



sipt
servicio de innovación y
planificación tecnológica

UNIVERSITAS
Miguel Hernández

Accesibilidad en formularios web

Técnicas WCAG 2.0 asociadas a la
implementación de formularios accesibles

Técnicas WCAG 2.0 para formularios accesibles

1. Permitir a los usuarios completar el formulario sin limitación de tiempo <http://www.w3.org/TR/WCAG20-TECHS/G5.html> Si hay algún tiempo límite para cumplimentar el formulario, porque haya un timeout por ejemplo, el usuario debe poder anular o ampliar ese tiempo. Por ejemplo podemos tener un checkbox al comienzo de un formulario de múltiples páginas para que el usuario solicite ampliar el tiempo de sesión o cancelar cualquier limitación de tiempo <http://www.w3.org/TR/WCAG20-TECHS/G133.html>
2. Informar al usuario de lo que ocurrirá antes de activar un control de formulario que provoca un cambio de contexto <http://www.w3.org/TR/WCAG20-TECHS/G13.html>. Por ejemplo, unos radio buttons con idiomas que al ir haciendo click cambia el idioma del formulario, deberíamos facilitar información en la pantalla antes de llegar a esas opciones informando que el idioma cambiará al seleccionar uno. Una encuesta que se compone de varios pasos o pantallas, es conveniente informar al comienzo del proceso que al responder a cada pregunta se pasará a la siguiente.
3. Proporcionar un botón de tipo submit para iniciar un cambio de contexto <http://www.w3.org/TR/WCAG20-TECHS/G80.html>. Se trata de un mecanismo que permite a los usuarios explícitamente solicitar un cambio de contexto. En el ejemplo anterior de la encuesta, en lugar de que al responder se inicie un cambio de contexto automático, que tendríamos que informar previamente, se podría utilizar un botón “Siguiente”.
4. Proporcionar descripciones textuales para identificar campos obligatorios que no han sido cumplimentados <http://www.w3.org/TR/WCAG20-TECHS/G83.html>. Debemos notificar al usuario en forma de texto que un campo obligatorio no ha sido completado. Una buena práctica es por ejemplo incluir en el “title” de la página el mensaje de error, cosa que

anunciaría un lector de pantalla de inmediato. Otra buena práctica es incluir mensajes de error en la propia validación del formulario en cliente <https://www.w3.org/TR/WCAG20-TECHS/SCR18.html>

5. Proporcionar descripciones textuales cuando el usuario introduce valores no admitidos en la lista de elementos permitidos en un campo de formulario <http://www.w3.org/TR/WCAG20-TECHS/G84.html>. Una buena práctica sería mostrar una advertencia textual indicando el campo que contiene el error, naturaleza del error (valor no permitido) y un ejemplo de valor permitido, o si hubiera una lista cerrada de valores permitidos se indicarían. No es suficiente devolver el foco al campo con el error y marcarlo en rojo por ejemplo.
6. Proporcionar descripciones textuales cuando el usuario introduce un valor en un campo de formulario que no corresponde con el formato o valores solicitados <http://www.w3.org/TR/WCAG20-TECHS/G85.html>. Igual que en el caso anterior, es necesario informar sobre el campo y naturaleza del error. Se puede utilizar un alert, por ejemplo, que se muestra en cuanto se cumplimenta dicho campo al cambiar a otro, informando del tipo/formato de dato admitido, poner ejemplos e incluso dependiendo del campo, facilitar instrucciones sobre cómo cumplimentarlo. Si se valida el formulario una vez pulsado el botón submit, sería conveniente mostrar el error por debajo de dicho botón y no en la parte superior de la página.
7. Informar sobre el formato de dato que se espera en un campo de formulario y facilitar ejemplos <http://www.w3.org/TR/WCAG20-TECHS/G89.html>. El objetivo es evitar errores en la introducción de datos en un formulario. Por ejemplo, indicando en la label de un campo de fecha el formato que se espera. Si fuera un botón para la subida de ficheros, deberíamos informar sobre los formatos de archivos admitidos, si hubiera un peso máximo admitido, etc.
8. Es necesario proporcionar mecanismos al usuario para que puede revisar y corregir datos antes de enviar el formulario <http://www.w3.org/TR/WCAG20-TECHS/G98.html>. Este criterio es especialmente importante para acciones irreversibles en aplicaciones con implicaciones legales o de transacciones económicas pero es una buena práctica tenerlo en cuenta para cualquier

formulario. En un formulario de una sola página no es necesario porque el usuario puede revisarlo todo en pantalla antes de enviar la información, pero sí es conveniente que el botón submit solicite una confirmación antes del envío. En formularios que conllevan varios pasos es posible que el usuario no recuerde datos introducidos en pasos anteriores, por lo que una solución sería poder volver atrás o al final, antes de enviar, proporcionar un resumen con todos los datos para su revisión antes del envío del formulario.

9. Proporcionar un mecanismo para recuperar información borrada <http://www.w3.org/TR/WCAG20-TECHS/G99.html>. Cuando una aplicación web permita eliminar datos se tienen que proporcionar mecanismos de recuperar la información, por ejemplo con la típica papelera de reciclaje para mantener los datos durante un periodo determinado. Otro acercamiento a este requisito sería almacenar la información de forma que pueda ser restaurada bajo petición del usuario, a modo de historial por ejemplo.
10. Proporcionar un checkbox en la primera página de un formulario de múltiples páginas para que el usuario pueda ampliar el tiempo de sesión o eliminar cualquier límite temporal para cumplimentarlo <http://www.w3.org/TR/WCAG20-TECHS/G133.html>.
11. Crear mecanismos que permitan a los usuarios ir a los campos con errores directamente <http://www.w3.org/TR/WCAG20-TECHS/G139.html>. Por ejemplo, cuando se detecten errores en la cumplimentación de un formulario se pueden poner enlaces a los campos erróneos en el propio mensaje que describe el error.
12. Utilizar componentes de la interfaz de usuario que son destacados por el navegador cuando reciben el foco <http://www.w3.org/TR/WCAG20-TECHS/G149.html>. Simplemente se trata de asegurarnos de que el campo con el foco se identifica visualmente en el navegador. Por ejemplo, que en nuestro css no tengamos ningún campo con `focus outline:0`.
13. Proporcionar un checkbox junto con un botón submit <http://www.w3.org/TR/WCAG20-TECHS/G155.html>. Tiene el objetivo de que el usuario haya revisado toda la información antes de enviarla e incluso

pueda confirmar que ha leído y acepta las condiciones, política de privacidad o lo que corresponda. También se pueden utilizar otras fórmulas, como una alerta advirtiendo si está seguro de querer enviar el formulario.

14. Colocar etiquetas para maximizar la predicción de relaciones <http://www.w3.org/TR/WCAG20-TECHS/G162.html>. Cuando colocamos las labels donde el usuario espera encontrarlas, es más fácil para entender formularios complejos y ubicar campos concretos. Para la mayoría de campos las etiquetas se colocan inmediatamente antes del campo, para idiomas que se leen de izquierda a derecha, bien a su izquierda o justo en la parte superior. Para idiomas que se leen de derecha a izquierda es justo al contrario, es decir, a la derecha del campo o justo por debajo de este. Para radio buttons y checkboxes las etiquetas se esperan después de dichos campos.
15. Proporcionar un periodo de tiempo definido tras el envío del formulario en el que el usuario puede actualizar o cancelar el pedido <http://www.w3.org/TR/WCAG20-TECHS/G164.html>. Esta técnica está más relacionada con asuntos legales como contratos o pedidos y es una buena práctica informar del tiempo que dispone el usuario para modificar o cancelar el pedido realizado.
16. Utilizar un botón cercano para etiquetar el propósito de un campo <http://www.w3.org/TR/WCAG20-TECHS/G167.html>. Cuando un botón realiza una llamada/acción sobre un campo input, el botón también actúa como etiqueta para ese campo (por ejemplo un input de búsqueda con el botón “Buscar”). Esa etiqueta ayuda al usuario a identificar el propósito de dicho campo sin necesidad de repetirlo con una etiqueta como tal. Este tipo de botones normalmente se colocan después del campo input.
17. Solicitar confirmación para continuar con la acción seleccionada <http://www.w3.org/TR/WCAG20-TECHS/G168.html>. Esta buena práctica simplemente busca que el usuario pueda confirmar que la acción que ha realizado es la correcta. Por ejemplo, cuando un usuario espera la colocación de los botones “submit” y “cancel” en un orden diferente y pulsa

un botón rápidamente, por lo que se debe ofrecer las opciones de confirmar o cancelar.

18. Proporcionar una sugerencia de corrección de texto <http://www.w3.org/TR/WCAG20-TECHS/G177.html>. El objetivo de esta técnica es sugerir un texto correcto cuando la información facilitada por el usuario no es aceptada y se ofrecen posibilidades de texto correcto. Por ejemplo, en un campo de texto donde se espera introducir el nombre de una ciudad, se introduce con error de ortografía y se facilita una sugerencia de nombre correcto.
19. Proporcionar instrucciones en texto al comienzo de un formulario o un conjunto de campos que describe el formato de los datos necesarios <http://www.w3.org/TR/WCAG20-TECHS/G184.html>. Esta técnica es muy útil cuando tenemos un formulario en el que varios campos comparten el mismo formato de datos porque de esta forma informamos al principio y no repetimos en cada campo el formato de datos que se espera.
20. Proporcionar revisión ortográfica y sugerencias para campos input <http://www.w3.org/TR/WCAG20-TECHS/G194.html>. Sobre todo pensando en usuarios con problemas cognitivos o de movimiento, que pueden tener problemas a la hora de escribir correctamente. Por ejemplo, en el buscador de Google cuando indica ¿Quiso decir...? y te ofrece el enlace para seleccionar la cadena correcta.
21. Utilizar etiquetas, nombres y alternativas textuales de forma unificada para contenido que tiene la misma funcionalidad <http://www.w3.org/TR/WCAG20-TECHS/G197.html>. Se trata simplemente de mantener una coherencia de tal forma que si tenemos un enlace “Contacto” en la cabecera de la página y otro en el pie, los dos enlaces tengan el mismo texto de enlace.
22. Proporcionar un mecanismo para que el usuario pueda cancelar una posible limitación de tiempo <http://www.w3.org/TR/WCAG20-TECHS/G198.html>. El objetivo es que gente que no pueda realizar una tarea en un tiempo determinado puedan cancelar dicho límite y la propia

funcionalidad no tenga límite de tiempo. Por ejemplo, un banner con noticias que se van actualizando, es posible que el usuario no tenga tiempo suficiente para realizar las tareas oportunas.

23. Proporcionar información confirmando que los datos han sido enviados correctamente <http://www.w3.org/TR/WCAG20-TECHS/G199.html>. Por ejemplo, en un formulario de contacto confirmar al usuario de forma explícita y clara, sin tener que buscar en la página información de confirmación, que los datos se han enviado correctamente y, por ejemplo, recibirá respuesta en un plazo determinado.
24. Crear un orden lógico de tabulación para navegar por los enlaces, campos de formulario y objetos <http://www.w3.org/TR/WCAG20-TECHS/H4.html>.
25. Proporcionar botones de tipo submit <http://www.w3.org/TR/WCAG20-TECHS/H32.html>. Simplemente se trata de un mecanismo para que el usuario solicite expresamente un cambio de contexto. Por ejemplo, en un desplegable con varios destinos no utilizar simplemente el cambio de estado del desplegable para realizar la acción y facilitar un botón de “Ir a la página” para realizar el cambio de contexto o envío de datos.
26. Utilizar elementos label para asociar etiquetas con controles de formulario <http://www.w3.org/TR/WCAG20-TECHS/H44.html>. Se trata de utilizar el atributo “for” con el id del campo de formulario. Aunque la etiqueta puede ser ocultada por css porque pueda ser obvia, es necesario que las etiquetas sean visibles ya que proporciona ayuda a usuarios que necesiten entender el propósito del campo. Elementos que tienen labels asociadas de forma explícita son: text, checkbox, file, password, textarea y select. No se utiliza label para botones submit y reset (utilizan el atributo “value”) o botones de imagen (utilizan “alt”).
27. Utilizar el atributo title para identificar controles de formulario cuando no se puede utilizar el elemento label <http://www.w3.org/TR/WCAG20-TECHS/H65.html>. El objetivo de esta técnica es utilizar el atributo title para etiquetar controles de formulario cuando por motivos del diseño visual no

se utiliza etiqueta, por ejemplo en una caja de búsqueda sin label que solo tiene el botón “Buscar” pondríamos un title al input.

28. Proporcionar una descripción para grupos de controles de formulario utilizando “fieldset” y “legend” <http://www.w3.org/TR/WCAG20-TECHS/H71.html>. El objetivo de esta técnica es proporcionar una agrupación semántica de controles de formulario, ya que ayuda a los usuarios a entender las relaciones entre los elementos y a interactuar con el formulario de forma más rápida y efectiva. Agruparíamos varios elementos relacionados semánticamente con el elemento “fieldset” y el primer elemento dentro del fieldset debe ser un elemento “legend” que proporciona la etiqueta o instrucciones para el grupo de controles. Esta técnica se suele utilizar para agrupar radio buttons y checkboxes relacionados, para agrupar varios campos relativos a una dirección (calle, número...).
29. Utilizar un botón con un elemento select para realizar una acción <http://www.w3.org/TR/WCAG20-TECHS/H84.html>. El objetivo es permitir al usuario controlar cuando se realiza la acción, en lugar que el propio cambio en el estado del select realice la acción. El usuario puede echar un vistado a los elementos disponibles en el select y marcar alguno de forma accidental, y es especialmente importante para personas que solo utilizan teclado para navegar/operar.
30. Utilizar “optgroup” para agrupar elementos “option” dentro de un control “select” <http://www.w3.org/TR/WCAG20-TECHS/H85.html>. El objetivo de esta técnica es agrupar elementos en una lista de selección que tenga un conjunto de valores permitidos, como una lista de selección múltiple o un combo box. A veces esas listas de selección tienen grupos de elementos relacionados, que deben ser semánticamente identificados, como por ejemplo un desplegable de comida con grupos de alimentos clasificados en fruta, verdura, etc.
31. Utilizar el atributo title para ofrecer ayuda contextual <http://www.w3.org/TR/WCAG20-TECHS/H89.html>. La ayuda puede incluir formatos admitidos, ejemplos de valores, etc

32. Indicar campos de formulario obligatorios <http://www.w3.org/TR/WCAG20-TECHS/H90.html>. Es necesario identificar claramente los campos de formularios que son obligatorios para el envío del formulario, bien con texto o con un símbolo en la label. Si se utiliza un símbolo, es necesario informar sobre ello antes de que se utilice por primera vez.

33. Uso de HTML estándar en campos de formulario y enlaces <http://www.w3.org/TR/WCAG20-TECHS/H91.html>. El objetivo de esta técnica es utilizar campos y enlaces con estándar HTML, indicando correctamente su nombre, valor y estado, con el fin de facilitar la operación con teclado y tecnologías de asistencia.