

Formularios accesibles (3)

Esta lección pertenece al [curso “Aprende Accesibilidad Web Paso a Paso”](http://accesibilidadweb.es) ¹

Prohibida su distribución sin permiso del autor.

Presentación

Hola, soy Olga Carreras Montoto, consultora en Experiencia de Usuario y Accesibilidad web en [“Usable y accesible”](http://www.usableyaccesible.com) ² (España).

Esta lección es la tercera de la serie “Formularios accesibles”. En ella vamos a aprender a:

- prevenir los errores de los usuarios en los formularios
- mejorar la accesibilidad de los mensajes de validación

Recordemos que:

- en la primera lección aprendimos a etiquetar correctamente los controles de los formularios;
- en la segunda:
 - repasamos los requisitos aplicables propiamente a los controles de los formularios
 - aprendimos a definir semánticamente la relación entre los mismos

¹ <http://accesibilidadweb.es>

² “Usable y accesible”, <http://www.usableyaccesible.com>

¿Cómo prevenir los errores de los usuarios en los formularios?

Identifica los campos obligatorios

En primer lugar debemos identificar los campos obligatorios, preferiblemente con un texto en la etiqueta del campo, y por tanto antes del mismo.

Por ejemplo:

```
<label for="nombre">Nombre  
(obligatorio):</label>
```

```
<input type="text" name="nombre" id="nombre" />
```

Podéis consultar diferentes formas de hacerlo en la técnica [H90: Indicating required form controls using label or legend](#)³

Adicionalmente podemos añadir al campo la propiedad “aria-required” de WAI-ARIA⁴, como veremos en una lección posterior.

Por ejemplo:

```
<input type="text" name="nombre" id="nombre"  
aria-required="true" />
```

No podemos identificar los campos obligatorios solo con el color, por ejemplo con el color del propio campo o de su etiqueta, si acaso puede ser una pista visual adicional.

Si te basas solo en el color para identificar los campos obligatorios, esta información pasará desapercibida para muchas personas, por ejemplo, a aquellas con problemas para percibir el color; o por ejemplo a aquellas que acceden con un producto de apoyo, como un lector de pantalla o una línea braille.

³ WCAG 2.0, Técnica “H90: Indicating required form controls using label or legend”, <http://www.w3.org/TR/WCAG20-TECHS/H90.html>

⁴ WCAG 2.0, Técnica “[ARIA2: Identifying a required field with the aria-required property](#)”, <http://www.w3.org/TR/WCAG20-TECHS/ARIA2.html>

Informa del formato requerido

Otra manera de evitar errores a los usuarios es informándoles de las restricciones en el formato de datos que deben introducir, aunque debería intentarse que los formularios fueran flexibles para introducir diferentes formatos de datos.

Técnica de referencia de las WCAG 2.0: [G89: Providing expected data format and example](#)⁵

Por ejemplo, se puede indicar el formato requerido en la etiqueta del campo y proporcionar un ejemplo bajo el mismo:

```
<label for="fecha">Fecha (dd/mm/aaaa) :</label>  
  
<input type="text" aria-describedby="ayuda" ... />  
  
<p id="ayuda">Por ejemplo: 12/03/2014</p>
```

En el tema sobre WAI-ARIA explicaremos el atributo “aria-describedby” que permite asociar semánticamente la ayuda contextual con el campo, de modo que sea anunciada a los usuarios de lector de pantalla cuando el campo coja el foco.

Puedes consultar un ejemplo del uso de “aria-describedby” en mi artículo [“Ayuda contextual de los formularios más accesible con "aria-describedby" \(WAI-ARIA\)”](#)⁶

⁵ WCAG 2.0, Técnica “G89: Providing expected data format and example”, <http://www.w3.org/TR/WCAG20-TECHS/G89.html>

⁶ Carreras Montoto, Olga: “Ayuda contextual de los formularios más accesible con "aria-describedby" (WAI-ARIA)”, <http://olgacarreras.blogspot.com.es/2013/12/ayuda-contextual-de-los-formularios-mas.html>, 2013

Instrucciones y ayuda

También ayuda a prevenir los errores de los usuarios:

- incluir instrucciones al comienzo del formulario, si son necesarias.
- incluir ayuda dependiente del contexto mediante:
 - un enlace a una página de ayuda,
 - un asistente,
 - una solución de corrección ortográfica que ofrezca sugerencias y un mecanismo simple para seleccionar una de ellas.

Técnicas de referencia de las WCAG 2.0:

- [G184: Providing text instructions at the beginning of a form or set of fields that describes the necessary input](#) ⁷
- Técnicas asociadas al criterio de conformidad [“3.3.5 Ayuda: Se proporciona ayuda dependiente del contexto. \(Nivel AAA\)”](#) ⁸

Prevenir errores legales, financieros y de datos

Por otro lado, el criterio de conformidad [“3.3.4 Prevención de errores \(legales, financieros, datos\)”](#) ⁹ de las WCAG 2.0 trata sobre cómo prevenir errores legales, financieros y de datos para las páginas que:

⁷ WCAG 2.0, Técnica “G184: Providing text instructions at the beginning of a form or set of fields that describes the necessary input”, <http://www.w3.org/TR/WCAG20-TECHS/G184.html>

⁸ WCAG 2.0, Criterio de conformidad “3.3.5 Ayuda: Se proporciona ayuda dependiente del contexto. (Nivel AAA)”, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-context-help.html>

⁹ WCAG 2.0, Criterio de conformidad “3.3.4 Prevención de errores (legales, financieros, datos)”, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-reversible.html>

- representan para el usuario compromisos legales o transacciones financieras;
- modifican o eliminan datos controlables por el usuario en sistemas de almacenamiento de datos;
- envían las respuestas del usuario a una prueba.

En estos casos se debe cumplir al menos con uno de estos requisitos:

- el envío es reversible;
- se verifica la información para detectar errores en la entrada de datos y se proporciona al usuario la oportunidad de corregirlos;
- se proporciona un mecanismo para revisar, confirmar y corregir la información antes de finalizar el envío de los datos.

Para alcanzar el nivel AAA¹⁰ debe cumplirse alguno de estos requisitos en todas las páginas que requieren del usuario envío de información.

Mejorar la accesibilidad de los mensajes de validación

Vamos a ver ahora las técnicas relacionadas con la información que se ofrece cuando el usuario comete un error en el formulario. Veamos cómo hacer más accesibles los mensajes de validación de los formularios

Identificar y describir los errores mediante texto

En primer lugar es necesario identificar y describir los errores, -bien porque no se han rellenado campos obligatorios, bien porque no se respeta determinado formato o valor- y hacerlo

¹⁰ WCAG 2.0, Criterio de conformidad “[3.3.6 Prevención de errores \(todos\) nivel AAA](http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-reversible-all.html)”, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-reversible-all.html>

mediante texto. Complementariamente, si se desea, se puede apoyar esta información con colores, estilos de texto o imágenes.

El objetivo es asegurarse de que todos los usuarios se percatan de que se ha producido un error y dónde¹¹.

¿Cómo lo hacemos?

Preferiblemente, aunque se admiten otras opciones (como ventanas de alertas o información contextual):

- informado mediante texto, al comienzo del formulario, de los errores existentes;
- describiendo el problema e identificando los campos con error (sin basarse solo en el color, como hemos visto anteriormente);
- si es posible, indicando o sugiriendo valores posibles.

Vamos a poner un ejemplo.

Tenemos un formulario donde se solicita obligatoriamente el campo “nombre y apellidos” y el campo “email”.

Imaginemos que pulsamos el botón de envío sin rellenar el campo “email”.

En la parte superior del formulario se informa mediante texto de los errores existentes:

Ha ocurrido el siguiente error:

[icono de advertencia] El campo “email” es obligatorio.

Introduzca una dirección de correo electrónico válida en la cual desee recibir el boletín.

¹¹ WCAG 2.0, Criterio de conformidad [“3.3.1 Identificación de errores \(Nivel A\)”](http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-identified.html),
<http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-identified.html>
WCAG 2.0, Criterio de conformidad [“3.3.3 Sugerencias ante errores \(Nivel AA\)”](http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-suggestions.html),
<http://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-suggestions.html>

Se identifica el campo que ha dado error por su nombre, con un enlace al mismo ([campo “email”](#)).

El propio campo del formulario se resaltará, coloreando de rojo su borde y de rojo claro su interior. Pero no solo se identificará de esta manera que es el campo que ha provocado el error, sino que también se incluirá un mensaje asociado al mismo, colocado antes del campo:

[icono de advertencia] Campo obligatorio. Por favor, indique su dirección de correo electrónico.

Los mensajes aclaran por qué se ha producido el error (el campo era obligatorio) y se indica al usuario que introduzca su dirección de correo electrónico.

Otras consideraciones

Ten en cuenta además que:

- La validación debería ser realizada tanto **en cliente como en servidor**.
- El mensaje de error al comienzo de la página **debe coger el foco** para que no pase desapercibido a ningún usuario. Trataremos el tema del foco en una lección más adelante.
- Si la validación es en cliente, mediante javascript, y por tanto los **mensajes se crean dinámicamente**, deben insertarse correctamente con **funciones DOM**. De esta manera, los agentes de usuario pueden acceder al DOM para recuperar el nuevo contenido y en el orden correcto. Hablaremos más adelante, en una lección posterior, sobre javascript y accesibilidad, pero puedes consultar la técnica SCR21¹² donde se incluye un ejemplo.

¹² WCAG 2.0, Técnica “[SCR21: Using functions of the Document Object Model \(DOM\) to add content to a page](http://www.w3.org/TR/WCAG20-TECHS/client-side-script.html#SCR21)”, <http://www.w3.org/TR/WCAG20-TECHS/client-side-script.html#SCR21>

- Las **zonas de la página que cambian dinámicamente, sin intervención del usuario**, como es este caso, deben anunciarse al usuario de productos de apoyo para que no le pasen desapercibidos los cambios en las mismas. Esto se hace mediante WAI-ARIA, especificación a la que dedicaremos varias lecciones posteriores, pero puedes consultar la técnica ARIA19¹³ si quieres ver un ejemplo de uso del `role="alert"` de WAI-ARIA.

Si sigues todos los consejos que hemos visto en estos vídeos sobre formularios accesibles, mejorarás no solo la accesibilidad de los mismos sino también su usabilidad, y en definitiva la experiencia de tus usuarios.

En las WCAG 2.0 no hay un listado de todas las técnicas relacionadas con los formularios web, por ello te puede ser de utilidad la recopilación que incluyo en el artículo [“Formularios accesibles según las WCAG 2.0”](#)¹⁴

¹³ WCAG 2.0, Técnica [“ARIA19: Using ARIA role=alert or Live Regions to Identify Errors”](http://www.w3.org/TR/WCAG20-TECHS/aria.html#ARIA19), <http://www.w3.org/TR/WCAG20-TECHS/aria.html#ARIA19>

¹⁴ Carreras Montoto, Olga: “Formularios accesibles según las WCAG 2.0 ”: <http://olgacarreras.blogspot.com.es/2009/06/formularios-accesibles-segun-las-wcag.html>, 2009

Autor

Olga Carreras Montoto. Consultora en Experiencia de Usuario y Accesibilidad Web en [“Usable y accesible”](#) (España)

Ya para terminar, estas son las direcciones de mi blog, de mi web y de mi cuenta de Twitter. En ellas puedes encontrar información sobre mí y sobre mi trabajo, así como más artículos sobre accesibilidad.

- Blog: <http://olgacarreras.blogspot.com.es>
- Web: <http://www.usableyaccesible.com>
- Twitter: <https://twitter.com/olgacarreras>

Referencias

- Carreras Montoto, Olga: [“Formularios accesibles según las WCAG 2.0”](#):
<http://olgacarreras.blogspot.com.es/2009/06/formularios-accesibles-segun-las-wcag.html>, 2009
- Carreras Montoto, Olga: [“Ayuda contextual de los formularios más accesible con "aria-describedby" \(WAI-ARIA\)”](#),
<http://olgacarreras.blogspot.com.es/2013/12/ayuda-contextual-de-los-formularios-mas.html>, 2013
- W3C, [Web Content Accessibility Guidelines \(WCAG\) 2.0](#),
<http://www.w3.org/TR/WCAG20/>, 2008

Técnicas de las WCAG 2.0 vistas en esta lección:

- [H90: Indicating required form controls using label or legend](http://www.w3.org/TR/WCAG20-TECHS/H90.html), <http://www.w3.org/TR/WCAG20-TECHS/H90.html>
- [ARIA2: Identifying a required field with the aria-required property](http://www.w3.org/TR/WCAG20-TECHS/ARIA2.html), <http://www.w3.org/TR/WCAG20-TECHS/ARIA2.html>
- [G205: Including a text cue for colored form control labels](http://www.w3.org/TR/WCAG20-TECHS/G205.html), <http://www.w3.org/TR/WCAG20-TECHS/G205.html>
- [G89: Providing expected data format and example](http://www.w3.org/TR/WCAG20-TECHS/G89.html), <http://www.w3.org/TR/WCAG20-TECHS/G89.html>
- [G184: Providing text instructions at the beginning of a form or set of fields that describes the necessary input](http://www.w3.org/TR/WCAG20-TECHS/G184.html), <http://www.w3.org/TR/WCAG20-TECHS/G184.html>
- [SCR21: Using functions of the Document Object Model \(DOM\) to add content to a page](http://www.w3.org/TR/WCAG20-TECHS/client-side-script.html#SCR21), <http://www.w3.org/TR/WCAG20-TECHS/client-side-script.html#SCR21>
- [ARIA19: Using ARIA role=alert or Live Regions to Identify Errors](http://www.w3.org/TR/WCAG20-TECHS/aria.html#ARIA19), <http://www.w3.org/TR/WCAG20-TECHS/aria.html#ARIA19>

Mejorar la accesibilidad para los usuarios con discapacidad casi siempre mejora la usabilidad para todos los usuarios. J.Nielsen