

Recomendaciones para contenido dinámico (WAI-ARIA)

Recomendaciones para contenido dinámico (WAI-ARIA)	1
Introducción	1
¿Por qué esta guía?	1
Ejemplos	2
Recomendaciones	3
Recursos	4

Introducción

¿Por qué esta guía?

Las páginas web tienen actualmente gran cantidad de componentes dinámicos. A menudo, estos componentes no se desarrollan utilizando HTML de acuerdo a las especificaciones. En general se los denomina “custom components”, lo que puede traducirse como “componentes personalizados”.

Cuando se utilizan componentes “personalizados” el uso de WAI-ARIA es necesario para cumplir con las WCAG, especialmente¹ con dos criterios: “[4.1.2 Nombre, Rol, Valor](#)” y “[2.1.1 Teclado](#)”.

El criterio “4.1.2 Nombre, Rol, Valor” indica que “el nombre y el rol pueden ser programáticamente determinados”, y que los “estados, propiedades y valores que pueden ser seteados por el usuario pueden ser programáticamente seteados; y que la notificación de cambios en estos ítems está disponible para browser y tecnologías asistivas.”

¹ También tiene incidencia en el criterio “1.3.1 Información y relaciones”. No se desarrolla, por ser similar a la incidencia en el “4.1.2 Nombre, Rol, Valor”.

Es decir, de utilizarse componentes personalizados, deberá utilizarse WAI-ARIA para poder cumplir con las WCAG.

Por su parte, el criterio “2.1.1 Teclado” establece que los componentes tienen que poder ser operados mediante el teclado.

Ejemplos

A continuación se presentan dos ejemplos para ilustrar lo mencionado en el punto anterior: un “checkbox de acuerdo a la especificación HTML” y un “checkbox usando WAI-ARIA”.

Ejemplo 1. Un checkbox de acuerdo a la especificación HTML:

```
<input type="checkbox" id="label">  
<label for="label">Recordar mi usuario</label>
```

Ejemplo 2. Un checkbox “personalizado” sin usar WAI-ARIA, más Javascript y CSS:

```
<div>Recordar mi usuario</div>
```

Ejemplo 3. Un checkbox “personalizado” usando WAI-ARIA, más Javascript y CSS:

```
<div role="checkbox" aria-checked="false" tabindex="0">Recordar mi usuario</div>  
que al ser chequeado cambia a:  
<div role="checkbox" aria-checked="true" tabindex="0">Recordar mi usuario</div>
```

Los tres se ven de manera similar (Imagen 1), pero sólo los ejemplos 1 y 3 serán accesibles:

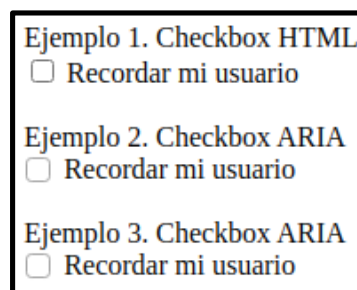


Imagen 1

En el ejemplo 3, se agregó:

- `role="checkbox"`, para indicar a las tecnologías asistivas que `<div>` es un “checkbox”. En términos del criterio de conformidad “4.1.2 Nombre, Rol, Valor”, estamos estableciendo el “Rol” del componente.
- `aria-checked="false/true"`, para indicar el estado. En términos del criterio de conformidad “4.1.2 Nombre, Rol, Valor”, estaríamos estableciendo el “estado” del componente.

El “nombre” del componente es “Recordar mi usuario” en ambos casos².

Por su parte, para cumplir con el criterio “2.1.1 Teclado” se estableció la propiedad `tabindex="0"`, para que el componente reciba foco de teclado.

En síntesis, el uso de WAI-ARIA permite cumplir con los requerimientos de accesibilidad en el caso de no usar HTML de acuerdo a la especificación.

Recomendaciones

1. Construir componentes usando HTML según las especificaciones

Como se vio en el ejemplo 1, si se utiliza HTML de acuerdo a las especificaciones no es necesario utilizar nada más. Se cumplirá con los requisitos de accesibilidad y componente podrá ser utilizado con tecnologías asistivas.

2. En el caso de no seguir lo indicado en 1), construir componentes utilizando WAI-ARIA

Como se vio en el ejemplo 3, construir componentes utilizando WAI-ARIA requiere un buen entendimiento de cómo aplicarlo, y uso de Javascript y CSS.

Según el World Wide Consortium (W3C), “No usar ARIA es mejor que usar ARIA incorrectamente”, en inglés “No ARIA is better than Bad ARIA”. En el [sitio de W3C se dedica](#)

² Aclaración: en el caso de los checkbox no existe un “Valor”, en los términos del criterio “4.1.2”.

[un apartado completo al tema](#). El ejemplo del checkbox es relativamente simple, pero componentes más complejos requerirán aún más conocimiento de ARIA.

Se recomienda consultar en la especificación WAI-ARIA, cuales son los estados y propiedades requeridos y aceptados para el atributo usado. Ejemplo: para el rol “checkbox”, [la especificación requiere el estado “aria-checked”](#), tal como se hizo en el ejemplo 3.

3. En el caso de seguir 2), buscar en la web “componentes accesibles”

En lugar de desarrollar un componente desde cero, se puede utilizar alguno ya desarrollado. Especialmente si está publicado por alguna organización especializada en el tema. Recomendamos particularmente [“WAI-ARIA Authoring Practices 1.2” de W3C, un repositorio y guía de componentes accesibles](#).

4. En el caso de seguir 2), testear

Se recomienda testear utilizando herramientas de verificación automatizadas y tecnología de asistencia (lectores de pantalla, fundamentalmente) en escritorio y móvil.

Recursos

- W3C, 2017. [Accessible Rich Internet Applications \(WAI-ARIA\) 1.1](#).
- W3C, 2019. [WAI-ARIA Authoring Practices 1.2](#).
- W3C, 2018. [Web Content Accessibility Guidelines \(WCAG\) 2.0](#).