

Tutorial: Validar los datos introducidos por el usuario en una página de formularios Web Forms

Este tutorial muestra cómo utilizar los controles validadores de ASP.NET para comprobar los datos proporcionados por el usuario en una página Web.

- Trabajarás con controles que realizan las comprobaciones automáticamente, sin necesidad de código.
- Crearás un validador personalizado mediante código
- y finalmente, verás cómo aplicar una validación u otra dependiendo de la opción que el usuario elige en la página

Crearás una página **que permita realizar una reserva**. Dado que la finalidad del tutorial consiste en ilustrar la validación, el carácter de la reserva no es importante (por ejemplo, podría ser para un restaurante, una sala de reuniones de un centro social u otro tipo de reserva), pero la página realmente no procesa la reserva.

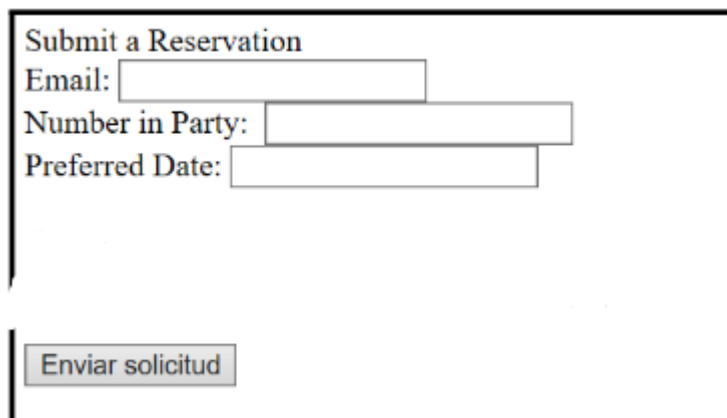
En la página se pedirá al usuario: una dirección de correo electrónico, el número de personas para las que solicita reserva y una fecha. La página también permite al usuario solicitar confirmación telefónica de la reserva. (La opción predeterminada para esta página, si se implementara, sería una confirmación por correo electrónico).

Durante este tutorial aprenderás a:

- Utilizar los controles validadores para comprobar los datos proporcionados por el usuario en una página Web ASP.NET.
- Dar formato a la presentación de los mensajes de error de validación y
- Crear una validación personalizada mediante código.

Primera parte

Agregar los siguientes controles y texto para lograr el diseño propuesto. Escribe el texto delante de los cuadros de texto como títulos.



Submit a Reservation

Email:

Number in Party:

Preferred Date:

a. Control	b. Propiedades
TextBox	ID : textEmail
TextBox	ID : textNumberInParty
TextBox	ID : textPreferredDate
Button	ID : buttonSubmit Text : Enviar solicitud ValidationGroup : AllValidators
Label	ID : labelMessage Text : (en blanco)

Haz doble clic en el botón **Submit Request** (*Enviar*) con objeto de crear un controlador para su evento Click y, a continuación, agrega el siguiente código:

```
Protected Sub buttonSubmit_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs)
    If Page.IsValid Then
        labelMessage.Text = "Tu reserva ha sido procesada..."
    End If
End Sub
```

El controlador del botón sólo muestra un mensaje; no realiza ningún procesamiento real. Sin embargo, este mensaje te va a permitir probar el efecto de la validación. (El texto se va a escribir cuando la página esté libre de errores)

En la página .aspx, cambia a la **Vista de diseño**, haz doble clic en una zona en blanco de la página para crear un controlador **Page_Load** y, a continuación, agrega el código:

```
Protected Sub Page_Load(ByVal sender As Object, _
    ByVal e As EventArgs)
    lblMessage.Text = ""
End Sub
```

Cada vez que la página se carga, este código borra cualquier mensaje que pueda haber quedado de una devolución (postback) anterior.

Agregar validación básica

Para tu sistema de reservas imaginario, necesitas que se realicen las siguientes comprobaciones de validación:

- la dirección de correo electrónico es obligatoria y su formato tiene que ser correcto; por ejemplo, alguien@ejemplo.com
- el número de personas debe ser un valor numérico.
- la fecha ha de ser válida

Vas a realizar todas estas validaciones mediante **controles de validación**, que se encargan de realizar las comprobaciones y muestran los errores automáticamente.

Desde el grupo **Validación** del **Cuadro de herramientas**, arrastra un control [RequiredFieldValidator](#) y colócalo junto al cuadro de texto **textEmail**.

Establece las siguientes propiedades del control [RequiredFieldValidator](#):

Propiedad	Ajuste
ControlToValidate	textEmail Enlaza el control validador al cuadro de texto cuyo contenido desea validar.
Display	Dinámico Especifica que el control sólo representa (y ocupa espacio en la página) si es necesario mostrar un error.
ErrorMessage	"Se requiere la dirección de correo electrónico."

	Proporciona el texto de un control ValidationSummary , que se configurará más adelante en este tutorial.
Text	<p>*</p> <p>Un asterisco es una convención que se usa para indicar que se trata de un campo obligatorio. Este texto se muestra en el control de validación y se muestra únicamente si hay un error.</p>
ValidationGroup	<p>AllValidators</p> <p>Al igual que ocurre con los botones de radio, es posible reunir los validadores en grupos que se tratan como una unidad.</p>

El control validador que acabas de configurar asegura la obligatoriedad de completar el cuadro de texto que recogerá el mail. Pero además, precisamos que el mail se ajuste a una expresión concreta, para ello:

Desde el grupo **Validación** del **Cuadro de herramientas**, arrastra un control [RegularExpressionValidator](#) y colócalo junto al control [RequiredFieldValidator](#) que acabas de agregar.

Establece las siguientes propiedades del control [RegularExpressionValidator](#):

Propiedad	Configuración
ControlToValidate	<p>textEmail</p> <p>Está validando de nuevo la entrada del usuario en el cuadro correspondiente al correo electrónico.</p>
Display	Dinámico

ErrorMessage	<p>"Las direcciones de correo electrónico deben estar en el formato nombre@dominio.xyz."</p> <p>Es el mensaje de error "largo/detallado" que se mostrará mediante un control ValidationSummary que configurarás más adelante</p>
Text	<p>"Formato no válido."</p> <p>Es un mensaje de error "corto" que se mostrará en la posición en la que coloques este control de validación.</p>
ValidationGroup	AllValidators

Con el control [RegularExpressionValidator](#) todavía seleccionado, haz clic en el botón de puntos suspensivos en la propiedad [ValidationExpression](#) (ventana Propiedades del control).

Las expresiones regulares constituyen un lenguaje que se puede utilizar para buscar con precisión modelos definidos en cadenas. En el control [RegularExpressionValidator](#), debemos definir una expresión regular válida para el modelo (en este caso, el modelo de una dirección de correo electrónico válida). El **Editor de expresiones regulares** contiene una lista de expresiones regulares utilizadas con frecuencia para que podamos usar el control validador sin tener que conocer la sintaxis de las expresiones regulares.

En la lista **Expresiones estándar**, haz clic en **Dirección de correo electrónico de Internet**. La expresión regular para una dirección de correo electrónico se coloca en el cuadro **Expresión de validación**. Haga clic en **Aceptar** para cerrar el cuadro de diálogo. Con esto has terminado de validar el mail.

Preparando la validación de **textNumberInParty**:

Agrega otro control [RequiredFieldValidator](#) para hacer obligatoria la entrada de un valor en el cuadro de texto **textNumberInParty**. Ten en cuenta que ahora el control a validar es otro, la propiedad [ErrorMessage](#) tomará otro valor (ej. "Indique cuántas personas habrá en la fiesta."), etc.

El número de personas queremos limitarlo a 20. Para ello: arrastra un control [RangeValidator](#) y colócalo junto al control [RequiredFieldValidator](#) que acabas de agregar. Configura las siguientes propiedades del control [RangeValidator](#):

Propiedad	Configuración
ControlToValidate	textNumberInParty

Display	Dinámico
ErrorMessage	"Escriba un número entre 1 y 20"
MaximumValue	20
MinimumValue	1 En esta aplicación, para una reserva se requiere por lo menos una persona.
Text	"Escriba un número entre 1 y 20."
Type	Integer
ValidationGroup	AllValidators

Este control [RangeValidator](#) realiza dos funciones:

- garantiza que los datos que introduce un usuario son **numéricos** y
- comprueba que el número introducido se encuentra entre los **valores mínimo y máximo especificados**.

Probar la página

Ahora puedes probar los controles validadores que tienes hasta el momento. Para ello:

1. Presiona CTRL+F5 para ejecutar la página.
2. Cuando la página aparece en el explorador, haz clic en el botón **Submit Request**.
3. Se muestran varios errores de validación porque no has rellenado algunos campos necesarios. **Ojo!!** Observa cómo se muestran los errores de validación inmediatamente: **no se envía la página al servidor!**. De forma predeterminada, los controles validadores insertan ECMAScript de cliente (JavaScript) en la página para realizar una comprobación de validación en el explorador. Esto proporciona a los usuarios información instantánea sobre errores de validación; sin script de cliente, para la comprobación de errores de validación se necesitaría una acción de ida y vuelta al servidor, lo que en ocasiones podría

resultar lento. De hecho, no se va a enviar la página hasta que todas las comprobaciones de validación del lado cliente se realizan correctamente.

4. Comprueba la validación en el cuadro de texto **textEmail**; para ello, escribe una dirección de correo electrónico no válida y, a continuación, repite la prueba con una dirección de correo electrónico válida.
5. Comprueba que sólo se pueden escribir valores numéricos entre el 1 y el 20 en el cuadro de texto **Número de personas de la fiesta**.
6. Cuando hayas escrito todos los valores válidos, haz clic en el botón **Enviar solicitud**. Si todos los controles pasan la validación, verás el mensaje "Se ha procesado su reserva".
7. Después de haber confirmado que esa validación funciona según lo previsto, cierra el explorador.

Si la validación no funciona según lo previsto, asegúrate de que has establecido todos los valores de configuración de propiedades enumerados anteriormente y después ejecuta de nuevo la página.

Es importante que llegues hasta aquí habiendo entendido lo que acabas de hacer. No continúes si no lo has logrado. Revisa y repite el proceso hasta lograrlo.

Formas alternativas de mostrar la información de errores

De forma predeterminada, los controles de validación muestran el texto de los errores en la ubicación del control validador en la página. Pero hay ocasiones en las que puede interesar que la información de los errores se muestre de otro modo. Los controles validadores de ASP.NET proporcionan estas opciones adicionales:

- Resumir/juntar todos los errores de validación en un mismo lugar de la página *(además de que se muestren en la posición del control validador o bien como alternativa a dicha presentación)*.
- Mostrar un mensaje en una ventana emergente en el explorador. *Esta opción solo funciona en exploradores habilitados para el uso de scripts.*

Estas alternativas a la forma de presentar los errores la logramos con [el control ValidationSummary](#):

1. Arrastra un control ValidationSummary desde el grupo **Validación** del **Cuadro de herramientas** hasta la parte inferior de la página.
2. Asigna a la **propiedad ValidationGroup** el valor AllValidators.
3. Ejecuta la página.
4. Realiza las mismas pruebas que has realizado antes cuando has hecho las pruebas de una validación básica:

- a. Ahora para cada error verás información en dos lugares: un mensaje de error “breve” (el valor de la propiedad **Text** del control validador) donde se encuentra ubicado el control validador, y el mensaje de error más largo (el valor de la propiedad **ErrorMessage**) en la posición en la que has ubicado el control [ValidationSummary](#).
 - b. Cierra el explorador.
5. Establece la **propiedad ShowMessageBox** del control [ValidationSummary](#) en **true**.
6. Ejecuta la página.
7. Realiza las mismas pruebas.
8. Ahora cada error produce un mensaje en una pequeña emergente del explorador.
9. Cierra el explorador.

Con esto has finalizado la primera parte de la práctica (validación básica + alternativas a la muestra de errores)

Segunda parte - validación personalizada

(validar el campo fecha)

Empieza haciendo obligatoria la entrada de un valor en **textPreferredDate** tal y como lo has hecho para el resto de campos del formulario.

El control [CustomValidator](#) permite escribir código de validación propio para comprobar cualquier condición que necesite nuestra aplicación.

Recuerda que hemos mencionado que los controles de validación de asp Net realizan en primer lugar la validación mediante JavaScript y además, el servidor de aplicaciones les aplica una validación adicional del lado servidor. La validación personalizada que preparemos con CustomValidator ha de hacer lo mismo: validación en el servidor que se ejecutará cuando se envíe la página y validación Javascript en el cliente. Requiere por tanto que escribas:

- código de validación en tu lenguaje de programación servidor (se ejecutará cuando se envía la página)
- y script de cliente (ECMAScript o JavaScript) que realizará una comprobación similar en el explorador para que el usuario tenga información inmediata

Desde el grupo **Validación** del **Cuadro de herramientas**, arrastra un [control CustomValidator](#) a la página y colócalo [junto al cuadro de texto textPreferredDate](#).

Establece las siguientes propiedades del control [CustomValidator](#):

Propiedad	Configuración
ControlToValidate	textPreferredDate
Display	Dinámico
ErrorMessage	"Escriba una fecha en el formato m/d/aaaa."
Text	"Formato de fecha no válido (se requiere m/d/aaaa)."
ValidationGroup	AllValidators

Haz doble clic en el control [CustomValidator](#) con objeto de crear un controlador para su evento [ServerValidate](#) y, a continuación agrega el código, tal y como se muestra en el siguiente ejemplo:

```
Protected Sub CustomValidator1_ServerValidate( _
    ByVal source As Object, _
    ByVal args As System.Web.UI.WebControls.ServerValidateEventArgs)
    Try
        DateTime.ParseExact(args.Value, "m/d/yyyy", _
            System.Globalization.DateTimeFormatInfo.InvariantInfo)
        args.IsValid = True
    Catch
        args.IsValid = False
    End Try
End Sub
```

Este código se ejecuta cuando el usuario envía la página. El código interactúa con el control validador mediante el objeto [ServerValidateEventArgs](#) (*args*) pasado al controlador.

El valor que el usuario ha especificado en el cuadro de texto **textPreferredDate** se pasa como la propiedad [Value](#) del objeto *args*. Después de haber comprobado que la entrada

del usuario es válida, establece la propiedad [IsValid](#) del objeto *args* en **true** o **false**. Si la establece en **false**, el validador mostrará su mensaje de error.

El código utiliza un bloque try-catch para determinar si la entrada del usuario se puede convertir en un objeto [DateTime](#). Si el usuario especifica un valor no válido (que no se ajusta a una fecha de formato m/d/aaaa), el método [ParseExact](#) del objeto [DateTime](#) produce una excepción y se ejecuta el bloque **Catch**.

Nota

En el ejemplo de código se especifica que el formato que se va a analizar es independiente de la referencia cultural. Si necesita usar la referencia cultural actual, establezca el parámetro *provider* en **null**.

En el controlador `Button_Click`, agrega el código tal y como se muestra a continuación:

VB

```
Protected Sub buttonSubmit_Click(ByVal sender As Object, _  
    ByVal e As System.EventArgs)  
  
    If Page.IsValid Then  
        labelMessage.Text = "Tu reserva ha sido procesada..."  
    Else  
        labelMessage.Text = "Página no válida."  
    End If  
End Sub
```

Ojo!! Cuando se utiliza un control [CustomValidator](#), debe comprobarse la propiedad [Page.IsValid](#) en cualquier procesamiento de servidor para asegurarse de que se han superado todas las comprobaciones de validación. La propiedad [IsValid](#) devuelve el estado acumulado de todos los controles validadores en la página. Esta propiedad se utiliza para asegurarse de que todas las comprobaciones de validación han pasado el procesamiento de servidor.

Hasta aquí has agregado código de servidor para el control [CustomValidator](#) para comprobar la validez de una fecha. Recuerda que los controles validadores también comprueban la entrada de un usuario mediante script de cliente. Así que lo propio es agregar script de cliente al control [CustomValidator](#). En realidad, el script de cliente que escribirás duplica la lógica que aplica la validación de servidor. En realidad es al contrario, es el código de servidor el que duplicará la validación de cliente. *Has de saber que no siempre es viable escribir código de validación personalizado de cliente (por*

ejemplo, si el código personalizado valida una entrada del usuario mediante consulta en una base de datos de servidor). Sin embargo, en este caso sí que podemos crear código de cliente que realice esencialmente la misma comprobación que está realizando su código de servidor.

Agregar script de cliente al control CustomValidator

Abre o cambia a la página tarea6-ValidationExample.aspx, y cambia a la vista **Código fuente**.

En el elemento **<head>** de la página, agrega el siguiente bloque de script de cliente:

```
<script type="text/javascript" language="javascript">

    function validateDate(oSrc, args)
    {
        var iDay, iMonth, iYear;
        var arrValues;
        arrValues = args.Value.split("/");
        iMonth = arrValues[0];
        iDay = arrValues[1];
        iYear = arrValues[2];

        var testDate = new Date(iYear, iMonth - 1, iDay);
        if ((testDate.getDate() != iDay) ||
            (testDate.getMonth() != iMonth - 1) ||
            (testDate.getFullYear() != iYear))
        {
            args.IsValid = false;
            return;
        }

        return true;
    }

</script>
```

Nota

ECMAScript (JavaScript) distingue entre mayúsculas y minúsculas; escribe exactamente el código como se ve aquí.

*Este código es similar al de comprobación de validación de servidor, pero no tan estricto. La entrada del usuario se pasa a la función como la propiedad **Value** del objeto **args**, y se puede establecer la propiedad **IsValid** del objeto para indicar si el valor pasa la*

validación. El código comprueba que se han escrito fechas con un formato numérico válido.

Observa que el código es ligeramente diferente en el script de cliente que en el código de servidor. ECMAScript no proporciona exactamente la misma funcionalidad que en .NET Framework. ECMAScript, en lugar del método [ParseExact](#) que se utiliza en el servidor, analiza el texto escrito para comprobar la fecha. Las dos comprobaciones de validación (cliente y servidor) son bastante parecidas para proporcionarle la funcionalidad que necesita.

Coloca el punto de inserción en el elemento **asp:customvalidator** para seleccionarlo. En la ventana Propiedades, establece la propiedad [ClientValidationFunction](#) del control como *validateDate*, que es el nombre de la función ECMAScript que has creado.

De esta manera, has agregado una validación personalizada que funciona primero en el script de cliente para comprobar inmediatamente lo que el usuario ha tecleado y después, cuando la página se envía al servidor, lo comprueba de nuevo.

Para comprobar que lo que has preparado funciona correctamente, debes comprobar ambos códigos. La comprobación la haremos en el orden inverso a como realmente suceden ambas validaciones. Primero comprobarás si la validación en servidor funciona y después comprobarás si funciona en el navegador (cliente). El paso a paso de cómo hacerlo se detalla a continuación:

Para probar la validación personalizada

1. Deshabilita temporalmente la validación en el cliente; para ello, has de establecer la propiedad [EnableClientScript](#) del control [CustomValidator](#) en **false**.
2. Presiona CTRL+F5 para ejecutar la página.
3. Especifica una dirección de correo electrónico y el número de personas de la fiesta para que la validación de estos campos sea correcta.
4. En el cuadro de texto de la fecha, escribe una cadena que **no** sea una fecha y, a continuación, haz clic en el botón **Submit Request**.
5. La página realiza una acción de ida y vuelta al servidor web y la validación de servidor produce un error. *Se produce un error en la prueba de [IsValid](#) y [LabelMessage](#) indica que la página no es válida.*
6. Especifica ahora un valor de fecha válida (ej. 02/02/2023) y haz clic en el botón.
7. En esta ocasión, el valor de la fecha pasa la validación, por lo que has de ver el mensaje de confirmación creado en el controlador [Click](#) del botón.
8. Cierra el explorador.
9. Vuelva a habilitar la validación en el cliente; para ello, establece nuevamente la propiedad [EnableClientScript](#) del control [CustomValidator](#) en **true**.

10. Ejecuta de nuevo la página y escribe valores válidos para la dirección de correo electrónico y el número de personas de la fiesta.
11. Escribe un valor de fecha **no válido** en el cuadro de texto de la fecha y, a continuación, presiona **la tecla TAB**. (**Ojo!!** No hagas clic en el botón; pretendemos comprobar la validación en el cliente y ver si realmente se produce la validación **antes del postback**).
12. En cuanto el cuadro de texto de fecha pierde el foco, el script de validación de cliente se ejecuta y aparece un mensaje de error.

Nota

Si utiliza la función Autocompletar de Internet Explorer, al seleccionar un valor de la lista Autocompletar, se llenará el cuadro de texto con un valor, pero no se ejecutará el validador del cliente.

13. Haz clic en el botón **Submit Request**.
14. Dado que no se ha podido realizar la validación en el cliente para el cuadro de texto de fecha, no se envía la página.
15. Corrige el valor de la fecha y presiona de nuevo la tecla **TAB**.
16. El mensaje de error desaparece. Ahora sí que puedes **enviar el formulario**.

Tercera parte - validación condicional

En la parte final del tutorial, vas a agregar información opcional al formulario de reservas. Los usuarios pueden activar una casilla para indicar que desean confirmación telefónica de sus reservas. Si lo hacen, deben escribir su número de teléfono. Por consiguiente, la página contendrá dos controles nuevos: una casilla y un cuadro de texto.

Utilizarás un control [RequiredFieldValidator](#) para asegurarte de que los usuarios escriben un número de teléfono, y un control [RegularExpressionValidator](#) para comprobar el formato.

El número de teléfono es opcional; sólo se tiene que comprobar el número de teléfono si el usuario ha activado la casilla. Por lo tanto, escribirás un código sencillo con el que se

active o desactive la validación del número de teléfono en función del estado de esta casilla. Tu formulario tendrá el siguiente aspecto:

Submit a Reservation

Email:

Number in Party:

Preferred Date:

☐ Confirme la reserva por teléfono

Número de teléfono:

Enviar solicitud

Desde el grupo **Estándar** del **Cuadro de herramientas**, arrastra un control [CheckBox](#) a la página, y establece las propiedades siguientes:

Propiedad	Configuración
ID	checkPhoneConfirmation
AutoPostBack	<p>True</p> <p>Cuando el usuario haga clic en la casilla, la página realizará una acción de ida y vuelta y habilitará o no el cuadro de texto y la validación de éste.</p>
CausesValidation	<p>False</p> <p>No se producirá automáticamente la validación al hacer clic en CheckBox.</p>
Text	Confirme la reserva por teléfono.

Arrastra un control [TextBox](#) a la página, bajo el control [CheckBox](#) y establece las propiedades siguientes:

ID	textPhoneNumber
Enabled	False No se habilitará el cuadro de texto hasta que el usuario haga clic en la casilla.

Desde el grupo **Validación** del **Cuadro de herramientas**, arrastra [RequiredFieldValidator](#) a la página junto al cuadro de texto de número de teléfono y establece las siguientes propiedades:

Propiedad	Configuración
ID	validatorRequiredPhoneNumber No estableció el id. de los controles validadores anteriormente en el tutorial, pero en este caso hará referencia al control validador en código, por lo que te resultará útil proporcionarle un id. nemotécnico.
ControlToValidate	textPhoneNumber
Display	Dinámico
ErrorMessage	Debe proporcionar un número de teléfono.
Text	*

ValidationGroup	<p>(en blanco)</p> <p>Si esta propiedad está en blanco, significa que el validador no pertenece al grupo AllValidators que has establecido para los otros controles validadores en la página. Como consecuencia de ello, este validador no se comprueba de forma predeterminada cuando se hace clic en el control <code>buttonSubmit</code>.</p>
---------------------------------	---

Arrastra un control [RegularExpressionValidator](#) a la página y establece las propiedades siguientes:

Propiedad	Configuración
ID	validatorRegExPhoneNumber
ControlToValidate	textPhoneNumber
Display	Dinámico
ErrorMessage	El formato del número de teléfono no es válido
Text	Formato no válido
ValidationExpression	(Use el cuadro de diálogo Editor de expresiones regulares para seleccionar Número de teléfono en EE.UU. u otra expresión de número de teléfono.)
ValidationGroup	(Dejarlo en blanco)

Haz doble clic en el control **checkPhoneConfirmation** con objeto de crear un controlador para su evento [CheckedChanged](#) y, a continuación, agrega el siguiente código:

VB

```
Protected Sub checkPhoneConfirmation_CheckedChanged( _  
    ByVal sender As Object, _  
    ByVal e As System.EventArgs)  
    If checkPhoneConfirmation.Checked = True Then  
        textPhoneNumber.Enabled = True  
        validatorRequiredPhoneNumber.ValidationGroup = "AllValidators"  
        validatorRegExPhoneNumber.ValidationGroup = "AllValidators"  
    Else  
        textPhoneNumber.Enabled = False  
        validatorRequiredPhoneNumber.ValidationGroup = ""  
        validatorRegExPhoneNumber.ValidationGroup = ""  
    End If  
End Sub
```

Cuando el usuario hace clic en la casilla, los dos controles validadores asociados al cuadro de texto se agregan al grupo de validación al que pertenecen los otros controles validadores. El efecto es que todos los validadores de la página, incluidos los del número de teléfono, se comprobarán cuando el usuario envíe la página. Si el usuario desactiva la casilla, los validadores se quitan del grupo y, por lo tanto, no se procesan cuando se hace clic en el botón **Enviar solicitud**.

Probar la validación condicional

Ahora puedes probar si esa validación condicional funciona correctamente. Para ello:

1. Presiona CTRL+F5 para ejecutar la página.
2. Escribe la información correspondiente a la dirección de correo electrónico, el número de personas que hay en la fiesta y la fecha.
3. Haz clic en **Enviar solicitud**.
4. Se envía la página y aparece el mensaje de confirmación.
5. Activa la casilla **Confirmar la reserva por teléfono**.
6. Haz clic de nuevo en **Enviar solicitud**.
7. En esta ocasión, verás un mensaje de error (una estrella junto al cuadro de texto correspondiente al teléfono). Al hacer clic en la casilla, has habilitado la validación para el cuadro de texto.
8. Escribe un número de teléfono no válido y, a continuación, haz clic de nuevo en **Enviar solicitud** para confirmar que el cuadro de texto no aceptará un número de teléfono no válido.

9. Escribe un número de teléfono con un formato correcto y haz clic en el botón de envío para confirmar que el control validador acepta datos con un formato correcto.

Nota

Si has seleccionado **Número de teléfono de EE.UU.** en el Editor de expresiones regulares para la propiedad [ValidationExpression](#) del control [RegularExpressionValidator](#), un número de teléfono con el formato correcto consta del código de área (campo opcional con 3 caracteres numéricos entre paréntesis o seguidos de un guión), seguido de un conjunto de 3 caracteres numéricos, un guión y, finalmente, un conjunto de 4 caracteres numéricos. Ejemplos válidos son (425) 555-0123, 425-555-0123 ó 555-0123.

Ojo!!,



Muy importante!

- La propiedad [IsValid](#) NO está disponible en los eventos [Init](#) ni [Load](#) del [WebForm](#)