
```