

Para la validación de datos de usuario, ASP.NET dispone de controles integrados para controlar los datos introducidos por los usuarios, según los formatos estándar (*campos obligatorios, fechas bien formadas, números, e-mails...*). Estos controles se validan, en primer lugar, mediante JavaScript. Además, el servidor de aplicaciones les aplica una validación adicional del lado servidor. Por último, la página dispone de una propiedad `IsValid` para determinar si se han realizado con éxito todas las verificaciones.

Añade al **SitioWebTuNombre** el webform ***tarea6-ValidationExample.aspx*** siguiendo con atención las indicaciones del [tutorial propuesto en el aula](#)

Notas aclaratorias:

En una **primera parte** del tutorial realizarás las siguientes **validaciones básicas**:

1. La dirección de correo electrónico:
 - a. obligatoria
 - b. El formato tiene que ser correcto (por ejemplo algo de tipo [alguien@ejemplo.com](#)); Por lo general, no es viable comprobar si la dirección de correo electrónico es real; no obstante, sí se puede comprobar que se ajusta al modelo correspondiente a direcciones de correo electrónico mediante la expresión regular adecuada.
2. El número de personas:
 - a. obligatorio
 - b. debe ser un valor numérico entre 1 y 20
3. Fecha es de obligada introducción.
4. Si la validación pasa todos los controles, se ha de mostrar el mensaje **"Se ha procesado su reserva"**.

En esta primera parte los errores los mostrarás *"en contexto"*, es decir, junto al campo que está siendo validado. Posteriormente harás las modificaciones oportunas para mostrar los errores de otra forma: a modo de **resumen** en la parte inferior del formulario, o a modo de **ventana emergente en el explorador** (lo harás con un control ***ValidationSummary***)

Segunda parte (validación personalizada).

(Apartado: Agregar validación personalizada para comprobar la entrada de la fecha)

En esta parte del tutorial, harás una validación personalizada para el campo fecha con un control ***CustomValidator***: escribirás **código** para que en el servidor se valide que la fecha es válida, y escribirás también **código javascript** que llevará a

cabo una validación similar en el cliente. En realidad, escribirás script de cliente que duplica la lógica que aplica la validación de servidor.

Agregarás de esta forma una comprobación de validación personalizada que, funciona primero en el cliente comprobando de manera inmediata la entrada del usuario y, después, otra que lo volverá a comprobar cuando la página se envía al servidor.

Observa que se muestran los errores de validación inmediatamente: no se envía la página.

De forma predeterminada, los controles validadores que hasta ahora has utilizado insertan ECMAScript de cliente (JavaScript) en la página para realizar una comprobación de validación en el explorador, lo que proporciona a los usuarios información instantánea sobre errores de validación. Piensa que 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 es posible enviar la página hasta que todas las comprobaciones de validación del lado cliente se realizan correctamente. (Como medida de seguridad, se realiza de nuevo la misma comprobación de validación cuando se envía la página).

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.

Tercera parte (validación opcional)

Apartado: Agregar controles opcionales con validación condicional

Es la parte en la que se gestionará la validación del último campo del formulario, el cual habrá de validarse únicamente si se ha activado la casilla de verificación. Así que, escribirás un código sencillo con el que se active o desactive la validación del número de teléfono en función del estado de la casilla.

☐ Confirme la reserva por teléfono
 Número de teléfono:



AspNet -Tarea6-Controles de Validación (I)